A NEURAL NETWORK AND RULE BASED SYSTEM APPLICATION IN WATER DEMAND FORECASTING

A Thesis submitted for the degree of Doctor of Philosophy

by

Joseph Alan Hartley

Department of Electronics and Electrical Engineering,
Brunel University

April 1995

ACKOWNLEDGEMENTS

I would like to thank all the members of the BIPS/OCEPS and WASMACS research groups for their help and advice in completeing the work described in this thesis, in particular I would like to express my gratitude to my tutor Dr Roger Powell who has been extremely supportive and encouraging throughout my studies.

ABSTRACT

This thesis describes a short term water demand forecasting application that is based upon a combination of a neural network forecast generator and a rule based system that modifies the resulting forecasts. Conventionally, short term forecasting of both water consumption and electrical load demand has been based upon mathematical models that aim to either extract the mathematical properties displayed by a time series of historical data, or represent the causal relationships between the level of demand and the key factors that determine that demand. These conventional approaches have been able to achieve acceptable levels of prediction accuracy for those days where distorting, non cyclic influences are not present to a significant degree. However, when such distortions are present, then the resultant decrease in prediction accuracy has a detrimental effect upon the controlling systems that are attempting to optimise the operation of the water or electricity supply network. The abnormal, non cyclic factors can be divided into those which are related to changes in the supply network itself, those that are related to particular dates or times of the year and those which are related to the prevailing meteorological conditions. If a prediction system is to provide consistently accurate forecasts then it has to be able to incorporate the effects of each of the factor types outlined above. The prediction system proposed in this thesis achieves this by the use of a neural network that by the application of appropriately classified example sets, can track the varying relationship between the level of demand and key meteorological variables. The influence of supply network changes and calendar related events are accounted for by the use of a rule base of prediction adjusting rules that are built up with reference to past occurrences of similar events. The resulting system is capable of eliminating a significant proportion of the large prediction errors that can lead to non optimal supply network operation.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

CHAPTER 2

CHAPTER 3

CHAPTER 6

CHAPTER 1

INTRODUCTION

1.1 Introduction to Water Supply Systems in England and Wales.

1.1.1 History

Historically the existence of an organised public water supply system can be traced back to the first Government involvement in such schemes in the 1840's, when pressure for some sort of action to improve drinking water supplies was demanded after a number of cholera outbreaks. Over the next hundred years many small water supply bodies were established, either as private companies or run by local authorities, this led to a very complex and disorganised situation unable to provide for the larger scale and longer term planning decisions required by the growth of water demand with industrialisation.

A succession of Water Acts passed by parliament between 1945 and 1973 led to increasing centralisation of control over water resources, with the establishment in 1973 of the ten regional Water Authorities of England and Wales. Each having overall control of water supply, distribution and sewage treatment within their region but containing semi autonomous remnants of pre-existing water companies.

The 1989 Water Act led to the privatisation of the ten regional Water Authorities which initiated their operation as PLC's while retaining the existing overall geographical boundaries and many of the semi autonomous water companies as shown in figure 1.1.1.

1.1.2 Water Network Structure

A water network is a structure designed to facilitate the movement of water from a source or number of sources, to the locations where the water is required. There are many different interrelating levels of network that can be differentiated in terms of their scale. There are very large scale network schemes such as the Thames Water London Ring Main, which are designed to move large quantities of treated water from sources to distribution points. Both Yorkshire Water and North West Water use connecting networks of rivers, canals and aqueducts to transport raw water large distances from upland reservoirs to the cities where demand is concentrated. A schematic of the North West Water conjunctive use scheme is shown in figure 1.1.2.

The regional water PLC's are subdivided into a number of smaller self contained supply areas, some of which being semi autonomous companies, which control the water supply and distribution to one or more towns or a section of a larger city. Bulk supplies between such areas and any connections to region-wide networking schemes such as described above are generally governed by the regional company strategy, however, the day to day operation of the pumping and storage resources within the area is usually self contained. Within such supply areas are the elements of a typical water supply network, pumping stations for either ground water or surface water abstraction, raw and treated water reservoirs, water towers, booster pumps, treatment plants and the pipework itself. This is shown in figure 1.1.3.

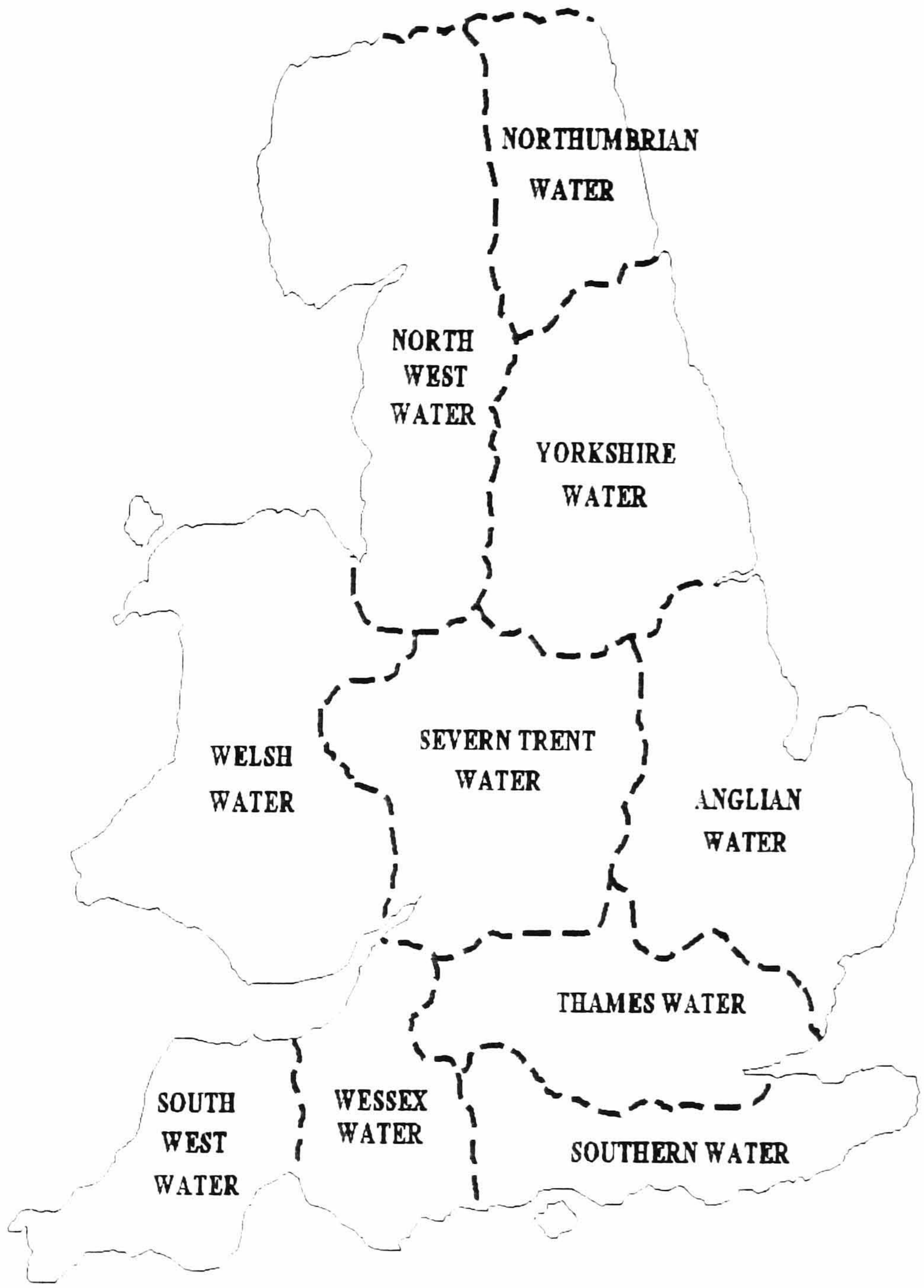


Figure 1.1.1 The UK Water Companies.

Figure 1.1.2 North West Water Conjunctive Use Scheme.

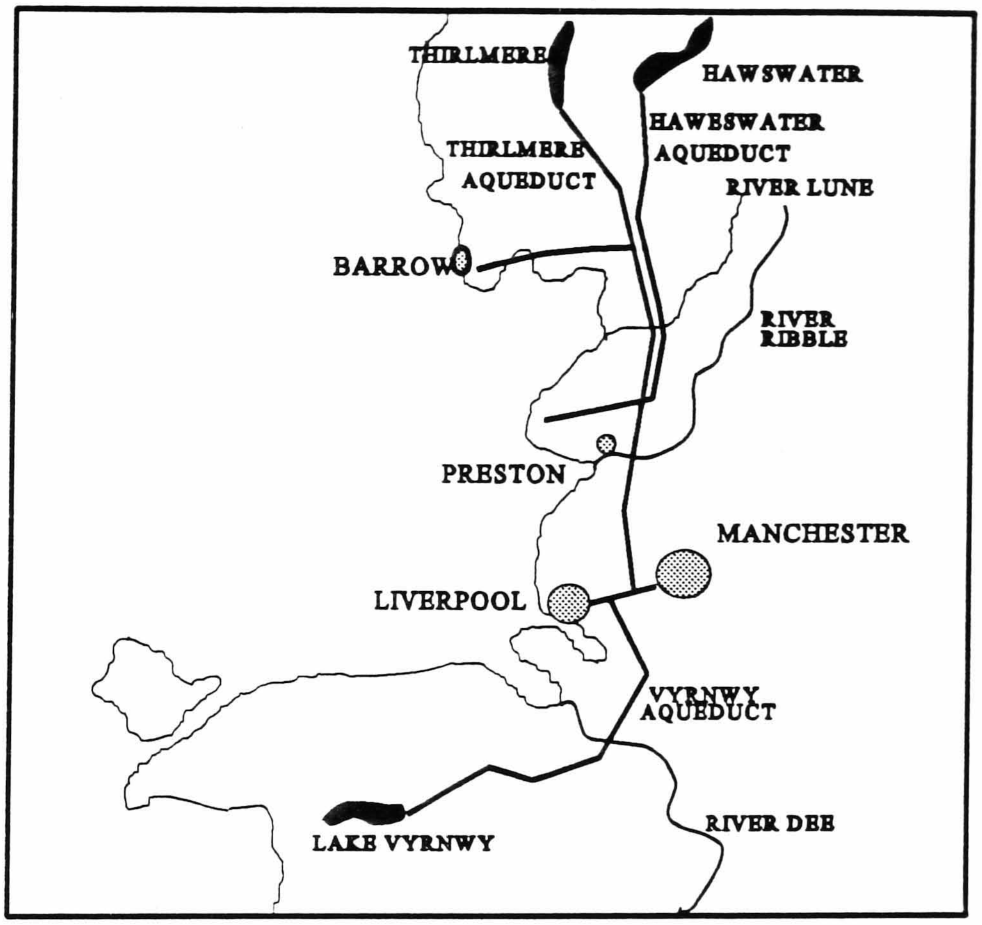
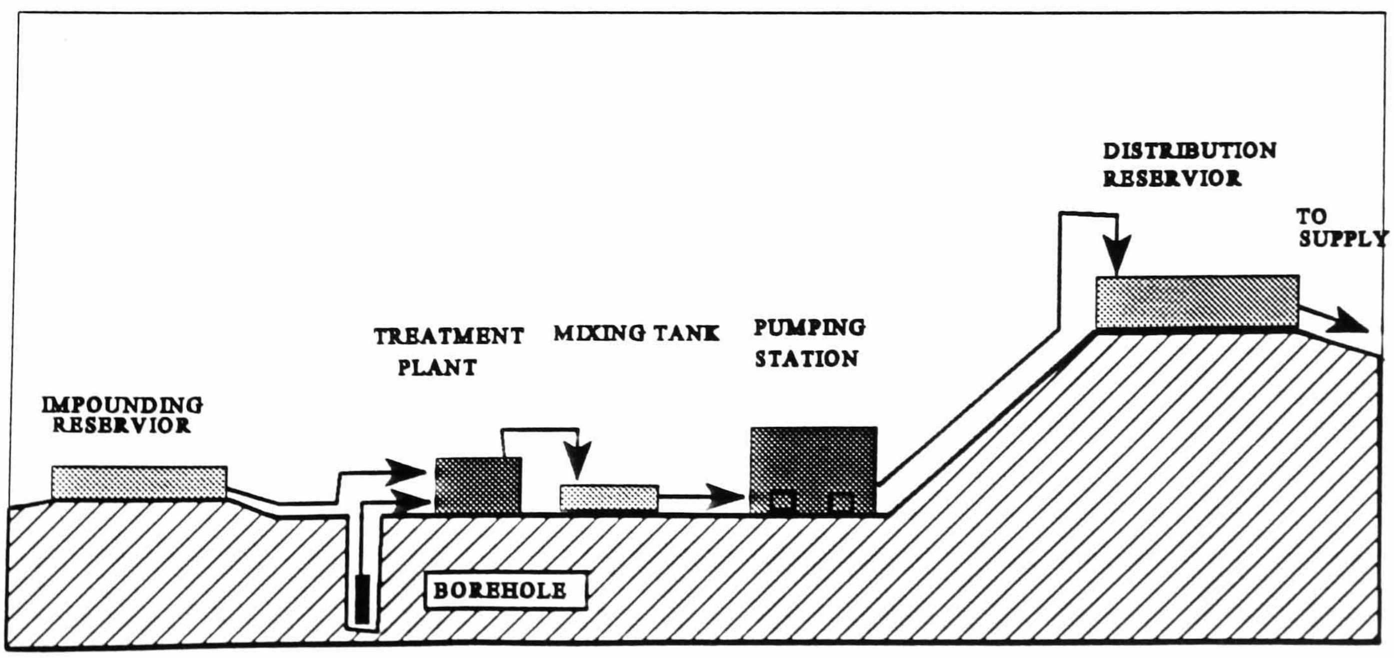


Figure 1.1.3 Key elements of a typical water supply network.



Over recent years the introduction of new technology, principally telemetry systems and automatic control systems for pump and valve operation, has allowed the running of a network to be controlled from one or more central control centres with the minimum of human operators. Figure 1.1.4 shows a diagrammatic representation of the telemetry and control system set up at Thames Water's Bourne End Control Centre which allows a single operator to monitor and control the water network supplying Slough, High Wycombe and Aylesbury.

Within the supply areas outlined above may be a number of pressure zones, these allow the maintenance of adequate water pressure over the varying ground elevations present in the network area. Such pressure zones are commonly interconnected via pressure controlling valves or booster pumps. Finally, within the pressure zones may be districts which are fed through a small number of valves, often with associated pressure or flow measuring devices used for leakage monitoring purposes.

1.1.3 Water Network Technology

As mentioned in the previous section the application of new technology to the field of water network monitoring and control has allowed the gathering of accurate data on the behaviour of the network in response to operational changes, and hence the development of improved network control strategies. Such data collection and control has been achieved principally through the installation of status monitoring devices that transmit information about the network to a central location via a SCADA (Supervising Control And Data Acquisition) telemetry system.

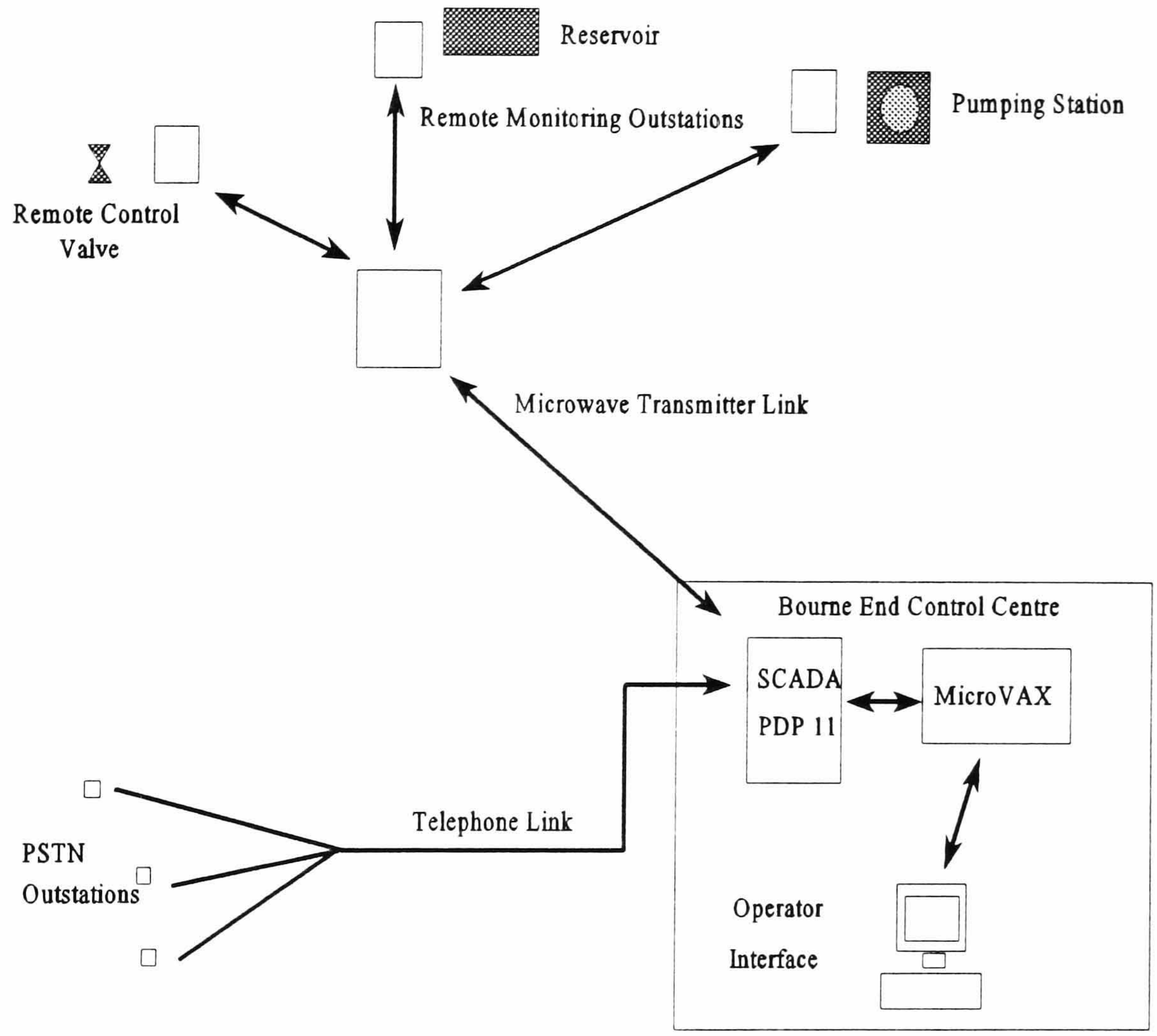


Figure 1.1.4 The SCADA System At Bourne End Control Centre

The basic elements that comprise a water network SCADA system are: the field instruments such as flow meters, pressure transducers, pump status indicators/switches, valve actuators, reservoir level transducers etc; the telemetry system that carries the signals from the field instruments to the control centre, this can be via private or rented telephone wires or microwave radio links etc; the computing hardware and software that receives, interprets and stores the incoming telemetry data, provides the interface to the operator and generates the appropriate control signals.

It is the acquisition of such technology that allows control over the operation of the various elements of the network and the results of any changes to be viewed and archived for future reference, that has opened the door to the possible cost savings that can be achieved by running a network to much finer tolerances than was previously possible. Examples of such cost saving applications are programs designed to produce a minimum cost pump schedule based on the predicted level of water demand and the electricity tariff structure and programs that can assess the security of supply by simulating many 'what if' scenarios. Applications such as these have in turn highlighted the need for accurate methods of predicting future patterns and levels of water consumption. A pump scheduling plan that is based on a prediction of demand that turns out to be grossly in error will not represent the best use of pumping resources and a security assessment based on erroneous demand prediction data could have serious implications on the ability to maintain supplies.

1.2 Water Consumption.

1.2.1 Components of Water Consumption

Water consumption can be divided into various components each contributing towards the total observed consumption. Domestic consumption is defined as that associated with all forms of domestic activity, this includes drinking water, washing of clothes, flushing toilets, garden watering etc. Not only have population levels increased thereby increasing the overall demand for water but also the level of water consumption per household has been gradually increasing as water consuming domestic appliances such as dishwashers, automatic sprinklers etc. have become more commonplace. Estimates of the average amount of water consumed per head of population in the UK increased from 150 l/head/day in 1972 to 230 l/head/day in 1989 [13,39,99].

Industrial and commercial consumption ranges from the large amounts of water (typically between 10 to 100 l/sec) used by industries such as chemical, foodstuffs and paper manufacturing, through the medium water usage (1 to 10 l/sec) of light industry to the relatively small amounts of water used by offices and shops. Many industrial and commercial water users have their supplies metered, data from such meters can be of great value in accounting for their levels and patterns of water consumption when carrying out such tasks as network modelling and demand prediction.

Agricultural water usage can be divided firstly into water used for irrigation purposes, which may be abstracted privately by the farmer from surface or ground water sources, usually by licensed agreement with the water company. Secondly there is the water consumption associated with such activities as livestock watering, dairy production etc.

Losses of water from the network can account for a significant proportion of the water pumped into supply. Leakage occurs from both pipework and reservoirs, the amount of water lost can be up to 40% of the total supplied in areas with bad leakage problems. Other losses can be through activities such as reservoir washing and mains flushing, as well as water not registered due to faulty flow meter readings.

1.2.2 Factors Affecting the Level of Water Consumption

Each of the above components that contribute to the total amount of consumption is influenced by a complex interrelating set of factors that governs both the shape and level of the water demand profile. One of the most significant factors in determining the characteristics of the water consumption patterns in a particular area, is the relative proportions of the domestic, industrial and agricultural components that are geographically and demographically present. A mainly rural area with a few small towns will have very different demand characteristics to a large urban area with a significant amount of industry.

The domestic weekday water demand profile typically has the characteristic shape displayed in figure 1.2.1. This shows that a low level of usage through the night is followed by a sharp pick up in demand in the early morning as people get up, wash and prepare for work. A peak is reached between 08.30am and 10.00am and a gradual tail off occurs until another pick up in the late afternoon/early evening as people return from work. The demand reaches a second peak in mid to late evening then tails off towards the night-time consumption level, often with a small pick up just before midnight as people prepare for bed. The pattern for weekend consumption is similar to that of a weekday in normal weather conditions, but the morning pick up is significantly later and less sharp due to less people getting up simultaneously to go to work, this can be seen in figure 1.2.2. The work by Cubero [36] and Cembrano [120] highlights the correlation between the patterns of daily

Figure 1.2.1 Typical Weekday Demand Profile

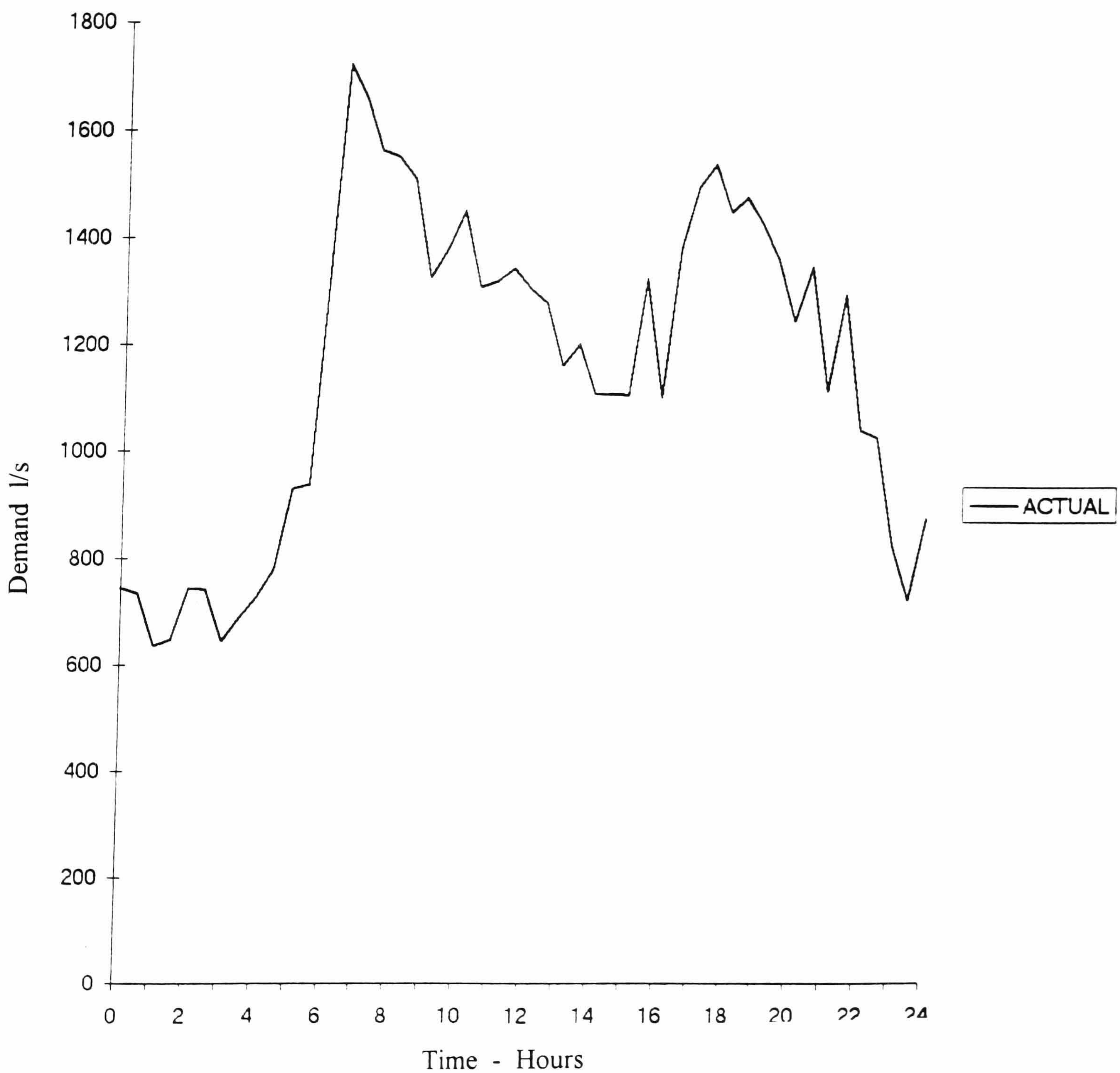
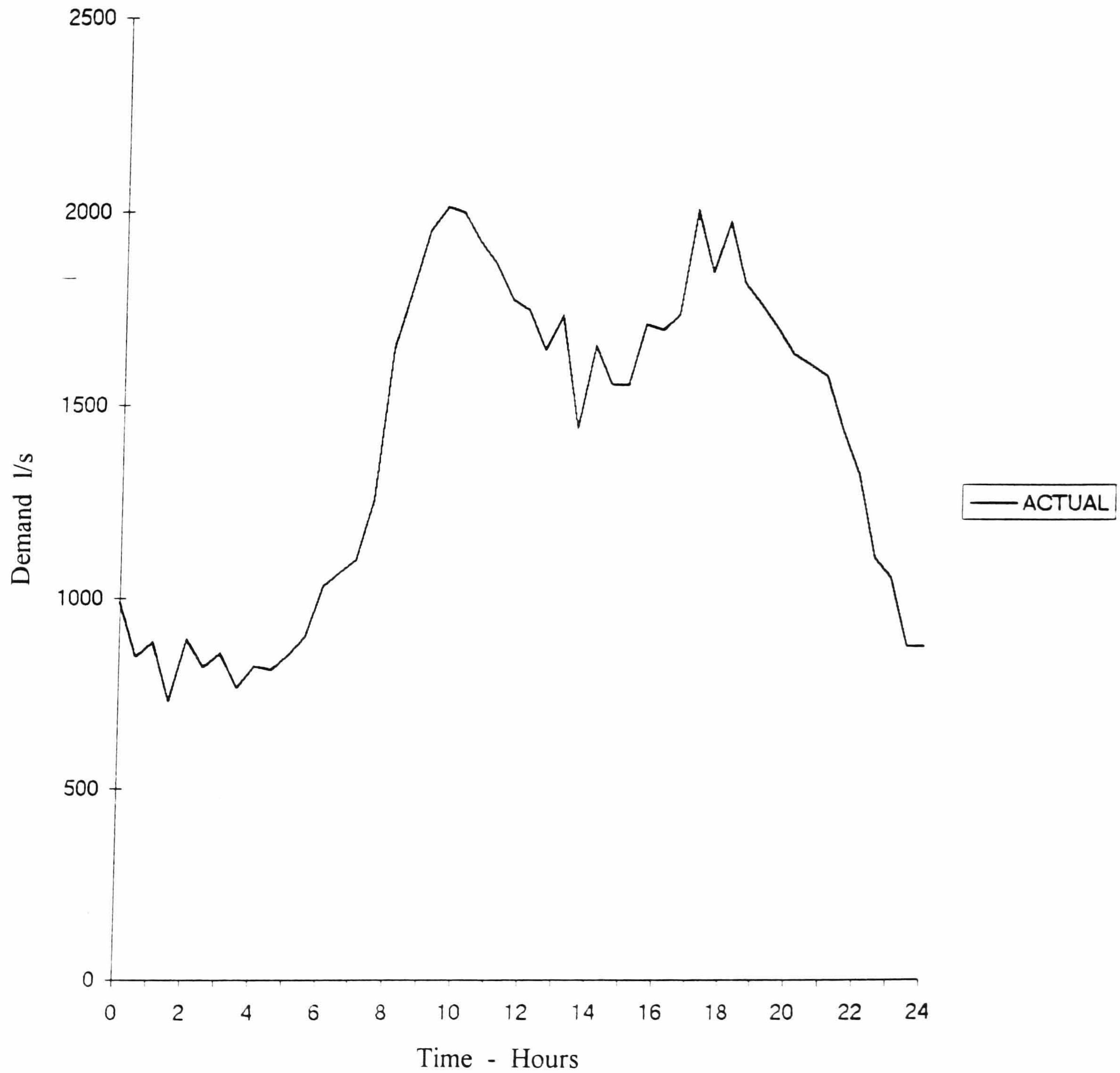


Figure 1.2.2 Typical Weekend Demand Profile



water consumption which result from the weekly cycle of social behaviour. As table 1.2.2 shows, the strongest correlation between daily demand values occur at time lags of 1, 7, 14 and 21 days. This indicates that the current days demand is strongly influenced by the previous days demand and the demand experienced on the same day over a number of preceding weeks.

There are many factors that can influence the level of domestic water usage, but to be significant they need to produce a change in the water consumption characteristics of a large proportion of the population within a particular area. Public Holidays are an example of a situation when a significant proportion of the population changes its routine, the demand profile of such holidays being closer to that of a Sunday than the normal weekday pattern. The effects of public and school holidays on water consumption are particularly evident in popular holiday locations which experience a large sudden influx of people resulting in a sharp increase in the level of water demand.

The weather can also have a significant effect upon the population's water usage characteristics. Increases can be seen in the level of water consuming activities such as clothes washing and car washing. There is also the more direct effect of an increase in garden watering during a period of hot, dry weather. Studies have shown [2,13,141] that the addition demand caused by garden watering can be between 20% - 40% of the average level of demand. The factors influencing the increases in consumption due to hot weather are not only the absolute temperature on any given day but also the number of hours of sunshine and the length of time since the last rainfall (termed the number of antecedent dry days). Figure 1.2.3 shows the profile from a hot day in June, the overall level of demand is increased during the day and the evening peak is large and spread over 5 hours.

Table 1.2.2 Autocorrelation of total daily demand over 3 weeks. (Reproduced From [102])

No. of Days Prior to Current Day	Autocorrelation Function
1	0.59
2	0.06
3	-0.07
4	-0.10
5	-0.01
6	0.35
7	0.71
8	0.33
9	-0.04
10	-0.14
11	-0.15
12	-0.07
13	0.30
14	0.63
15	0.26
16	-0.12
17	-0.21
18	-0.21
19	-0.13
20	0.21
21	0.54

Figure 1.2.3 Weather Influence On Water Demand

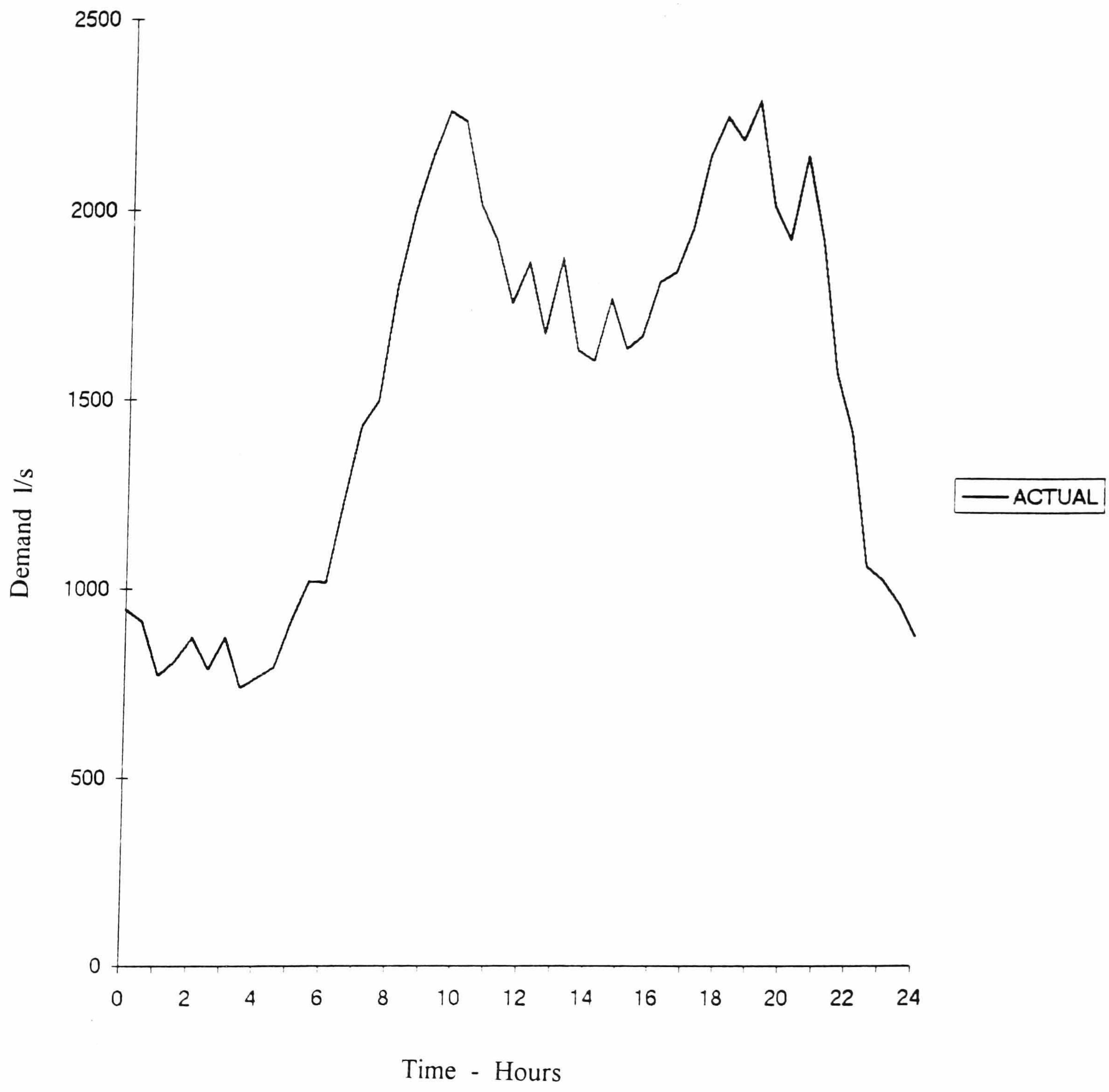
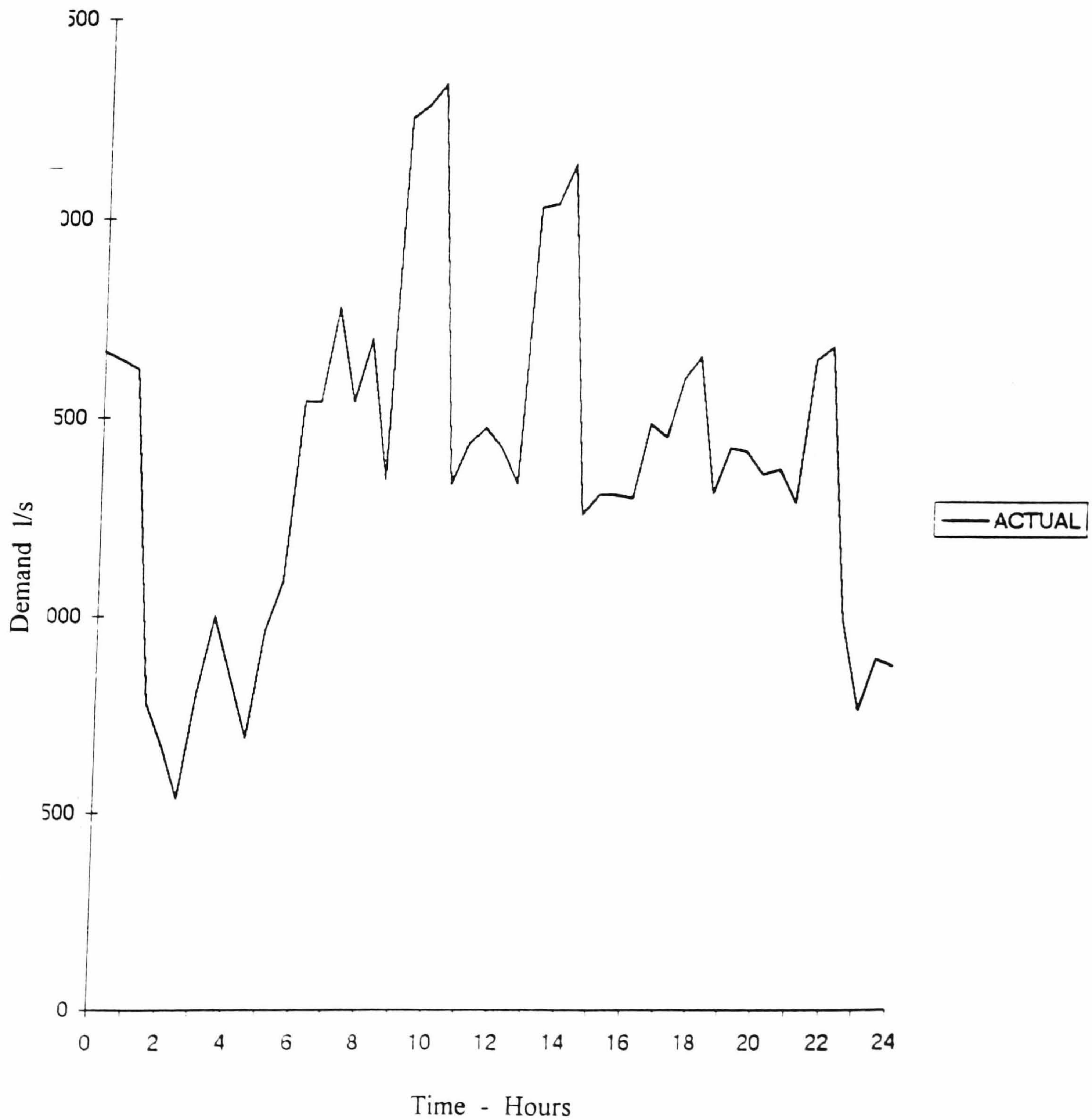


Figure 1.2.4 Industrial Influence On Water Demand



Industrial demand may have a unique profile, dependent upon the types and quantity of industry present within a supply area. The profiles from a heavily industrialised area can show radically different characteristics to that of a mainly urban area, figure 1.2.4 shows the diurnal profile form the Knowsley Industrial Park area of Liverpool. However, as previously mentioned, most large and medium industrial users either have their own water sources or they have their demand metered hence making the task of incorporating their effects into a demand prediction easier. The smaller industrial users and the commercial users commonly exhibit strong weekday/weekend variations, the weekend usage being significantly less than that during the week. The occurrence of public holidays means a large proportion of the industrial and commercial water use is reduced while the effects of events such as industrial disputes in major users can also reduce demand.

Agricultural usage for irrigation, if not taken from private sources, will have an effect upon observed demand that will be both seasonal and weather dependent. Activities such as harvesting can lead to a sudden increase in demand levels when the washing of produce is required.

The amount of water lost from the network due to the effects of leakage is dependent on three factors. The first is the condition of the pipework in the ground, this in turn is a function of the age of the pipes, the materials from which they are constructed, the effectiveness of any coatings applied to the pipes and the chemical characteristics of the water passing through them. The poorer the pipe condition the more leakage is likely to occur. Secondly, the pressure at which the pipes are operating will have an influence upon the leakage level, the higher the pressure the greater the likelihood of a break in the pipework and if a burst does occur then the higher pressure will result in a greater volume of water lost. The relationship between pressure and leakage has been documented in an industry standard publication, the Water Research Centre Report 26 [154], which showed the volume of water lost due to leakage to be a function of pressure.

The third factor governing the level of leakage, is the amount and effectiveness of the leakage repair work carried out by the water company in the area in question.

It should be emphasised that the factors that have an effect on the level of water consumption rarely do so in isolation. It is much more likely that a number of superimposed effects will be exerting a degree of influence upon the observed demand at any one time.

1.3 Demand forecasting.

As can be seen from the multiplicity of factors and effects outlined in the previous section, the accurate prediction of future levels of water demand is a complex task that requires the careful investigation of the demand characteristics of the water network being studied. In this respect, the more data that is available on the reactions of the network to previous conditions and the resulting variations in the levels of demand, the greater are the chances of correctly accounting for similar variations in the future. However, it is by no means certain that a complete and comprehensive catalogue of past data will be available. Data can be corrupted by the telemetry system, it can be lost due to a computer or instrument failure or can be absent altogether, therefore increasing the difficulty of the prediction task.

Predictions of future demand can be divided into three categories, each with their own specific uses. Long term demand forecasting is concerned with the prediction of the likely levels of demand several years in the future and is used for long term planning decisions such as predicting capacity requirements based upon the expected demographic and per-capita water usage changes. This allows the pumping and storage capacity of an area to keep pace with the water demands of the local population. Medium term demand forecasting is concerned with periods up to a year in advance and is usually aimed at contingency planning. This involves ensuring the pumping and storage strategy can cope with the peak month, peak week, and peak day predicted demands, carrying out 'what if' scenarios to test the ability to maintain supply if a pumping station or reservoir is out of service for a significant period of time.

Short term water demand forecasting is aimed at the successful prediction of water demand variations over a period of up to 3 or 4 days in the future, however most water utilities run their short term network control strategy over a prediction horizon of 24 hours. This 24 hour control strategy involves the production of an efficient pumping schedule based on three governing factors which are, meeting the predicted demand, minimising the pumping costs within the existing tariff structure and providing a security of supply capacity buffer in case of unforeseen circumstances. The studies described in this thesis are concerned with development of a methodology for the production of accurate demand forecasts over this 24 hour prediction horizon, thereby ensuring that a pump schedule based upon such a demand forecast remains valid in the light of the actual pattern of consumption experienced during the prediction day.

1.4 Presentation and Content of this Thesis

This thesis presents a study of the application of knowledge based and neural network techniques to the problem of generating accurate short term water demand forecasts. The development of an operational system and its testing and validation against water consumption data supplied by Thames Water are described, together with comparative results produced by submitting the consumption data to an auto-regressive integrated moving average (ARIMA) algorithm.

The present chapter gives an introduction to the field of water supply, in particular the components and characteristics of water networks and the technology used to monitor the behaviour of such networks. The various components identifiable as contributing to water consumption patterns are discussed along with the factors that cause variations in these components. The diverse and often interrelating nature of these variations highlights the problems to be overcome if a prediction methodology is to be sufficiently accurate for use in optimal network operation strategies.

Chapter 2 presents an overview of previously published methods of short term forecasting. It is argued that methodologies that have been developed for use in short term load forecasting within the electricity generating industry are relevant for study in relation to the topic of this thesis. The reason for this being the close similarities between the daily, weekly and seasonal cycles displayed by electrical energy usage and those displayed in the patterns of water consumption. The previously published work in the field of power systems load prediction is discussed, with a division being made between quantitative mathematically based methods such as spectral analysis, exponential smoothing and linear regression, and heuristic methods such as pattern matching, expert systems and load dissagregation. The advantages and disadvantages associated with the various approaches are discussed. A review of previous work associated with water systems demand prediction is then presented, again with the division being made into quantitative and qualitative approaches. The chapter concludes with a discussion of the degrees of success the reviewed approaches have achieved in overcoming the problems associated with short term water demand prediction.

Chapter 3 describes in detail the development and implementation of an ARIMA time series analysis based prediction algorithm. At present, this type of algorithm is commonly used by electricity and water supply utilities to generate predictions of future demand, it is therefore a suitable algorithm to use in this thesis as a benchmark, against which to compare the performance (in terms of prediction accuracy) of the methodologies described in Chapters 4, 5 and 6. Examples of predictions generated by the ARIMA algorithm for both electrical load data and water consumption data are provided and the relative merits and shortcoming of such a prediction methodology are discussed. The evidence presented shows that acceptable prediction accuracy can be achieved by an ARIMA demand predictor when the level of demand is governed almost solely by the stable diurnal and weekly cycles in the pattern of electricity or water usage. However, large discrepancies can develop between the predicted and actual demand when the influence of non cyclic external factors such as extreme weather conditions become significant.

Chapter 4 introduces a methodology designed to facilitate the incorporation of heuristic knowledge into the process of generating demand forecasts. This heuristic knowledge relates to events of a non cyclic nature that distort the demand profile and prevent purely mathematical prediction methodologies from consistently achieving the required prediction accuracy. The application described in this chapter is a rule based system developed within the POPLOG environment and written in the Artificial Intelligence orientated programming language POP-11. A subdivision of the non cyclic effects is made into Calendar, Network and Weather related effects, this division being useful in the categorisation and construction of the rules that are designed to account for the effects. The rules themselves are stored in the built in POP-11 database and are extracted and applied to the raw prediction profiles by a controlling inference engine. This inference engine also provides a menu driven user interface that allows the alteration of the rules held in the database.

The prediction profiles passed to the rule based system described in Chapter 4 are generated by a neural network prediction application that is detailed in Chapter 5. Following an introduction to the concepts and components of neural networks, a detailed description is given of the single layer linear associative network and the training algorithm that is used in the demand forecasting application. This network accepts as input a day type classification value that is based on the current and historical meteorological conditions and outputs a 24 hour prediction profile composed of 48 data points. The day type classification method provides a degree of coarseness to the meteorological data that avoids the requirement for the neural network to provide a mapping of each of the complex relationships between values of individual meteorological variables and resultant demand levels. Results of the comparison between predictions generated by the ARIMA algorithm and those produced by the neural network show that the neural net is significantly more successful at correctly predicting weather dependent demand.

Chapter 6 provides a review of related work into the use of neural networks for electrical load prediction and water demand prediction. Two additional neural network based demand forecasting applications are then described, each uses a network architecture that is significantly more complex than the single layer linear associator. The first is a back propagating neural net that takes as input the values of five meteorological variables and generates as output a 48 data point 24 hour prediction profile. The aim in the development of this network was to investigate the success or otherwise of doing away with the day type classification utilised for the linear associator and attempting to use the neural net to create a direct mapping between weather variables and the resultant demand profile. Results produced by the back propagating network showed little consistency in the accuracy of the predictions. The second type of network investigated was a counterpropagation network. This network is composed of two layers, the first layer is a Kohonen layer that attempts to perform the day type classification task carried out by the FORTRAN routine in the linear associator application. Meteorological variable values from a large number of training day examples are applied to the Kohonen layer which performs a classification on this example data such that similar examples always trigger the same Kohonen neuron. The Grossberg layer that comprises the rest of the network then takes the input from the triggered Kohonen neuron and generates a 48 data point prediction. Results generated by this network showed that the classifications performed by the Kohonen layer were highly unreliable and therefore compromised the prediction accuracy.

Chapter 7 provides a summary of the work described in the thesis, this highlights the problems encountered in developing an accurate demand forecasting system and describes the implementation of the novel approaches used in overcoming these problems. A brief outline of the possible future development of demand forecasting in the water industry in the UK is also provided.

CHAPTER 2

REVIEW OF PREVIOUS WORK

2.1 Introduction.

There are several important factors to be considered when designing a system for forecasting the future values of some variable or variables. Firstly, there is a need to define exactly what is to be forecast, secondly an evaluation must be made of the degree of accuracy required of the forecast and thirdly an examination must be conducted of the data available to the system upon which the forecast is to be based. This requires a careful study of the decision problem the forecasting system is aiming to solve.

Many forecasting problems are examples of attempts to predict future values of one or more variables which are dependent on a process that can be almost completely explained by one or more causal effects but also include a random element. The presence of this random element within such processes ensures a degree of error will always be present in the resulting predictions. However, although a degree of forecast inaccuracy introduced by the effects of this random element must be accepted as inherent, by increasing the understanding of the causal elements of the process we can minimise the magnitude of the prediction errors.

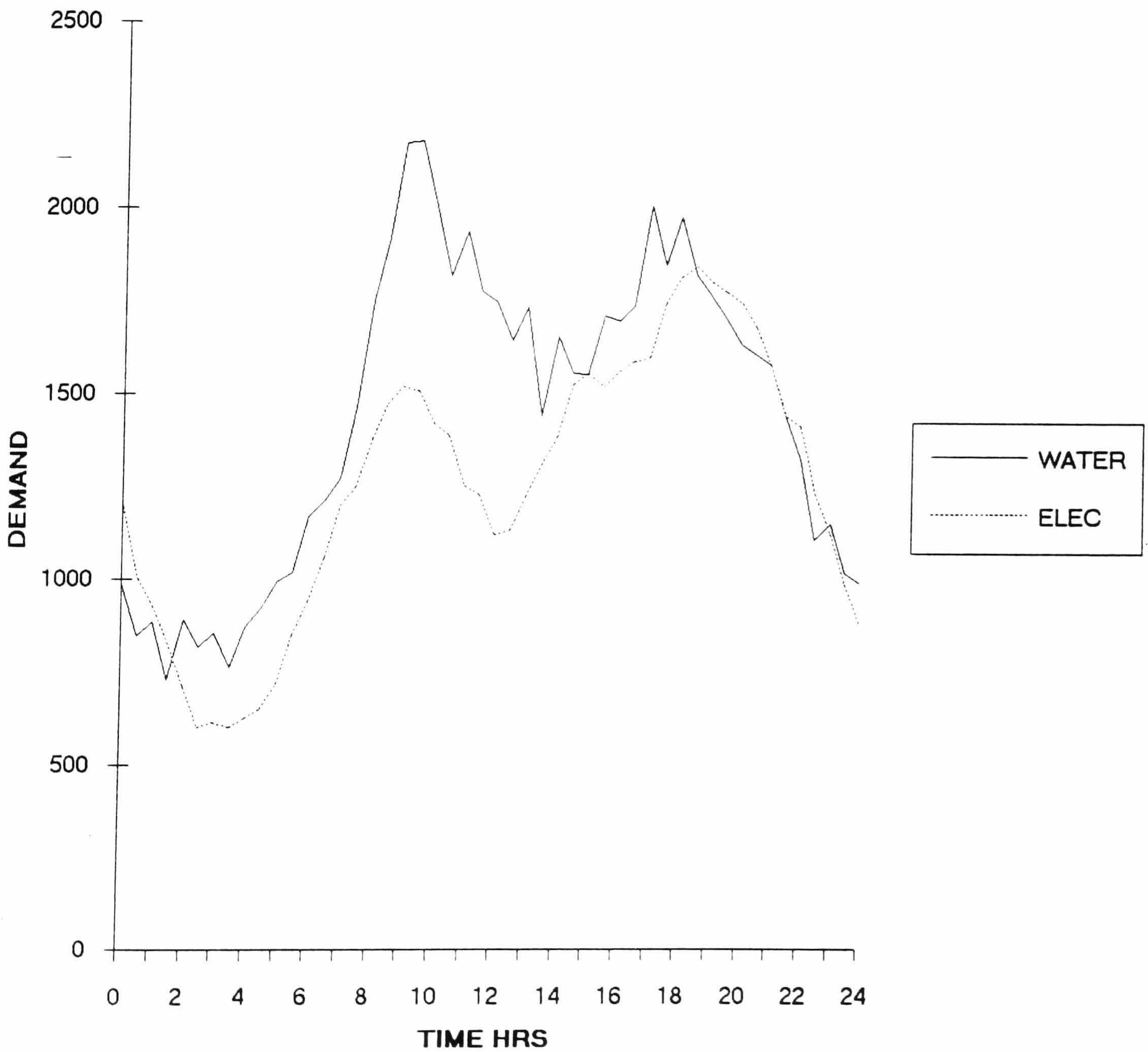
The degree of accuracy achieved in the modelling of the underlying causal processes is largely dependent upon the resources and effort directed towards the development of the forecasting system. This in turn is likely to be a reflection of the importance of the decisions that are to be based upon the forecasts produced. As the costs of developing a more accurate forecasting system increase, so the risk of arriving at an erroneous decision is decreased.

The field of short term prediction of electrical power consumption is very closely related to that of short term water demand prediction due to the similarity of the prediction problem. This has resulted in several algorithms which are common to both applications. It is therefore necessary to investigate the large body of work that has been undertaken into the prediction of electricity consumption (termed electrical load prediction) in order to fully understand the influence of such work upon the development of water demand prediction methodologies.

2.2 Load prediction in electrical power systems.

There are several factors that make investigations into the work carried out in the field of short term electrical load prediction relevant to the work described in this thesis. A comparison of the typical daily profiles for both electricity usage and water consumption for an urban area, highlights strong similarities in the characteristics of the two profiles, this can be seen in figure 2.2.1 . The diurnal cyclic variations observable in both profiles

Figure 2.2.1 Comparison of Electrical and Water Demand Profiles



are generated by the patterns of behaviour of the general population in terms of domestic, industrial and commercial activity.

It is also the case that methodologies developed initially for forecasting applications in one field have been adapted for use in the other, a good example being that of the ARIMA (Auto Regressive Integrated Moving Average)[17,151] algorithm described in detail in Chapter 3 of this thesis. This was initially developed and applied in the prediction of electrical load (examples are Box and Jenkins in 1970 and Vemuri et al in 1981), however, the algorithm was later adapted for use in water demand prediction[55,120,143].

Electrical load data exhibits strong cyclic patterns, these correspond with the daily, weekly and seasonal variations in the way the population of the society in which we live consumes electricity. Figure 2.2.2 shows a typical weekday electricity consumption pattern, termed a load profile. The characteristics of this typical profiles are as follows. A low level night usage between the hours of 1am and 5am is followed by a sharp morning pickup beginning an approximately 6.30am. The morning consumption peaks between 8am and 10am and there are further peaks around midday and in the early evening. During the evening there is a gradual tailing off back to the night consumption level. As with water consumption there are significant differences between weekday and weekend daily profiles, a typical weekend load profile can be seen in figure 2.2.3.

Deviations from the normal cyclic pattern of electricity consumption can be caused by a number of factors, some of which are common to both electricity and water networks, though the manifestations of their effect may be different (i.e. the effect of a period of high temperatures may be to increase water usage due to garden

Figure 2.2.2 Typical Weekday Electric Load Profile

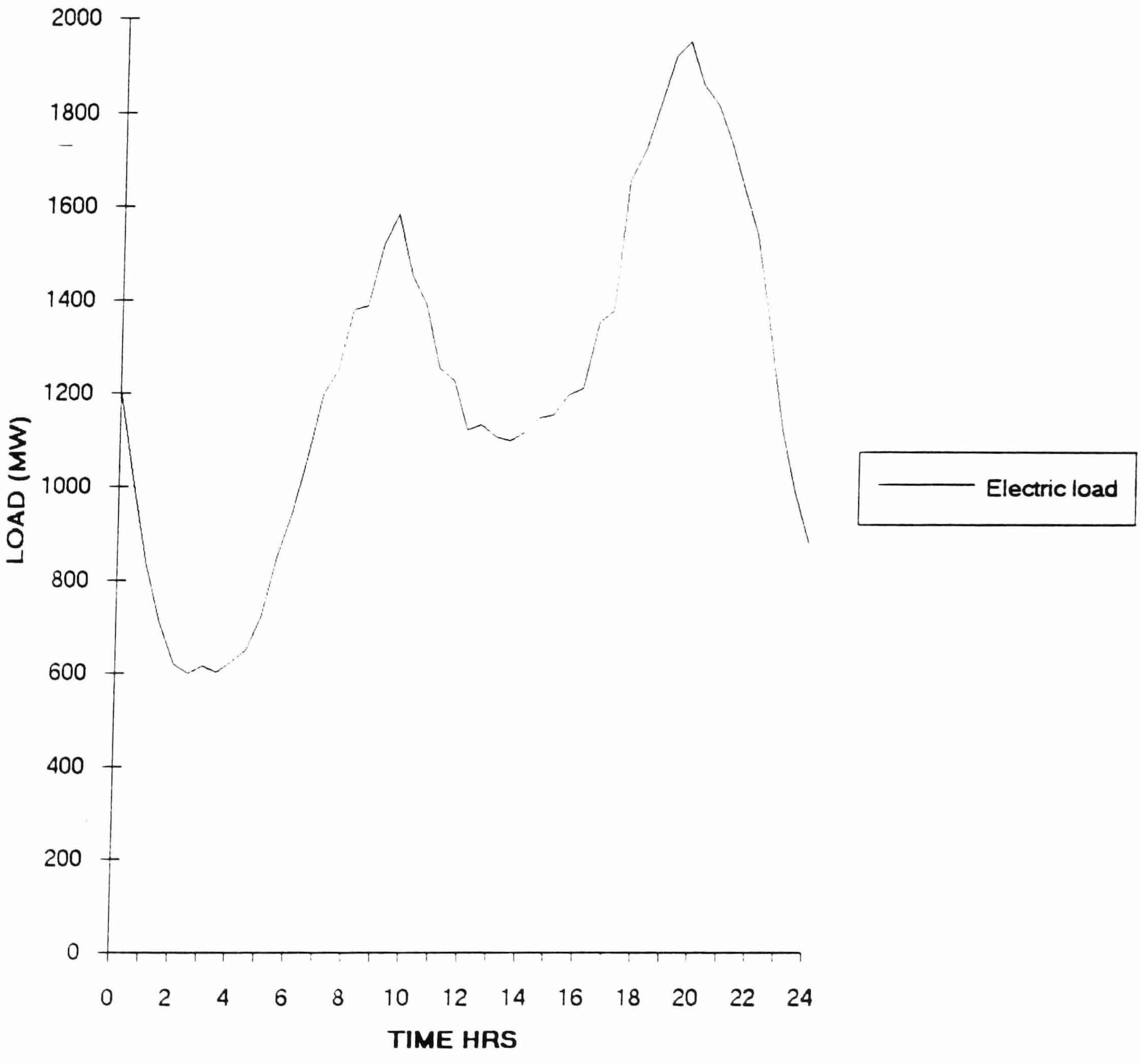
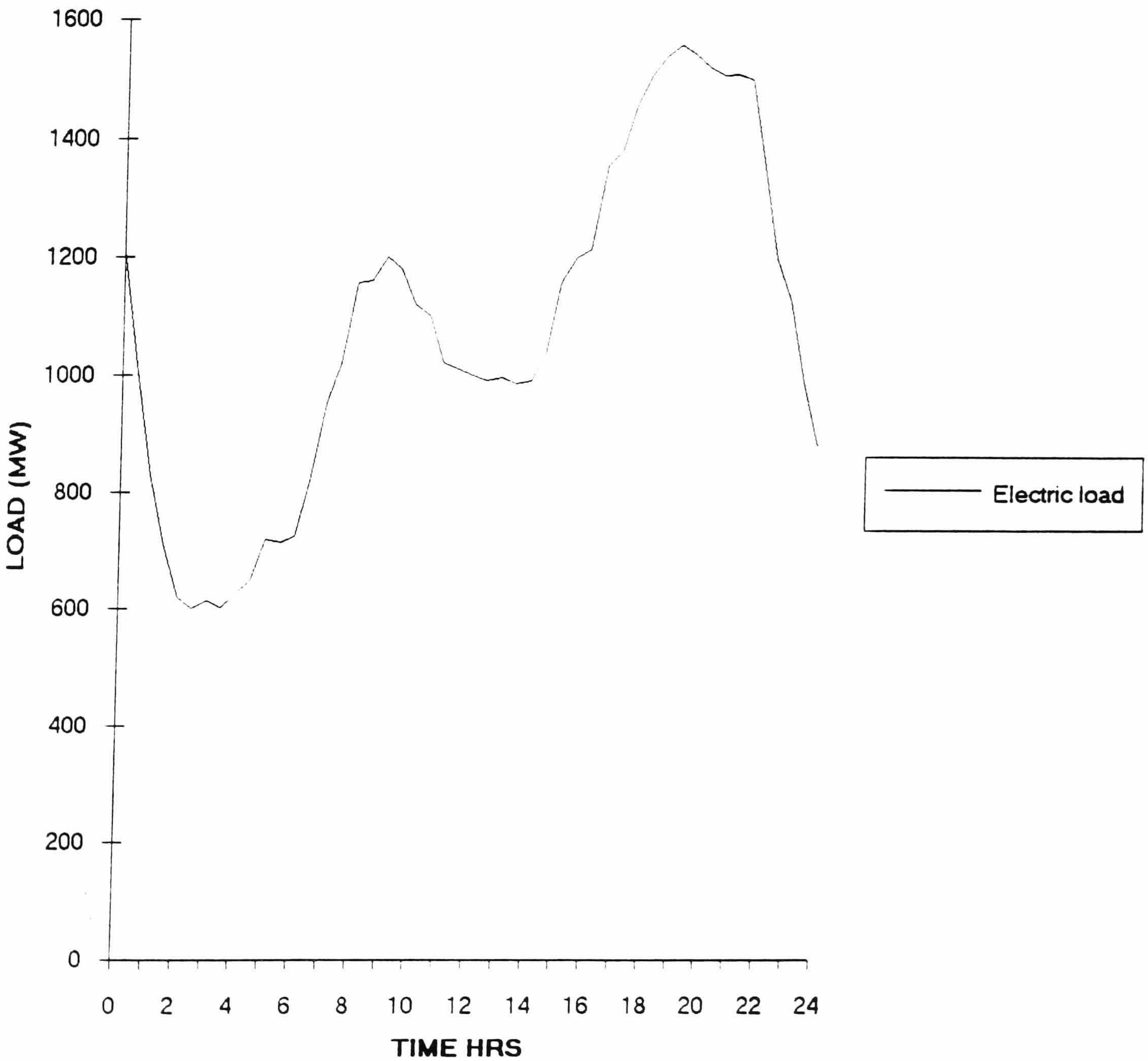


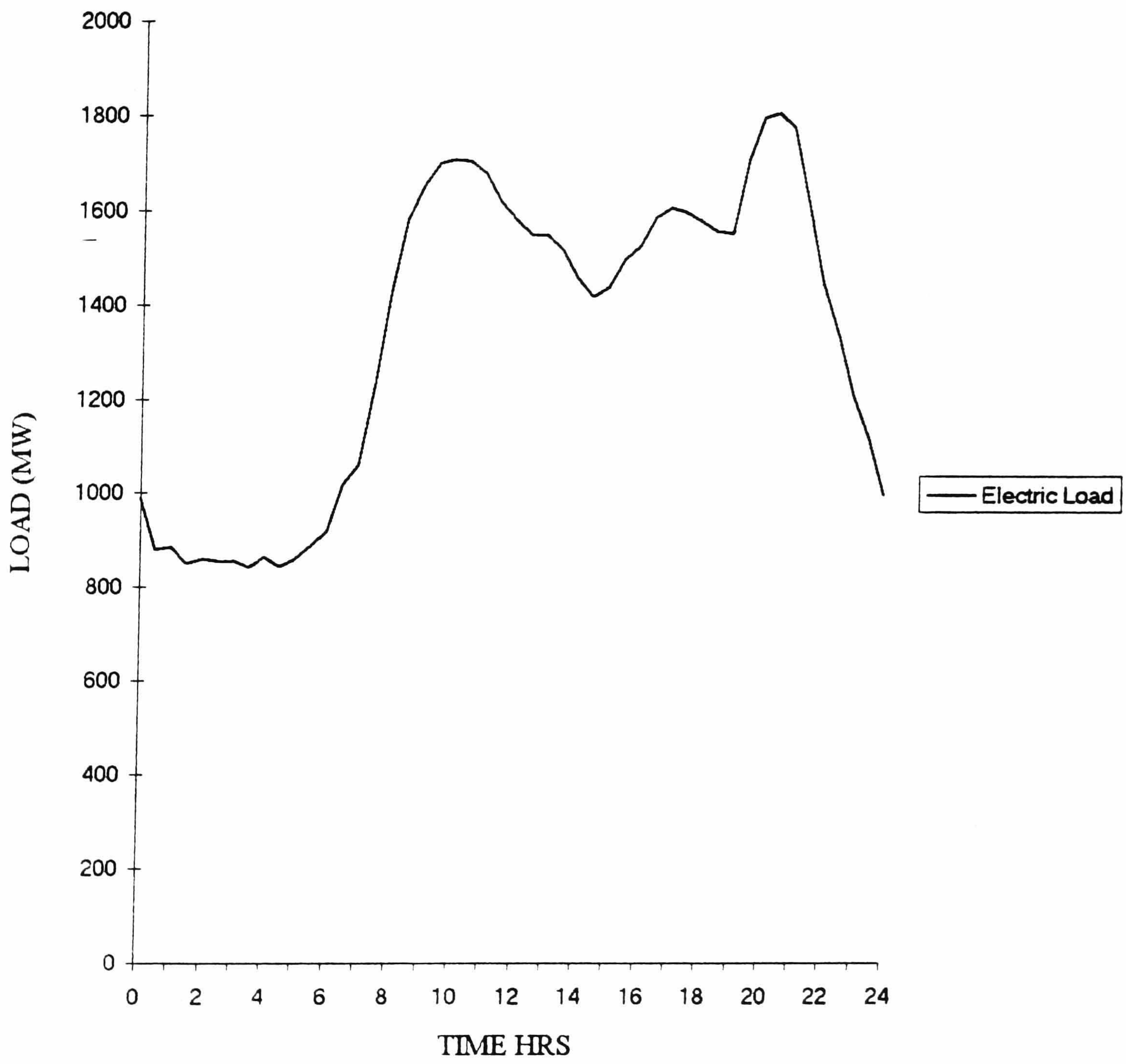
Figure 2.2.3 Typical Weekend Electric Load Profile



watering, but the electricity consumption may decrease due to reduced heating requirements). Examples of factors that affect the shape of the electrical load profile include weather related factors such as temperature, humidity, cloud cover and wind speed, all of which have an influence on the heating and lighting requirements of the population. Day length also has a significant influence upon the load profile which alters as the hours of daylight vary through the seasons, this being particularly evident around the transitional periods of spring and autumn. Public holidays have a marked effect upon electricity consumption, as can be seen in the August Bank Holiday Monday profile shown in figure 2.2.4. Shorter term factors such as the peaks following the ends of important social, media or sporting events have a far greater immediate impact upon the electricity consumption than upon that of water. The power supply companies have to anticipate the occurrence of such surges in load in order to ensure they have enough standby generating capacity (termed spinning reserve) to meet the demand, whereas the water utilities have the buffering effect of reservoir storage to smooth out the impact of such short term demands. The requirement to maintain adequate reserve generating capacity and hence be capable of predicting peaks in demand illustrates one of the important uses of short term load forecasting in the power industry. Load forecasting is also used in applications such as plant ordering, where the on or off status of each generating unit is determined in order that the days peak demand can be safely met and economic dispatch, where the output of each generating unit determined as being in operation is specified for each time interval of the forecast period.

The importance of load prediction to the power generation industry is reflected in the amount of research conducted in this field over the past two to three decades. In order to provide a coherent review of the significant published work on electrical load

Figure 2.2.4 Bank Holiday Electric Load Profile



prediction, a general division of the approaches described in the literature has been made into quantitative and heuristic approaches.

2.2.1 Quantitative Approaches

Quantitative approaches to load prediction are mathematical and statistical methods that involve the examination of historical data in order to determine the underlying processes generating the observed variations in the load. Assuming that these underlying processes are stable, they can be modelled and then extrapolated to produce the required prediction. A subdivision of such quantitative approaches can be made into time series based methods, causal approaches and those applications that combine elements of both.

2.2.1.1 Time Series Based Methods

A time series is a time ordered sequence of observations of a variable. Time series analysis is the use of mathematical techniques to develop a model that can accurately track the observed variations in these past data values and then be used to extrapolate a prediction of their likely future values. Examples of time series based methodologies are:

2.2.1.1.1 Exponential smoothing:

Simple exponential smoothing is a method of forecasting used in a variety of applications, [107,109,144] it assumes that the average level of demand is stationary or only changing very slowly. The process can be modelled as:

$$x_t = \bar{x}_t + e_t \tag{2.2.1.1}$$

where x_t is the actual value of the demand at time t, $\bar{x}_t$ is the expected value of the demand and e_t is a random error component having zero mean. Given a time series of past demand data $x_1, x_2, \ldots, x_t$ an estimate of the value of $\bar{x}_t$ at time t can be calculated as an exponentially weighted average:

$$\bar{x}_t = ax_t + a(1-a)x_{t-1} + a(1-a)^2 x_{t-2} + a(1-a)^3 x_{t-3} + \ldots \tag{2.2.1.2}$$

Substituting (t - 1) for t and multiplying through by $(1-a)$ yields

$$(1-a)\bar{x}_{t-1} = a(1-a)x_{t-1} + a(1-a)^2 x_{t-2} + a(1-a)^3 x_{t-3} + \ldots \tag{2.2.1.3}$$

Subtracting equation 2.2.1.2 from equation 2.2.1.3 gives

$$\bar{x}_t = ax_t + (1-a)\bar{x}_{t-1} \tag{2.2.1.4}$$

Where a is the smoothing constant, having a value between 0 and 1.

If the current trend T_t of the smoothed values is calculated then an estimate of the future smoothed values can be made

$$T_t = C(\bar{x}_t - \bar{x}_{t-1}) + (1-C)T_{t-1} \tag{2.2.1.5}$$

$$\bar{x}_{t+1} = \bar{x}_t + T_t \tag{2.2.1.6}$$

Where C is a constant with a value between 0 and 1.

Exponential smoothing is a procedure which adjusts the estimate of the new smoothed value by an amount proportional to the most recent forecast error. The simple exponential smoothing equation can be augmented by the inclusion of factors that are designed to account for trends in the data such as seasonal effects [28,109]. It should be noted that the values chosen for constants such as α have a direct effect upon the sensitivity of the forecast results to sudden changes in the data. Lower values for α produce a forecast that is more influenced by data further back in the time series and less affected by sudden random changes in the more recent data. Conversely higher values of α produce a faster response to the more recent changes in the data. An application dependent compromise has to be found between stability and sensitivity of the algorithm.

2.2.1.1.2 Discussion on Exponential Smoothing

Exponential smoothing has the advantages of requiring little computing time and not requiring the storage of large amounts of past data. It is also relatively easy to adjust the sensitivity of the equation to changes in the process being modelled. The chief disadvantage in terms of application to water systems demand prediction being that the prediction period is typically very short, ranging from minutes ahead up to two hours ahead. The problem being that values chosen for the smoothing constant and any seasonal and/or trend constants do not remain valid over the 24 period of a typical demand profile. The short prediction period is not a problem for some applications in electrical load forecasting [107], the much faster response times involved in plant switching means that forecasts of minutes ahead are required to allow operators to react within the short prediction horizon.

2.2.1.1.3 Spectral analysis/expansion

Most power generation/distribution utilities or companies maintain records of past electrical load data, however they are less likely to keep records of contemporaneous meteorological data. It is therefore desireable to have a method of prediction that only requires past load values for the prediction process i.e. it is not necessary to obtain historical records of weather data. Such a method of prediction based on spectral expansion was first proposed for use in the power supply industry by Farmer [48] in 1963 and involves the dissagregation of the load into a base load and a weather dependent component.

The basic premise of the spectral expansion methodology is the division of the load into a slowly varying base component and a residual component that can be attributed to variations in meteorological conditions. In order to generate a prediction, it is assumed that the level of this weather related component will not change significantly over the prediction period. The base load is calculated and then the current consumer response to the existing weather conditions is identified and extrapolated to produce the prediction.

Mathewman and Nicholson [100] proposed that since the load pattern repeats itself every 24 hours it is possible to consider the time series for each day, whether continuous or discrete, as being a member of an ensemble of time series. The problem then becomes one of predicting a nonstationary process, given an ensemble of sample functions. The daily load curve can be divided into overlapping part day periods of between 4 and 24 hours duration. By taking the part day load curves for a particular period over a number of days, it is possible to define the value of the load on day m at time t as x_{mt} where:

$$x_{mt} = \alpha_{mt} + f_1(T_m)\beta_{mt} + f_2(L_m)\gamma_{mt} + f_3(W_m)\zeta_{mt} \qquad (2.2.1.8)$$

In the above equation $f_1(T_m), f_2(L_m), f_3(W_m)$ are functions of mean temperature, illumination and wind speed respectively. For the duration of the period under consideration these functions are considered to be linear. The quantity α_{mt} represents the base load and the factors $\beta_{mt}, \gamma_{mt}, \zeta_{mt}$ account for the varying importance of these individual weather components with the time of day. Each load vector is therefore linearly dependent upon the vectors $\alpha, \beta, \gamma, \zeta$.

A method of calculating the values of the weather dependent component without specific reference to the actual meteorological data is provided by use of Karhunen's spectral expansion of stochastic processes [83]. For a part day period made up of N discrete values taken from M example past days it is possible to set up a K dimensional linear manifold in which Karhunen's spectral expansion can be written:

$$x_{mt} = \sum_{k=1}^{K} a_{mk} \lambda_k^{\frac{1}{2}} \phi_{kt} + e_{mt} \qquad t = 1, 2, \ldots, N \qquad (2.2.1.9)$$

Where e_{mt} is the error in the predicted value. Equation 2.2.1.9 can be expressed in matrix notation as:

$$\mathbf{X} = \mathbf{C}\Lambda^{\frac{1}{2}}\Phi + \Delta \qquad (2.2.1.10)$$

Where:

$$\begin{aligned}
x_{mt} &\in \mathbf{X}; \dim(\mathbf{X}) = M \times N \\
a_{mk} &\in \mathbf{C}; \dim(\mathbf{C}) = M \times K \\
\phi_{kt} &\in \Phi; \dim(\Phi) = K \times N \\
e_{mt} &\in \Delta; \dim(\Delta) = M \times N \\
diag\left(\Lambda^{\frac{1}{2}}\right) &= \left[\lambda_{11}^{\frac{1}{2}}, \lambda_{22}^{\frac{1}{2}}, \ldots . \lambda_{kk}^{\frac{1}{2}}, \ldots ., \lambda_{KK}^{\frac{1}{2}}\right]
\end{aligned}$$

From the above form of the expansion it is possible [100] to derive a matrix eigenvector equation which when solved yields eigenvectors which are equivalent to the vectors $\alpha, \beta, \gamma, \zeta$ of equation 2.2.1.8.

Various methods have been proposed for finding the set of coefficients a_k including the approach described by Farmer [50] which is to calculate the most probable value of the coefficients using the method of conditional probability outlined by Kalman [81].

2.2.1.1.4 Discussion on Spectral Expansion

The main advantage of this method of prediction is that recognition is made of the fact that weather conditions have an impact on the consumption of electricity and a way of accounting for this impact is made without the requirement for large volumes of past meteorological data . In addition, some of the derived versions of the methodology [144] are relatively computationally efficient, requiring only the first and/or second eigenvalues to be calculated. The disadvantage with the methodology is that it assumes a static relationship between the weather related factors and the load over the prediction period. This is not a problem as long as weather conditions remain the same during the prediction day, but if sudden changes do occur in the meteorological conditions then serious errors can be introduced. In addition, abnormal days such as bank holidays are not accounted for unless a separate model is derived for each occurrence.

2.2.1.1.5 ARMA and ARIMA Time Series Models

A brief overview of this class of time series model and their application to load forecasting is given in this section. Because an ARIMA prediction algorithm is used to provide comparison results for the accuracy assessment of the proposed new prediction methodologies described within this thesis, a more detailed study of this algorithm is provided in the following chapter.

The use of time series methods such as exponential smoothing assumes that an element of the time series in question will consist of the mean of the series plus a random error component. However, there are many examples of time series where the elements of the series are clearly not independent and forecasting techniques have been developed which are designed to exploit the dependency displayed by time series observations. Such techniques were described by Box and Jenkins [17] and are known as Auto Regressive Moving Average (ARMA) techniques.

The variations displayed by a time series of past electrical load data can be thought of as the result of a number of cyclic processes acting upon the data, plus a random non deterministic element. If all the cyclic processes can be correctly identified and modelled then the residual noise remaining after their removal should display a Gaussian distribution. The general form of the ARMA models used in the prediction of electrical load is given below.

$$\phi(B)\Phi\left(B^{s}\right)W_{t} = \theta(B)\Theta\left(B^{s}\right)a_{t} \tag{2.2.1.12}$$

Where a_t is a Gaussian noise sequence W_t is a stationary time series of load data obtained by the transformation:

$$W_{t} = \nabla^{d}\nabla_{S}^{D} Z_{t} \tag{2.2.1.13}$$

Where $Z_t(t = 1,2,\dots,N)$ is the non stationary time series,

∇ the backward difference operator

$$\begin{aligned}\nabla_{s} Z_{t} &= \left(1 - B^{s}\right)Z_{t}\\ &= Z_{t} - Z_{t-s}\end{aligned}$$

d and D the difference orders

$$\nabla_s^D Z_t = \left(1 - B^s\right)^D Z_t$$

$$\nabla^d = \left(1 - B\right)^d$$

B the backward shift operator

$$BZ_t = Z_{t-1}$$

B^S the seasonal backshift operator of period S

$$B_S Z_t = Z_{t-S}$$

The Auto regressive components are:

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p \qquad (2.2.1.14)$$

$$\Phi\left(B^s\right) = 1 - \Phi_1 B^s - \Phi_2 B^{2s} - \ldots - \Phi_p B^{ps} \qquad (2.2.1.15)$$

Where p is the order of the AR component and P is the seasonal difference order.

The Moving Average components are

$$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \ldots - \theta_q B^q \qquad (2.2.1.16)$$

$$\Theta\left(B^s\right) = 1 - \Theta_1 B^s - \Theta_2 B^{2s} - \ldots - \Theta_q B^{qs} \qquad (2.2.1.17)$$

The general form of the model given here is modified to a form that is relevant to the actual time series data being used. Statistical analysis methods such as sample autocorrellation are used to identify the cyclic periods operating upon the data and hence the significant ARMA components. The fitting of the specific form of the model to the data requires determination of the values of these ARMA components such that the sum of the squares of the residuals due to the parameter fit are minimised.

Several hill climbing procedures have been used to achieve this minimisation [46,57,90,118,139], however a computationally efficient method is described by Sterling and Bargiela[147] based on a derivative of the Newton Raphson iterative procedure. This uses an approximation to the Hessian matrix calculated by the Davidon, Fletcher, Powell[146] method, which provides additional information on the state of the function to be minimised in the current search direction and avoids the possible problems of non convergence often associated with complex non-linear problems of this type.

A prediction is produced from the fitted model by expanding the model forward in time for the length of the required prediction and assuming that the values of the noise series a_t are equal to zero over this prediction period.

2.2.1.1.6 Discussion on ARIMA and ARMA Methods

ARIMA algorithms have been shown [90,118,146] to be capable of accurately modelling the cyclic patterns displayed in a typical electrical load time series as long as the series remains stable or only changes slowly. The methodology will successfully track the slow changes in consumption due to the general weather patterns of the seasons, however it cannot take account of the more sudden changes in load due to the day to day variation in weather conditions. This is particularly true during the transitional seasons such as spring where marked fluctuations in the weather conditions often occur and hence large differences in electrical consumption are common. Hagan [63] developed a multivariate ARIMA model that included the most influential weather factor upon load, namely the temperature of the prediction day. However, little improvement was found in the accuracy of results achieved, this probably being due to the non-existence of a constant relationship between variations in temperature and the corresponding changes in load.

The results produced by the ARIMA algorithm are also distorted by the occurrence of abnormal days such as Bank Holidays, not only in terms of the reduction in prediction accuracy for the particular abnormal day in question, but also in terms of the distorting effect of the presence of the abnormal day within the past data that is used to produce future predictions.

2.2.1.2 Causal Methods

Causal approaches seek to explain the variations in a time series of electrical load data by examination of the variations in one or more causal variables, in the case of the load forecasting examples described below, it is the weather variables that are deemed significant in determining the variations in the time series [24,46].

2.2.1.2.1 Weather Weighting

The method of weather weighting described by Dryar [44] was an early attempt at describing in mathematical terms the effect weather conditions have upon load. The technique involves the division of the load into two components, a base load and a weather dependent load. The value for the base load is fixed at a level determined from past data by removing the estimated effects of the weather conditions at the time the readings were taken. The correct estimates of these past weather influences are determined by trial and error as shown below, the significant weather factors chosen for this application are temperature, cloud cover and wind speed. Tables are constructed so that for a given time of year and a given set of values for temperature, cloudiness and wind speed, the percentage weighting to be applied to the base load to account for these weather conditions is derived. An example of such a table for determining the weighting values for different temperatures at different times of year is shown in table 2.2.1 reproduced from [44].

Figure 2.2.1 Weight Values for Load Prediction.

Weights	Dec Jan Feb	Apr Nov	May Oct	Jun Sept	Jul Aug
%	Deg F	Deg F	Deg F	Deg F	Deg F
10	15	25	35	95	100
8	20	30	40	90	95
6	25	35	50 80	85	90
4	30	40	55 75	50 80	85
2	35	45	60 70	55 75	80
0	40	50	65	60 70	75
-2	45	55		65	70
-4	50	60			65
-6	55	65			
-8	60				

Once an estimate of the base load has been made for the prediction period, a forecast of the weather conditions for the coming 24 hours is used to generate an estimate of the weather dependent load via reference to the weather weighting tables. Re-estimates can be made of the weather weightings as more accurate weather forecasts become available during the prediction period.

2.2.1.2.2 Discussion on Weather Weighting

The accuracy achievable by a method such as weather weighting is dependent on a number of factors. Firstly, the consistency of the base load profiles upon which the weather weights are superimposed is essential if the predictions produced are to reflect the true load values. For example the peak load needs to occur at the same point in the day for each week day and have roughly the same magnitude (when corrected for the weather conditions at the time of the reading). However, the base load consumption does change with the seasons and this can lead to problems, particularly for Saturdays and Sundays where, due to the need to use a number of past examples of such days to generate their base load profiles, data from several weeks in the past may need to be used. This data may not reflect the current consumption characteristics. In addition the choice of the correct weather weightings to apply at the time of the forecast is heavily dependent upon the accuracy of the prediction of particular weather factors such as temperature i.e. If the temperature forecast is wrong by even a small number of degrees then this may result in the wrong weather weightings for temperature being used and hence an inaccurate forecast being generated.

2.2.1.2.3 Linear Regression.

A distinction is made in this method between the base load and the weather sensitive load. The relationship between the weather sensitive load and the weather conditions can be expressed explicitly in the form of explanatory variables by a multiple linear regression method [44,62,97]. The explanatory variables are selected on the basis of correlation analysis of the time series in question. In the application poposed by Davies[38] the meteorological factors chosen as significant are temperature T, wind speed W, illumination L and the rate of precipitation P. A regression equation of the form shown below can be fitted to the data:

$$x = a + b_1 T + b_2 W + b_3 L + b_4 P + F(t) + d \qquad (2.2.1.17)$$

Where x is the demand at a particular time of day. T, W, L, P correspond to the significant meteorological factors temperature, wind speed, illumination index and precipitation rate. b_1, b_2, b_4, b_5 are the regression coefficients of these meteorological factors determined using least squares estimation based on past load and weather data and a is a constant, d is the day of the week correction and $F(t)$ is a polynomial function of the time of year for a particular week and accounts for variations in the base load with the time of year, thus $a + F(t)$ is the base load at week t.

2.2.1.2.4 Discussion on Linear Regression

Although the linear regression methodology provides a computationally simple way of generating load forecasts, the major drawback with the method is the assumption of a steady linear relationship between the values of certain meteorological factors and the corresponding electrical load consumption. In reality the relationship is highly non- linear and changes from week to week and month to month and therefore imposing a linear relationship in order to calculate future load values will lead to the introduction of significant errors.

2.2.1.3 Combined Time Series and Causal Methods

Prediction methodologies have been proposed that aim to combine within a time series approach, causal elements relating to the influence upon load of weather conditions. Bolzern and Fronza [15] carried out a study in the Milan area of Italy to evaluate the advantages of including weather factor values within an ARMA based prediction algorithm, the resulting algorithm being termed ARMAX , the X representing the additional exogenous inputs. The form of the ARMAX model used is as follows:

The standardised load variable is

$$Z_k(i) = \frac{(y_k(i) - \mu_i)}{\sigma_i} \tag{2.2.1.18}$$

Where μ is the mean and σ is the load standard deviation in the ith interval of the day.

The ARMAX model is:

$$z_k(i+1) = \sum_{j=1}^{p} \phi_j z_k(i-j+1) + a_k(i+1) + \sum_{j=1}^{q} \theta_j a_k(i-i+1) + \zeta_T g_T\left[T_k^M(i+1)\right] + \zeta_L g_L\left[L_k(i+1)\right] \tag{2.2.1.19}$$

Where $a_k(i)$ is assumed to be a white noise sequence, $T_k^M(i)$ is the average of the temperature in the M hours before the$(i+1)$ interval. $L_k(i)$ is the average illumination in the (i)th interval. The model weighting parameters are $\phi_j, \theta_j, \zeta_T, \zeta_L$ and p, q are the model orders determined from the past data.

The method is applied to weekday and weekend data separately in order to account for the weekly load cycle, furthermore, the temperature and illumination exogenous inputs are set to zero during the night. The weather data was provided by a number of local meteorological stations and data over a two year period was used in the assessment of the ARMAX predictor. The results gained from the testing of both the standard ARMA predictor and the ARMAX predictor showed that although a small improvement in prediction accuracy was achieved the authors considered that this improvement may not be large enough to justify the additional effort in collecting the necessary weather data. It should be noted that this weather data collection was done manually, more recently on line computer records of this type of data have become available.

2.2.2 Heuristic Approaches.

Heuristic methods of load prediction have generally been developed in an attempt to overcome the problems the time series based approaches have in accounting for the effects of weather variations and abnormal day occurrences. The areas of study include pattern matching techniques, the use of expert systems to improve prediction performance, load dissagregation and the implementation of neural networks in the generation of predictions. This latter category will be discussed in a later chapter of this thesis.

2.2.2.1 Pattern Matching

Such techniques are very useful when the relationships between a process and the measurable causal variables are complex and not well enough understood to enable them to be successfully modelled mathematically. The behaviour of the pattern of electrical load in response to meteorological variation is just such a problem. The basic assumption being that if a load demand has exhibited a particular pattern in response to a certain set of weather conditions, then a similar type of demand pattern will result if the same type of weather conditions occur again.

Mathewman and Nicholson[100] proposed a pattern recognition technique based on the division of similar daily demand profiles into 'clusters' so that each cluster contains profiles that display closely related characteristics and each cluster is sufficiently different from every other cluster as to be uniquely distinguishable. The number of clusters that are required is determined by the accuracy of prediction that is to be achieved, and to avoid overloading the computer memory, only representative examples of each category are stored, these are termed the locates of the categories. The decision to classify a particular input load into a particular category is based on a minimum distance classification between the locates of the various categories. Here the locus of the points representing the load data equidistance from the nearest member of the two adjoining classes forms the decision boundary .

The categories were set up during an initial training phase using CEGB data, sampled at half hourly intervals, twenty clusters in all were established. However, problems were encountered in the correct classification of input load profiles with consequent introduction of unacceptable input errors. The research revealed that the cause of the errors lay in the fact that clusters were not behaving as clearly defined regions i.e. the level of variation in the daily profiles was too great for a consistent classification to be maintained.

A different approach to pattern recognition forecasting based on the matching of present meteorological conditions with examples from past data. The ALFA (Automatic Load Forecasting Assistant) [78] employs a database of fifteen years past load data and meteorological data to develop a forecasting system. Firstly, a base load for each day of a whole year was calculated using an average of all the examples of the relevant days of the week drawn from the database of past load profiles. The weather dependent load for the prediction day is then determined by pattern matching the forecast values for temperature, humidity, wind speed and cloud cover with the eight closest examples of such weather conditions found in the past meteorological data. The corresponding weather dependent load profiles for each matched example are extracted and averaged, the resulting weather dependent load profile is added to the appropriate base profile to produce a prediction. The search for days with similar weather conditions is restricted to the same season in which the prediction day falls.

The results from this application in the Eastern USA were a significant improvement over a conventional ARIMA based prediction system, however the methodology relies on a very large sample of past load data and meteorological data being available, and this is not the case in many situations, this is particularly true with water consumption data.

2.2.2.2 Expert System Applications

An expert system is a system which can store and apply heuristic knowledge to a problem in a way that is similar to a human expert. Most commonly the knowledge is encoded in the form of conditional rules and the control over the firing of these rules in response to input data is controlled by an inference engine. More details on the various types of expert system that have been developed and the way they operate are given in Chapter 4 of this thesis.

The application of expert systems in the field of electrical load prediction [43] can be divided into those systems which provide guidance on the correct type and form of mathematical prediction algorithm to be used, and those systems that use heuristic knowledge to modify the predictions produced by mathematical algorithms.

An example of a system that has been developed to provide advice on algorithm choice is that described by Pratt [119]. This application uses a rule based expert system to determine which one out of a possible eight ARIMA based prediction models is best suited to the time series under consideration. Initially the set of eight possible ARIMA models are selected by an operator who bases this choice on the characteristics displayed by the time series under consideration. The expert system determines which of the eight is the best fit to the particular time series that is presented, this being done by the application of a number of rule encoded criteria covering such areas as convergence, the distribution of the residuals, parameter values lying outside the range -1 to +1 (thus indicating a poor fit of the model to the data). Similar work has been conducted by Singh et al [136] where the use of an expert system was proposed to determine the most appropriate form of time series prediction in terms of the model order, the amount of past data and the sampling rate. However no results were presented in this paper so evaluation of its importance is not possible.

The use of expert systems in the modification of predictions produced by mathematical techniques has been a fairly recent development. Remoir and Ayuso[127] proposed a methodology that involved the use of a Box Jenkins[13] based algorithm to produce both weekly and daily load predictions in the form of a total load figure for each day. This was then spread out into a 24 hour profile by the application of templates representing the typical types of load profile encountered for the day in question. The expert system was developed using DEC's OPS-5 language and was designed to apply rules that determined the profile template that was most appropriate for the prediction day. Rules could also be invoked to carry out modifications to this template to account for the occurrence of such events as public holidays, night irrigation etc. This being achieved by the application of correction factors to the prediction values for the duration of the effect in question.

Rahman and Bhatnagar[124] proposed a temperature based method of calculating a base forecast which is then modified by rules relating to the season in which the prediction day falls, the prevailing weather conditions on the prediction day and those of the preceding days i.e. two consecutive hot days will produce a greater impact upon the load curve than a single hot day. The base forecast is calculated by predicting the temperature for each hour of the prediction day and matching this temperature profile with a database of past temperature and load data. The load profiles of the three closest days, in temperature terms, to the prediction day are pulled out and averaged to give the predicted base load profile. In a more recent development on this work Rahman[122] proposes a priority vector based technique that uses a pattern matching algorithm to derive a set of similar days whose type (day of week, season etc.) and temperature profile match that of the current day.

In Taiwan, an expert system has been developed by Ho, Hsu et al [71] that uses rules encoded in PROLOG. Initially a five year database of hourly load data was subjected to a simple pattern matching algorithm in order to identify the number of different types of day present in the database. In all, eleven day types were identified, examples of which are weekdays, Saturdays, Sundays, Chinese New Year etc. Typical profiles are available for each of the day types, the task of the expert system is to aid the operators, via a question and answer session, in the determination of the correct day type for the prediction day. The results achieved are shown to be an improvement upon the results from a regression model used on the same data, this is particularly so on abnormal days.

A related field is the use of knowledge based load dissagregation [73] to generate load forecasts that can account for abnormal day types. The basis of such methodologies is to break down the daily load profile into components that can be identified as being related to specific uses or causes. Dissagregation can range from fairly coarse subdivisions of the load into industrial, domestic and commercial components, to more detailed divisions into lighting load, cooking load, domestic heating etc. Li [95] describes a system where the load is dissagregated in a pyramidal fashion where the initial coarse divisions into industrial, domestic and commercial load are each further split down into more detailed components such as heating, lighting and refrigeration. Each component has its own daily load profile, some of which will be weather dependent, others will be dependent on factors such as shift work etc. The profile shapes of the identified components are calculated from past data and are described using a collection of standard curve types, ramp, slope, constant etc. To generate a prediction for a particular day all the profiles of the components considered to be present on the day in question are summed to give a total load profile. This allows the simple inclusion of components that are designed to account for abnormal events or effects taking place on the prediction day.

Two problems arise with dissagregation methodologies, firstly the identification of the possible components that are present within a daily load profile requires a detailed knowledge of the social, commercial and industrial profile of the geographical area under investigation. Secondly the attempted allocation of the correct magnitudes to each dissagregated component is a potential source of significant error unless very comprehensive measurement and metering studies are undertaken. Although these allocation errors may cancel out in the production of a final prediction, there is also the potential for such errors to be compounded i.e. a large number of components with values that have been over-estimated may occur on a particular prediction day.

2.2.3 Summary of Electrical Load Forecasting

The investigation and analysis of the various methodologies that have been applied to the field of short term electrical load prediction provides an insight into the problems faced when an attempt is made to model a process that is governed by the behavior patterns of individuals within an industrialised society. Many mathematical quantitative approaches provide acceptable prediction performance when applied to data exhibiting the stable cyclic daily and weekly patterns of normal electricity consumption. However, when this stable pattern is distorted by the influence of external factors that have a highly non-linear effect upon the level of electricity consumption, the prediction accuracy of such mathematical methodologies is compromised.

Heuristic approaches attempt to use heuristic knowledge to either augment or replace the mathematical prediction techniques. In doing this they are aiming to mimic the reasoning processes commonly carried out by operators who have for a long time been using their knowledge concerning the nature of electrical consumption to formulate their own predictions of the likely variations in load. If this can be done successfully, then this has the advantage of a standardisation of performance, in that the same result will be produced in response to the same input criteria, which may not be the case with human operatives. Also, once input into the system, the knowledge is always available and is not lost when an operator leaves the company. Problems that arise from a heuristic approach to load forecasting are the initial acquisition and organisation of the knowledge to be exploited and the task of ensuring the knowledge contained within the system remains valid.

Because of the similarities between the fields of electrical load prediction and water demand prediction, the above analysis of the load prediction methodologies provides a suitable background for the assessment of the techniques that have been applied to water demand prediction.

2.3 Water Demand Prediction Methodologies.

Although the field of water demand prediction in supply and distribution networks has previously experienced a lower level of research investment than has been the case with electrical load forecasting, recent improvements in data collection and water network control technology have led to a more widespread realisation of the importance of demand forecasting. The ability to observe and control the behaviour of the network via a telemetry system which supplies monitored flow and pressure data to a central computer, has opened up the possibilities of achieving optimal network operation. One of the key requirements of a system designed to make such optimal operation possible is the availability of sufficiently accurate short term demand predictions on which to base the calculations of possible least cost pumping schedules.

The calculation of the volume of water to be pumped in a given period should be based on an estimate of the likely consumption to be met during that period plus a volume required to provide a calculated degree of supply security. Without the ability to monitor and record the time varying behaviour of the water network it is impossible to make an accurate assessment of the security of supply requirements. Hence, prior to the widespread installation of telemetry systems it was common practice within the water industry to maintain reservoir levels at near their maximum no matter what the supply situation. Telemetry has provided the means whereby reliable risk assessments can be made and the necessary security of supply requirements calculated.

The availability of telemetry derived data has also led to the development of improved methods for the prediction of future consumption. Previously, the most common method of producing demand predictions for the forthcoming day was simply to use the profile either of the previous day or of the same day the previous week. The major drawback with such a method is the lack of consideration of the underlying factors which determine the level of demand on a particular day. The set of meteorological and social circumstances that combined to produce the observed consumption on one day may not be applicable to the day for which the prediction is required.

Of the more sophisticated methodologies that have been applied to water demand prediction, many are direct developments of work conducted initially into electrical load prediction. Like electrical load prediction, a subdivision of methodologies can be made into quantitative and heuristic approaches.

2.3.1 Quantitative Approaches

2.3.1.1 Time Series Based Methods

There are several examples of time series based approaches to water demand forecasting, each utilising the records of past demand data to generate predictions.

2.3.1.1.1 Exponential Smoothing.

Coulbeck, Tennent and Orr [35] developed an exponential smoothing based demand predictor for use in conjunction with the GINAS [34] water network analysis package. The program included routines for screening and smoothing the raw telemetry data prior to its submission to the forecaster. The screening of the data is designed to remove gross errors and is based on second order differencing while the smoothing operation is carried out in order to remove the small amplitude random errors present in the data and uses frequency thresholds based on Fourier analysis.

Statistical tests are conducted upon the past data in order to establish categories of days according to the similarity of their demand patterns, the prediction is then based on the extrapolation from the most recent data of the same day type category as the prediction day. The work carried out by Moss [106] indicated that triple exponential smoothing could provide good results in extrapolating a future profile from the current profile.

The vector of prediction errors of the current daily or weekly period is defined by:

$$\mathbf{e}_t = \hat{\mathbf{x}}_{t-1}(1) - \mathbf{x}_t \tag{2.3.1.1}$$

Where $\mathbf{x}_t$ is the vector of data sample values at the current period, and $\hat{\mathbf{x}}_{t-1}(1)$ is the 1 period ahead forecast. When the current demand profile becomes available, the error vector can be used to correct trend estimates according to:

$$\hat{\mathbf{a}}_t = \mathbf{x}_t + (1-w)^3 \mathbf{e}_t \tag{2.3.1.2}$$

$$\hat{\mathbf{b}}_t = \hat{\mathbf{b}}_{t-1} + \hat{\mathbf{c}}_{t-1} - 1.5w^2(2-w)\mathbf{e}_t \tag{2.3.1.3}$$

$$\hat{\mathbf{c}}_t = \hat{\mathbf{c}}_{t-1} - w^3\mathbf{e}_t \tag{2.3.1.4}$$

Where w is a smoothing parameter with a typical value set to 0.1 and $\hat{\mathbf{a}},\hat{\mathbf{b}}$ and $\hat{\mathbf{c}}$ are estimates of position, velocity and acceleration trend components at periods t and $t-1$. Initialisation procedures set the value of $\hat{\mathbf{a}}_{t-1}$ equal to that of $\mathbf{x}_{t-1}$ and the values of $\hat{\mathbf{b}}_{t-1}$ and $\hat{\mathbf{c}}_{t-1}$ to zero. The 1 period ahead prediction is then given by:

$$\hat{\mathbf{x}}(1) = \hat{\mathbf{a}}_t + \hat{\mathbf{b}}_t + 0.5\hat{\mathbf{c}}_t \tag{2.3.1.5}$$

2.3.1.1.2 Discussion on Exponential Smoothing.

A possible problem arises with the data smoothing and filtering operations described in this methodology, in that although truly erroneous data may be successfully removed from the past data, there is a corresponding risk that data is removed that reflects actual abnormal demand events. Great care has to be exercised in the choice of frequency thresholds, significant harmonics etc. in order to avoid the loss of true data. Although the exponential smoothing trend components will follow the relatively gradual changes in the external influence on demand of factors such as weather, more rapid changes in these external factors such as the sudden change in weather conditions brought on by the arrival of a frontal system after a period of stable anticyclonic weather will cause prediction accuracy to be reduced.

2.3.1.1.3 Spectral Expansion.

Sterling and Antcliffe [146] in the early seventies proposed the use of a spectral expansion technique very similar to that used by Farmer [48] (described in the previous section on electrical load forecasting), to produce predictions of total daily water consumption based on several previous years daily totals. The results of this application were directly compared to the results from both a manual 'best guess' technique that was previously employed by the water utility which supplied the data, and a linear regression technique whereby the best straight line, in a least squares sense, was fitted to the past data. The spectral expansion results compared favourably with the results from the other two methods only after the data submitted to the spectral expansion algorithm was pre filtered by linear regression to remove the monthly trend from the data.

Perry [116] in 1981 described the application of both a spectral expansion technique and a Kalman filter based technique to water demand forecasting. Comparisons were made between the two methodologies in terms of the accuracy of prediction over a number of prediction horizons. The spectral expansion technique provided significantly more accurate predictions over very short prediction horizons (up to 4 hours ahead) however in terms of overall performance over 24 hour predictions there was little difference between the two methodologies, average RMS errors being in the region of 3 to 5 percent.

2.3.1.1.4 Discussion on Spectral Expansion.

The comparison carried out by Perry showed the advantages associated with the spectral expansion technique for water demand prediction. A good performance can be achieved in terms of accuracy for very short term predictions, the only data requirements are for past consumption data and do not require meteorological measurements and the relatively low computing overheads are an advantage for on-line implementation. The main disadvantage with the technique is that it only models a static relationship between the consumption and the external influencing factors. This does not reflect the true situation where there exists a constantly varying relationship between factors such as weather and the corresponding level of demand. The methodologies described by Sterling et al and Perry take no account of the occurrence of abnormal demand days such as holidays.

2.3.1.1.5 Auto Regressive Moving Average Models.

In a continuation from the work carried out on the application of spectral expansion to demand forecasting, Sterling and Bargeila [144] proposed a water demand forecaster based on an ARMA model developed from the work of Box and Jenkins [13] in the field of electrical load prediction. In order to use the ARMA model the time series of past demand data must be transformed into a stationary series, this transformation being achieved by a differencing operation based on the periodicity of the data identified from the sample auto correlation function. The general form of the ARMA model is as given in equation 2.2.1.12 . Statistical analysis of the correlations between elements of the time series of past data is carried out to determine which of the auto regressive and moving average components are significant in the time series under test. Values for these significant components are determined by a Newton Raphson minimisation of the sum of the squared errors of the noise series a_t . The model is expanded forward in time to produce a prediction by assuming the error values in the noise series a_t over the prediction period are equal to zero.

Jowitt and Xu [80] proposed a prediction methodology that utilised an ARIMA algorithm to predict daily totals of consumption and then distributed the predicted total into a 24 hour profile via the application of standard template profiles. Different templates are available for different prediction days i.e. Summer Saturday, Winter weekday etc. The advantage of this method is that it does not require hourly or half hourly consumption data records which may not be available to the water utility, whereas all water utilities are likely to have records of the total amount of water supplied each day.

Recent work by Shamir, Shartser and Feldman [133] in Israel utilises a novel combination of pattern recognition techniques and an ARIMA algorithm. They propose that a typical 24 hour demand profile is composed of distinct segments or 'states' that are termed rising, falling and oscillating. The points of transition between these states are identified by pattern recognition and appropriate Auto Regressive time series models are constructed for each resulting segment. The accuracy of the predictions achieved by this method were in the range 6% to 11%, with testing being conducted over one month (July). Meteorological influences were assumed to be constant over the test period, although future work is aimed at incorporating weather effects into the system.

Other developments in the use of Auto regressive Moving Average models for demand forecasting have been the work carried out by Quevedo et al [120] in Spain and by Steiner [142,143] in the USA. Both these groups investigated the effects on prediction accuracy of including exogenous weather variables in ARIMA and ARMA models. Quevedo used intervention analysis to incorporate the influences of special holidays into an ARIMA model and then added a transfer function of temperature which was determined to be the most significant weather factor in influencing demand. The form of the general model is shown below:

$$Z_t = V_0 T_t + \sum_{i=1}^{5} w_i I_i + w_a I_a + w_e I_e + \frac{\left(1-\theta_7 B^7\right)}{\left(1-\phi_1 B - \phi_7 B^7\right)\left(1-B^7\right)} \qquad (2.3.1.7)$$

Where Z_t is the demand, I_i, I_a and I_e are the weighted intervention variables relating to five annual public holidays, all days in the month of August and the Easter period respectively, their values being set to 1 when they are considered to be in operation. T_t is the temperature at time t and V_0 is determined to be the significant coefficient of the transfer function $\frac{\omega(B)}{\delta(B)}$ when expressed as a polynomial:

$$V(B) = V_0 + V_1 B \ldots$$

The determination of the significant coefficients of the $V(B)$ polynomial being conducted by estimating a number of such coefficients in a model:

$$Z_t = \left(V_0 + V_1 B \ldots V_k B^k\right) X_t + N_t$$

Where Z_t is the output time series, X_t is the input time series and N_t is the part of Z not accounted for in terms of X.

2.3.1.1.6 Discussion on Auto Regressive Moving Average Models.

The results from the study by Quevedo on an area of the Barcelona water supply system showed that the inclusion of the temperature transfer function actually had a negative effect upon the accuracy of results, this being due to errors in temperature prediction and the changing effect of temperature upon demand through the seasons. Predictions for the whole of 1986 were made with the daily demand prediction errors ranging up to 18% and the average around 5 to 6%. Also important in this study was that the data entered into the database for use in generating the predictions, was filtered to remove anomalous data that could have a detrimental effect on future predictions, the threshold levels of the filter being set by the operator. In the application described by Steiner the influence of the weather factors such as temperature and the number of antecedent dry days were taken into account in predicting daily demand totals by removing their estimated effect from the past data stream by fitting multivariate regressions to the data.

The standard ARIMA and ARMA methodologies provide a method of accounting for the periodic variations and trends displayed by a stable water demand time series. The attempts to modify these techniques so that they are able to also account for the effects of external influences such as holidays and weather have had varying success. The inclusion of weighted intervention variables to account for special days through the year has the possible drawback of incorrectly assuming the effect of the holiday will be the same as the same day the previous year. The problems involved with the inclusion of the values of weather variables into the demand calculation is highlighted by the results of the Barcelona study, where the errors introduced through predicting the temperature and the changing nature of relationship between water use and temperature made the prediction accuracy worse.

2.3.2 Heuristic Methods.

There have been relatively few studies conducted into the application of heuristic approaches to water demand forecasting. Studies that have been carried out are of the dissagregation type, which aim to identify and assign values to a number of different water use categories. Reasearch was conducted in Portugal [32] such that the profiles that link the social composition of an area with particular demand characteristics were established by means of extensive consumer surveys, the examination of existing statistics and the installation of district flow meters. The aim of the project was to develop a method that would allow the demand profile to be determined from the urban characteristics of an area. However, this paper was an outline of a proposed system and no results were supplied and hence evaluation of this technique was not possible.

Boland and Dziegielewski [14] carried out a study on urban water use in the USA, producing a highly dissagregated model of water consumption, the IWR-MAIN model, that involves hundreds of different categories of usage. Divisions of a water supply area's urban composition are structured into high level divisions such as residential, commercial and industrial water use, and lower subdivisions into categories such as 'metered sewered single family residences'. Weighting factors are introduced to account for effects such as, the relative increase in residential water use with increasing house value, the effect of the price of water upon consumption (significant in the USA) and the effect of weather influences. The consumption value of each of the categories is estimated from the demographic make up of the area under investigation and they are all added together to produce a total demand figure. However, in a paper published in 1990 Wilson and Luke [160] levelled serious criticisms against the methodology and results of the IWR-MAIN model. The model is criticised as being flawed owing to a number of major faults, the most significant of these being that knowledge of the actual levels of water use for different purposes is not detailed enough to justify the degree of dissagregation used in IWR-MAIN. Wilson and Luke maintain that the model's ultimate accuracy depends on whether, through chance or manipulation, the errors of individual consumption estimates are self cancelling.

2.3.3 Summary of Water Demand Prediction

Most of the work conducted in the field of water demand forecasting has been based on the application of mathematical algorithms to records of past consumption values and/or causal meteorological variables. The standard mathematical models achieve acceptable performance in terms of accuracy when the influence of external non-cyclic factors are negligible or absent. However, this absence of external and abnormal factors is very rarely the case in the vast majority of water networks, hence in order to be capable of providing accurate forecasts in a real network situation, an ability to account for many of the occurrences of abnormal demands is necessary. Chapters 4 and 5 describe the methodologies used in the research upon which this thesis is based that have achieved significant success in incorporating non cyclic influences into demand predictions.

CHAPTER 3

THE ARIMA TIME SERIES PREDICTOR

3.1 Introduction.

The use of ARIMA based short term prediction algorithms has been well established in a number of fields including electrical load prediction[118,147,152] and water demand prediction[80,144] applications with many utilities in each of the two fields currently using such algorithms as the means of generating forecasts. It was therefore deemed logical to use a proven ARIMA model as representative of the current state of mathematical demand forecasting technology and hence utilise the results produced by such a model to provide accuracy comparisons with the results produced by the innovative techniques introduced in this thesis. This chapter provides a detailed description of the ARIMA methodology and introduces the algorithm used throughout the work described in this thesis to provide comparative results for both electrical load data and water demand data.

3.2 Auto Regressive Integrated Moving Average Models.

Many mathematical time series prediction methodologies, such as exponential smoothing, assume that at any point in the time series the observed value of the time series variable will consist of the deterministic mean of the process plus a random error component. However, the assumption of independent observations is frequently unwarranted, since many time series display a high degree of dependency between successive observations. A typical water or electricity demand time series displays such dependency between observations due to the presence of strong daily and weekly cycles within the consumption patterns, the ARIMA prediction methodology is designed to exploit this dependency in its generation of a forecast.

The variations observed in a typical time series of water or electricity consumption data can be considered to be the result of a number of causal processes acting upon that time series. If these processes can be correctly identified and modelled mathematically, then using these derived models we can generate predictions of the future values of the time series. However, in addition to the identifiable causal processes acting upon the time series, there will also be a non deterministic error component present. If all other causal processes have successfully been identified, then this non deterministic series will be random white noise.

3.2.1 Data Differencing Operation

A typical time series of water or electrical consumption data, $Z_t (t = 1,2,\ldots,N)$ will exhibit strong daily and weekly periodicity, illustrated by the plot of the autocorrelation between time lagged data points shown in figures 3.2.1. and 3.2.2 . However, an additional seasonal effect may be present that has a period greater than the span of the available data (such as the slow variation in the overall level of demand from season to season through the year), this will manifest itself in the mean of the time series

Figure 3.2.1 Autocorrelation Function with Time Lag 1

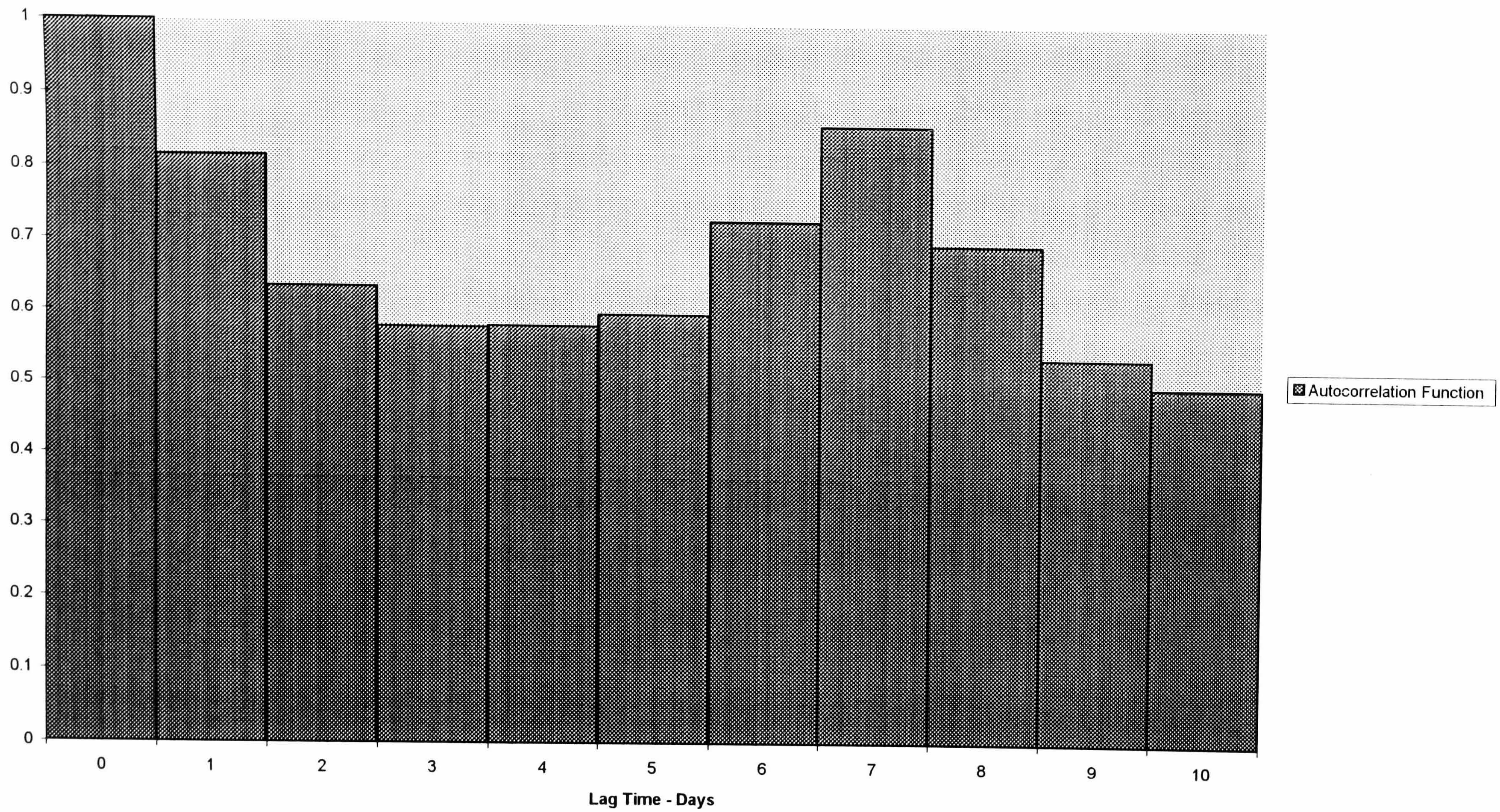
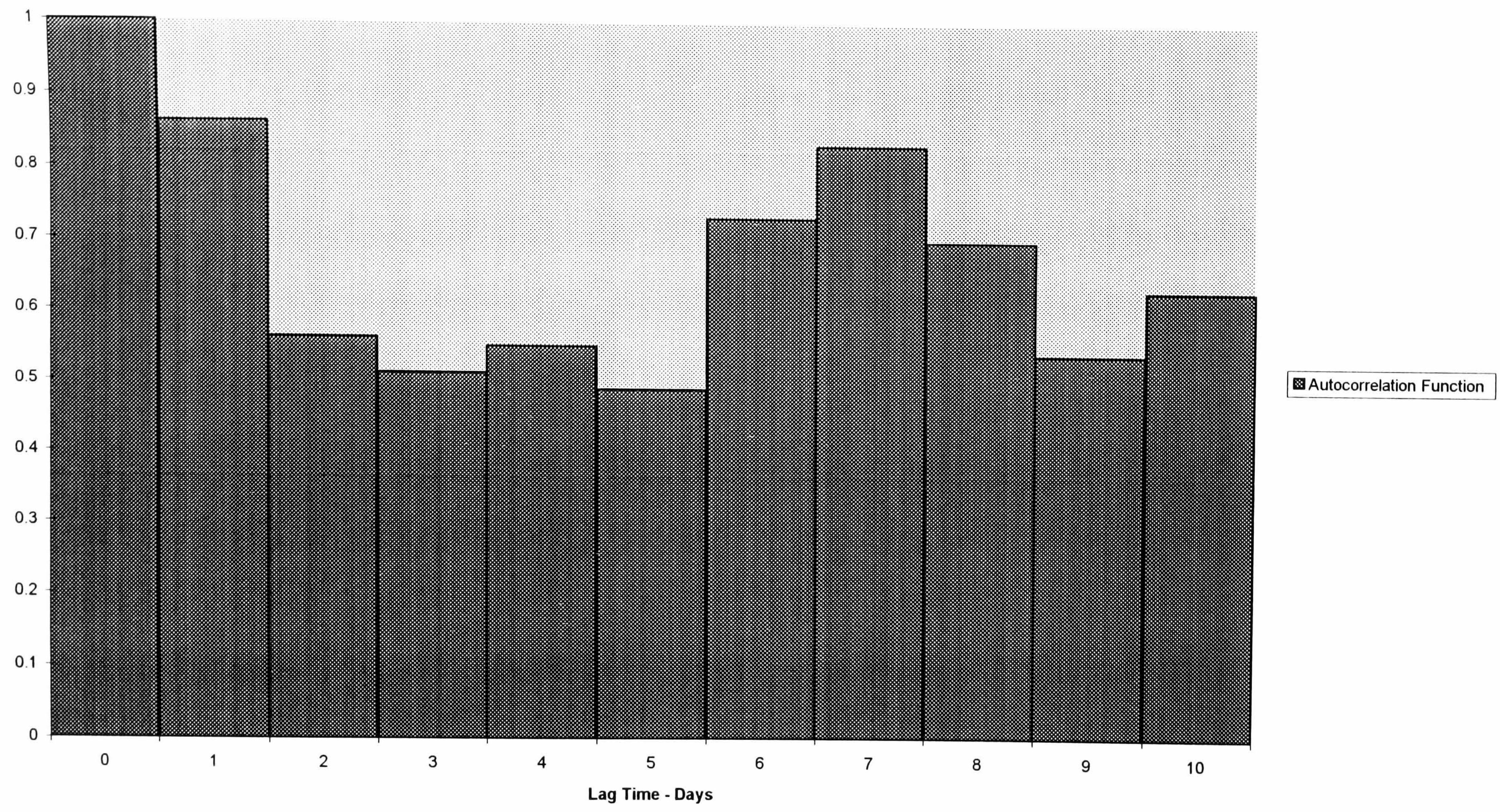


Figure 3.2.2 Autocorrelation Function with Time Lag 2



varying with time i.e. a non stationary series mean. It is therefore necessary to transform Z_t into a stationary mean series W_t, this transformation being obtained by:

$$W_t = f(Z_t) \tag{3.2.1.1}$$

Where f is the transformation required to achieve the stationary series W_t.

The transformation f is a differencing operation that can be represented by:

$$W_t = (1-B)^d (1-B^s)^D Z_t \tag{3.2.1.2}$$

Where B = backward shift operator,

$$BZ_t = Z_{t-1} \tag{3.2.1.3}$$

B^S = seasonal backshift operator,

$$B^S Z_t = Z_{t-s} \tag{3.2.1.4}$$

d, D = the daily and weekly difference orders and s is the seasonal periodicity of the time series Z_t

$$\nabla_s^D Z_t = (1-B^s)^D Z_t \tag{3.2.1.5}$$

$$\nabla^d = (1-B)^d \tag{3.2.1.6}$$

∇ = backward difference operator,

$$\nabla_s Z_t = (1-B^s) Z_t = Z_t - Z_{t-s} \tag{3.2.1.7}$$

The stationary time series W_t is therefore given by:

$$W_t = \nabla^d \nabla_S^D Z_t \qquad (3.2.1.8)$$

Examination of the sample autocorrelation function of the time series data used in the demand forecasting application indicated strong correlation between data points 1, 48 and 336 time steps appart.

3.2.2 The Selection of the Model Structure

The prediction problem is now reduced to the determination of a class of models that will adequately represent the stationary time series W_t. For a time series composed of N data points with seasonal periodicity s ,this can be achieved using a model composed of auto regressive and moving average components given by

$$\phi(B)\Phi(B^s)W_t = \theta(B)\Theta(B^s)a_t \qquad (3.2.2.3)$$

The Auto regressive components are:

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p \qquad (3.2.2.4)$$

$$\Phi(B^s) = 1 - \Phi_1 B^s - \Phi_2 B^{2s} - \ldots - \Phi_P B^{Ps} \qquad (3.2.2.5)$$

Where p is the order of the AR component and P is the seasonal difference order.

The Moving Average components are:

$$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \ldots - \theta_q B^q \qquad (3.2.2.6)$$

$$\Theta(B^s) = 1 - \Theta_1 B^s - \Theta_2 B^{2s} - \ldots - \Theta_Q B^{Qs} \qquad (3.2.2.7)$$

In order to determine the coefficients associated with each backshift operator it is necessary to fit the model via a minimisation of the sum of squared errors for each sample point i.e. determine the values of the coefficients $\phi, \Phi, \theta, \Theta$ that minimise:

$$S = \sum_{t=1}^{N} a_t^2(\phi, \Phi, \theta, \Theta) \tag{3.2.2.8}$$

In order to achieve the above minimisation equation 3.2.2.3 can be written for each time step t=1,...,N in matrix form as:

$$\mathbf{DFw} = \mathbf{TRa} \tag{3.2.2.9}$$

$$\mathbf{a} = \mathbf{T}^{-1}\mathbf{R}^{-1}\mathbf{FDw} \tag{3.2.2.10}$$

Where **D** represents $\phi(B)$, **F** represents $\Phi(B^S)$, **T** represents $\theta(B)$ and **R** represents $\Theta(B^S)$ as set out below:

$$\mathbf{D} = \begin{bmatrix} 1 & 0 & . & . & . & . & 0 \\ \phi_1 & 1 & & & & & . \\ \phi_2 & \phi_1 & 1 & & & & . \\ & . & . & . & & & . \\ & & . & . & . & & . \\ & & & . & . & . & . \\ & & & & \phi_2 & \phi_1 & 1 \end{bmatrix}$$

$$
\mathbf{F} = \begin{bmatrix}
1 & 0 & . & . & . & . & . & . & 0 \\
. & . & & & & & & & . \\
. & & . & & & & & & . \\
0 & & & . & & & & & . \\
\Phi_1 & 0 & . & . & 1 & & & & . \\
. & . & . & & . & . & & & . \\
. & & . & . & . & & . & & . \\
. & & & . & 0 & & & . & 0 \\
\Phi_P & . & . & & \Phi_1 & 0 & . & . & 1
\end{bmatrix}
$$

$$
\mathbf{T} = \begin{bmatrix}
1 & 0 & . & . & . & . & 0 \\
\theta_1 & 1 & & & & & . \\
\theta_2 & \theta_1 & 1 & & & & . \\
& . & . & . & & & . \\
& & . & . & . & & . \\
& & & . & . & . & . \\
& & & & \theta_2 & \theta_1 & 1
\end{bmatrix}
$$

$$
\mathbf{R} = \begin{bmatrix}
1 & 0 & . & . & . & . & . & . & 0 \\
. & . & & & & & & & . \\
. & & . & & & & & & . \\
0 & & & . & & & & & . \\
\Theta_1 & 0 & . & . & 1 & & & & . \\
. & . & . & & . & . & & & . \\
. & & . & . & . & & . & & . \\
. & & & . & 0 & & & . & 0 \\
\Theta_Q & . & . & & \Theta_1 & 0 & . & . & 1
\end{bmatrix}
$$

The dimentions of the matrices are assumed to be suitable for the vectors that they are multiplying and the intervals between non zero elements is given by the period s.

$$\mathbf{w}^{\mathrm{T}} = [w_1, w_2, \ldots, w_N]$$

$$\mathbf{a}^{\mathrm{T}} = [a_1, a_2, \ldots, a_N]$$

From equations (3.2.2.9) and (3.2.2.10) the function to be minimised can now be written as:

$$\mathbf{a}^{\mathrm{T}}\mathbf{a} = \mathbf{w}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{F}^{\mathrm{T}}\left[\mathbf{R}^{-1}\right]^{\mathrm{T}}\left[\mathbf{T}^{-1}\right]^{\mathrm{T}}\mathbf{T}^{-1}\mathbf{R}^{-1}\mathbf{F}\mathbf{D}\mathbf{w} \qquad (3.2.2.11)$$

Any suitable optimisation algorithm can be used to achieve this minimisation, Sterling and Bargiela proposed the use of the Newton Raphson iterative procedure in their investigation of applying ARIMA prediction models to water demand time series data. Initial estimates are made of the values of the parameters $\phi, \Phi, \theta, \Theta$ and corrections to these estimates are generated using an approximation to the inverse Hessian matrix according to the Fletcher-Powell method. The parameter estimate updating is carried out at each sucessive time step by re-evaluation of the gradient so as to incorporate the most recent data. The resulting model will be a best fit, in a least squares sense, to the time series data W_t up to time step N.

3.2.3 Identification of the Model

Equation 3.2.2.3 shows the general form of the ARIMA algorithm that is applicable to this kind of problem, in order to derive the specific form it is necessary to determine the parameters within 3.2.2.3 that are significant to the data in the specific time series that is to be modelled. To achieve this it is necessary to determine the following information from the time series: the seasonal periods present s_i , the difference orders d_i and the auto regressive and moving average orders p_i and q_i. Statistical examination of the time series is used to derive values for these parameters.

Although there are several statistical methods that are applicable to the determination of the above parameters[70], the method used in this application is the examination of the sample autocorrelation function r_k , which is used to indicate the relative strengths of correlation between data points k time steps apart in a time series containing N data points.

$$r_k(Z_t) = \frac{C_k(Z_t)}{C_0(Z_t)} \qquad k = 0, 1, 2, \ldots\ldots, N - 1 \quad (3.2.3.1)$$

Where $C_k(Z_t)$ is the autocovariance function given by:

$$C_k(Z_t) = \frac{1}{N}\sum_{t=1}^{N-k}(Z_t - \overline{Z})(Z_{t+k} - \overline{Z}) \qquad k = 0, 1, 2, \ldots\ldots, N - 1 \quad (3.2.3.2)$$

If the calculated values of $r_k(Z_t)$ are near to zero for samples k time steps apart then there is a poor correlation between the samples in question, however values of $r_k(Z_t)$ approaching unity indicate strong correlation between the samples separated by the current value of k.

3.2.4 Generating the Prediction.

The model produced by the process described above may be used to generate predictions of future values of the series Z_t . To do this the series Z_t of length N_t is extended by the number of time slots corresponding to the length L of the prediction period. The values of the residual noise series a_t over the length of the prediction period are assumed to be zero. The extended time series vectors **z'** and **a'** are:

$$\mathbf{z}' = [Z_1, Z_2, \ldots, Z_N, Z_{N+1}, \ldots, Z_{N+L}]^{\mathrm{T}}$$

$$\mathbf{a}' = [a_1, a_2, \ldots, a_N, 0, \ldots, 0]^{\mathrm{T}}$$

Substituting the above vectors in equation 3.2.2.4 gives:

$$\mathbf{FDz}' = \mathbf{RTa}' \qquad (3.2.4.1)$$

Since $\mathbf{w} = \mathbf{Dz}'$ we have:

$$\mathbf{Fw} = \mathbf{RTa}' \qquad (3.2.4.2)$$

if $\mathbf{F}^{-1}$ (the inverse of the Φ auto regressive regression operator matrix) exists.

3.2.5 Numerical Constraints

If the model has been correctly identified, the values of all the auto regressive and moving average parameters should lie within the range:

$$-1 < \phi_{ij} < 1 \qquad \forall_{i,j}$$
$$-1 < \theta_{kl} < 1 \qquad \forall_{k,l}$$

If the non-linear minimisation is unconstrained then parameter values outside this range may be tried during the search for the minimum, this can cause numerical overflow problems, hence the parameter values are artificially constrained to within the (-1 , 1) boundaries.

3.2.6 Data Considerations

There is minimum amount of past data that is required by the ARIMA algorithm in order for it to be able to correctly identify all the seasonal periods operating within the time series, the calculation of the minimum data requirement is given by Gann[55] as:

$$N_\theta = \sum_{i=1}^{N_s} (q_i s_i)$$
$$N_\phi = \sum_{i=1}^{N_s} (d_i + p_i) s_i$$
$$N_c = \sum_{i=1}^{N_s} (d_i + p_i + q_i) s_i + 2\max(q_i, s_i)$$
$$N_t \geq N_c + \max(N_\theta, N_\phi)$$

Where: s = seasonal period

N_s = number of seasonal periods identified.

p, d, q = orders of AR, difference and MA components respectively.

N_θ = data requirement of the MA components.

N_ϕ = data requirement of the AR components.

N_c = data requirement of the differencing components.

N_t = total data requirement.

3.2.7 Implementation

The ARIMA algorithm used in the applications described in this thesis was initially developed for load prediction in a power systems environment by Gann [55] at Durham University. It is written in FORTRAN and runs under the VMS operating system on VAX station 3100 series workstations. In the course of the current research, the algorithm was tested on power systems data supplied by the CEGB for 1984 and 1985 as part of the assessment of the prototype combined forecaster described in Chapter 4. The algorithm was then adapted for use with water consumption data and used to provide comparison with water demand prediction results.

The water consumption data was supplied by Thames Water PLC and consisted of flow measurements and reservoir level measurements from the Slough and High Wycombe water supply networks for the year 1990. The flow and level measurements were converted into half hourly consumption totals for each area and statistical testing was conducted to determine the particular ARIMA model that was applicable to the characteristics of the data. The model identification was carried out with the aid of partial autocorrelation function plots such as those displayed in figures 3.2.1 and 3.2.2. Three seasonal components were identified as being present within the data a) half hourly b) daily and c) weekly, corresponding to the dependency of the current value on the value of the immediately preceding data point, the value of the data point 24 hours earlier and the value of the data point one week earlier.

From the above the AR order p, the differencing order d and the MA order q for the model chosen for both the power systems and water consumption data were as follows:

$[p=1, d=0, q=1]$ For the first seasonal component i.e. half hourly.

$[p=1, d=0, q=1]$ For the second seasonal component i.e. daily.

$[p=0, d=1, q=1]$ For the third seasonal component i.e. weekly.

Calculations as outlined in section 3.2.7 were carried out to determine the minimum amount of data that would be required by the model, at half hourly data intervals this minimum corresponds to five weeks data. However, the most consistent prediction performance was found to be achieved when seven weeks data (2330 data points) were submitted, the weekly trend being more successfully incorporated into the model in the latter case. If too great an amount of data is used then there is the risk that consumption influencing factors that were affecting the data a number of weeks in the past but are no longer active will exert an undesired influence on the current prediction.

3.3 ARIMA Results.

3.3.1 Power Systems Data

At the start of the research described in this thesis, the initial aim was to prove the basic concept that knowledge based techniques could be used to improve the ability of one day ahead forecasting to account for abnormal non-cyclic influences. The following chapter describes the initial prototype system designed to achieve this aim, however, the only suitable data that was available during the development of this first prototype was half hourly electricity consumption totals from the years 1984 and 1985. Therefore, the initial testing of the accuracy of the ARIMA based predictions was carried out using this power systems data.

Prediction accuracy E_{RMS} is determined from the rms of the prediction errors:

$$E_{RMS} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}[Z_t - Z_t']^2}$$

Where Z_t and Z_t' are the actual and predicted values for the forecast variable and N is the number of elements in the forecast.

The ARIMA algorithm was found to produce 24 hour predictions (composed of 48 data points) with acceptable accuracy as long as the daily and weekly load patterns remained stable. The daily profiles of actual and predicted load displayed in figures 3.3.1, 3.3.2 and 3.3.3 are examples from days where there was little apparent influence by external distorting factors. As previously outlined such distortions to the normal load profile are however not uncommon, they can be caused by a wide array of factors and effects, examples in the case of electrical load are, the influence of weather

Figure 3.3.1 ARIMA and Actual Load Normal Weekday

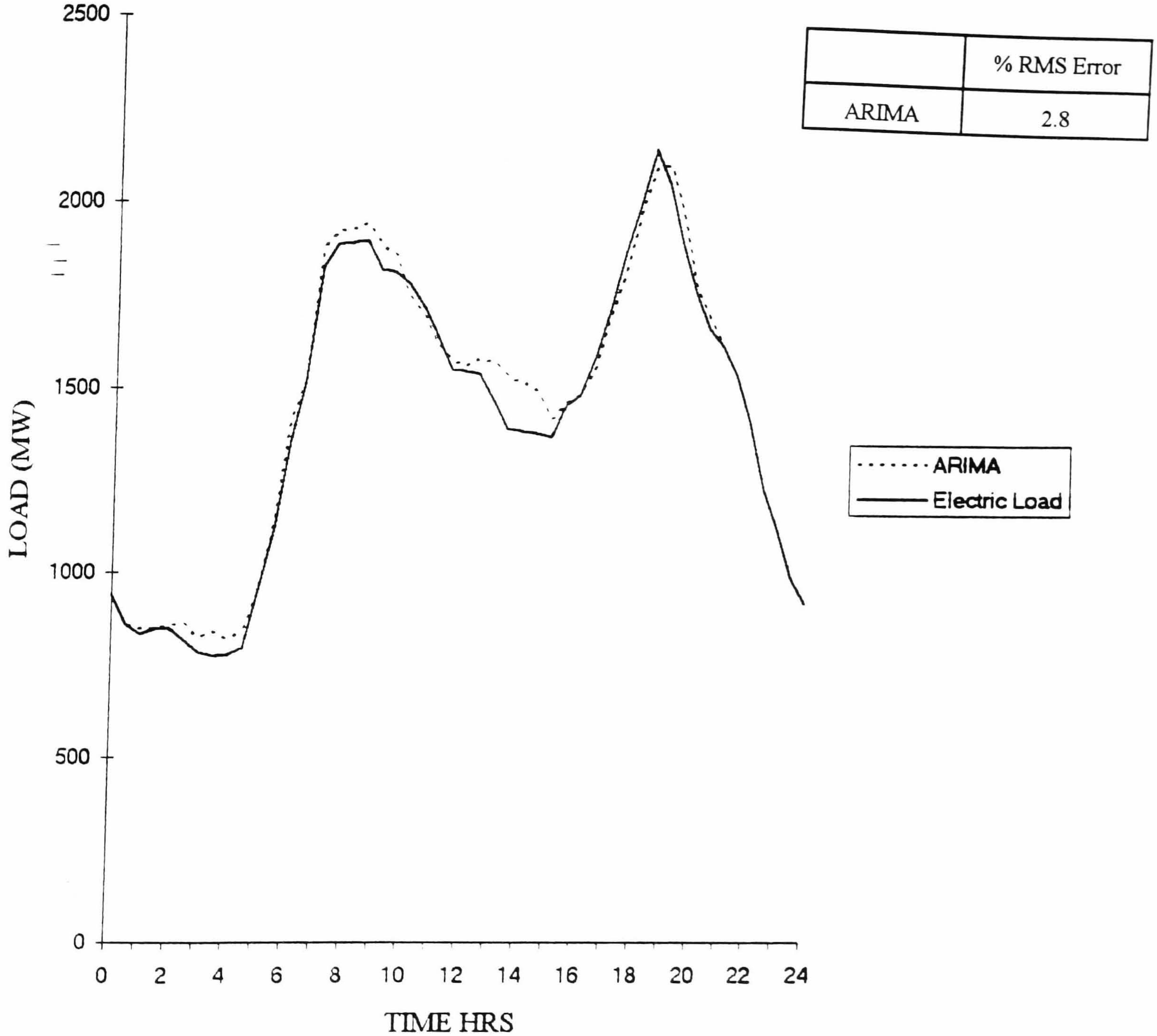


Figure 3.3.2 ARIMA and Actual Load Normal Weekend Day

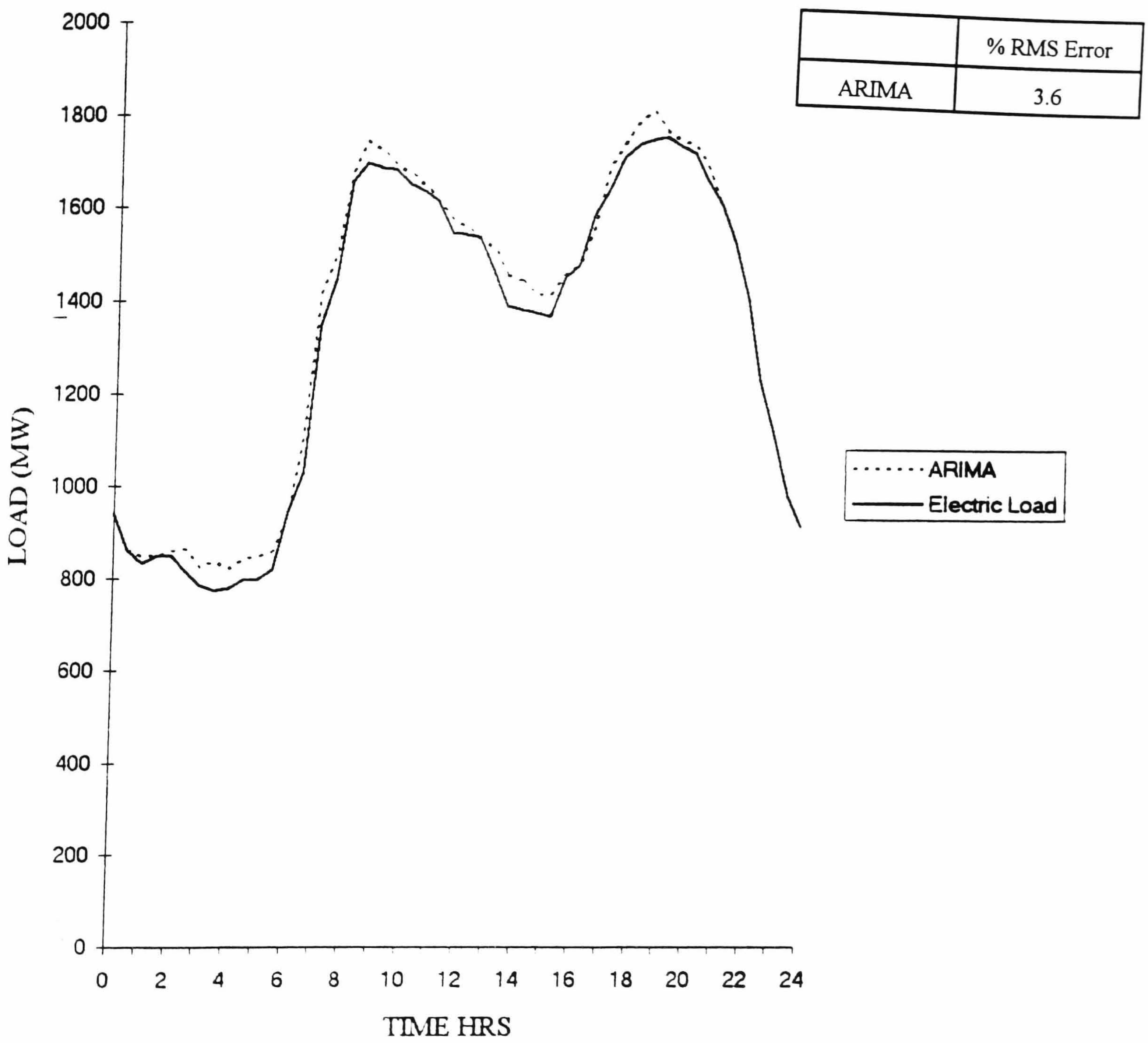


Figure 3.3.3 ARIMA and Actual Load Normal Weekday 2

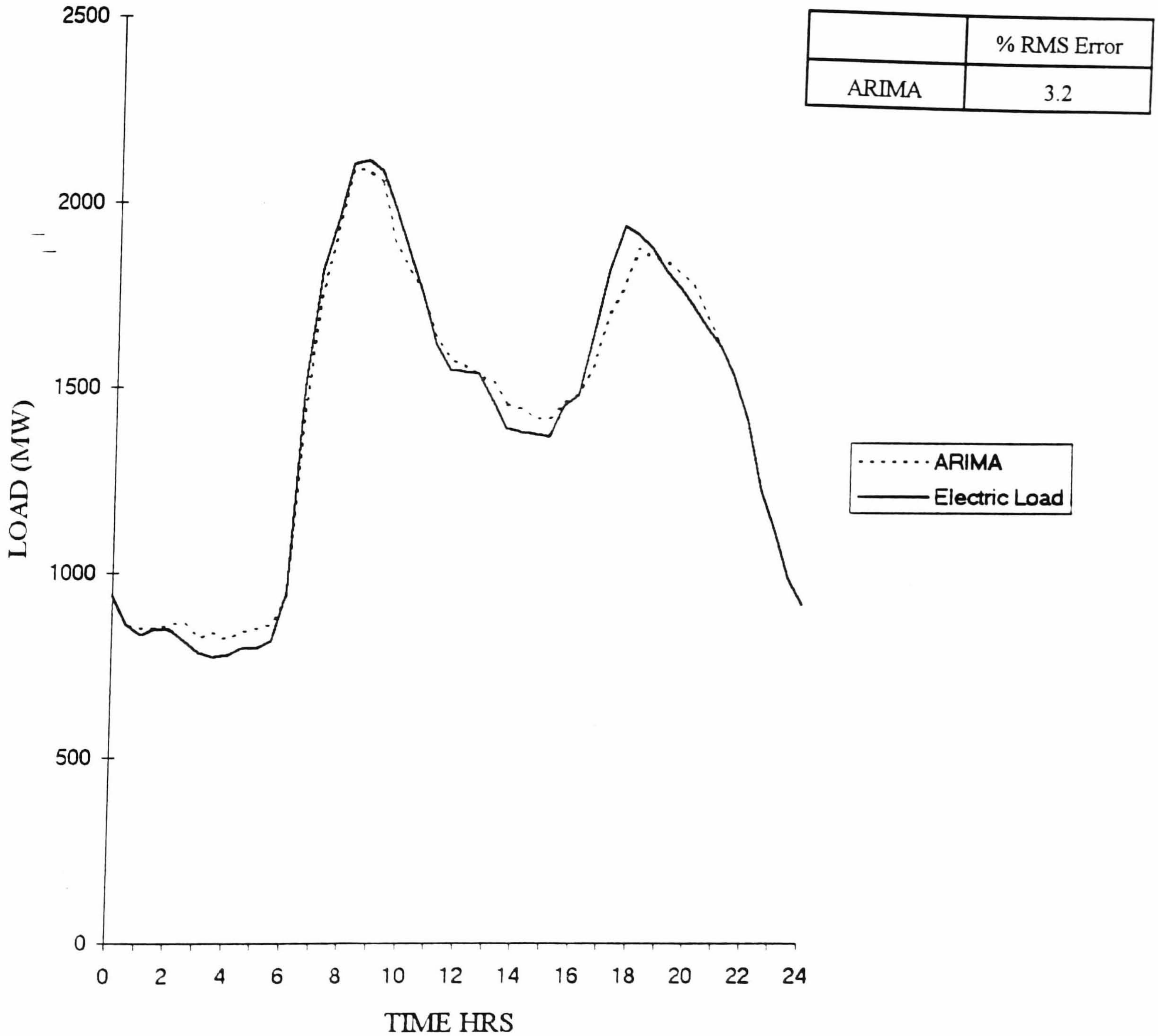


Figure 3.3.4 Actual and ARIMA Bank Holiday Load Profile

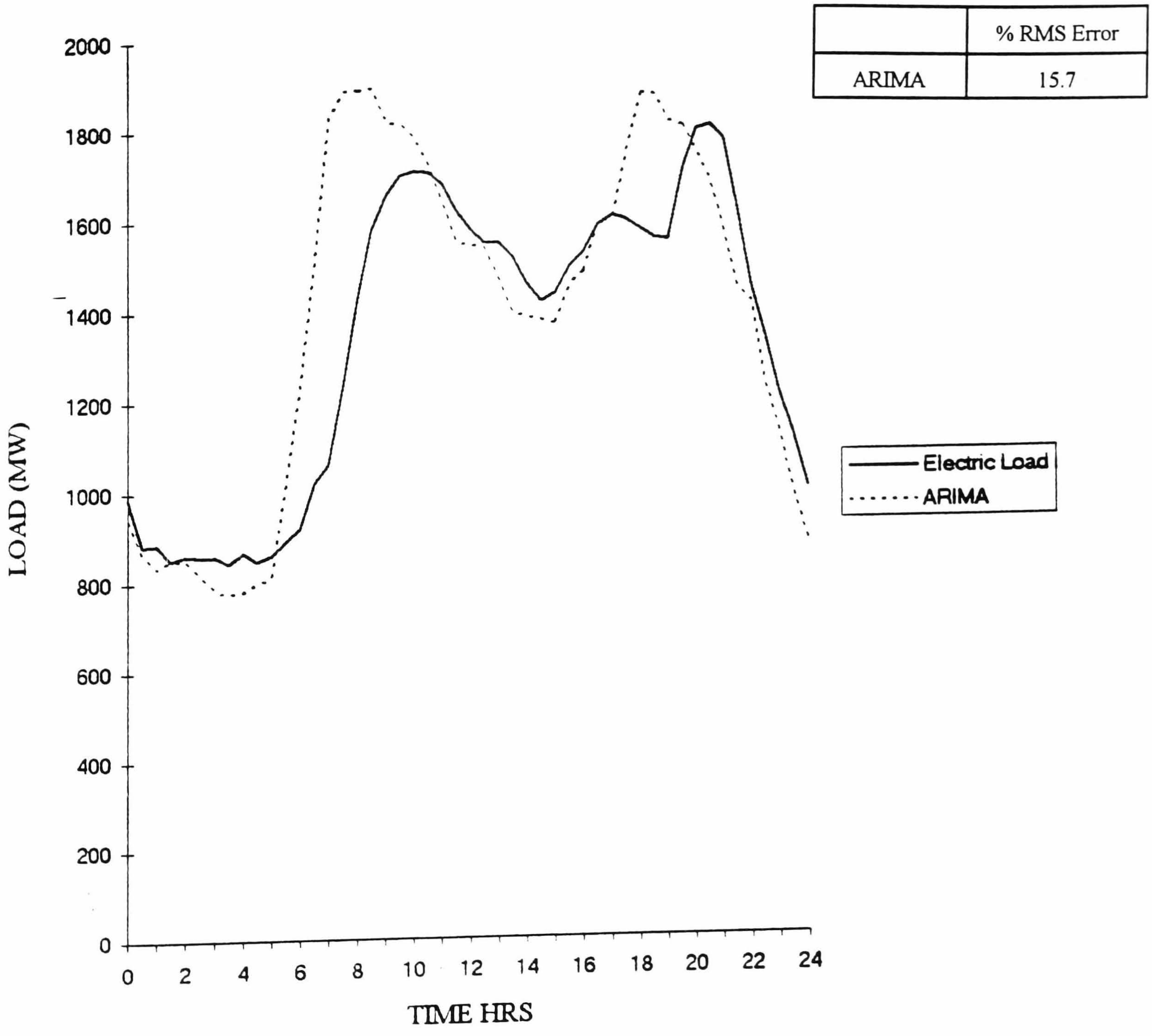


Figure 3.3.5 ARIMA and Actual Load Tuesday After Bank Holiday

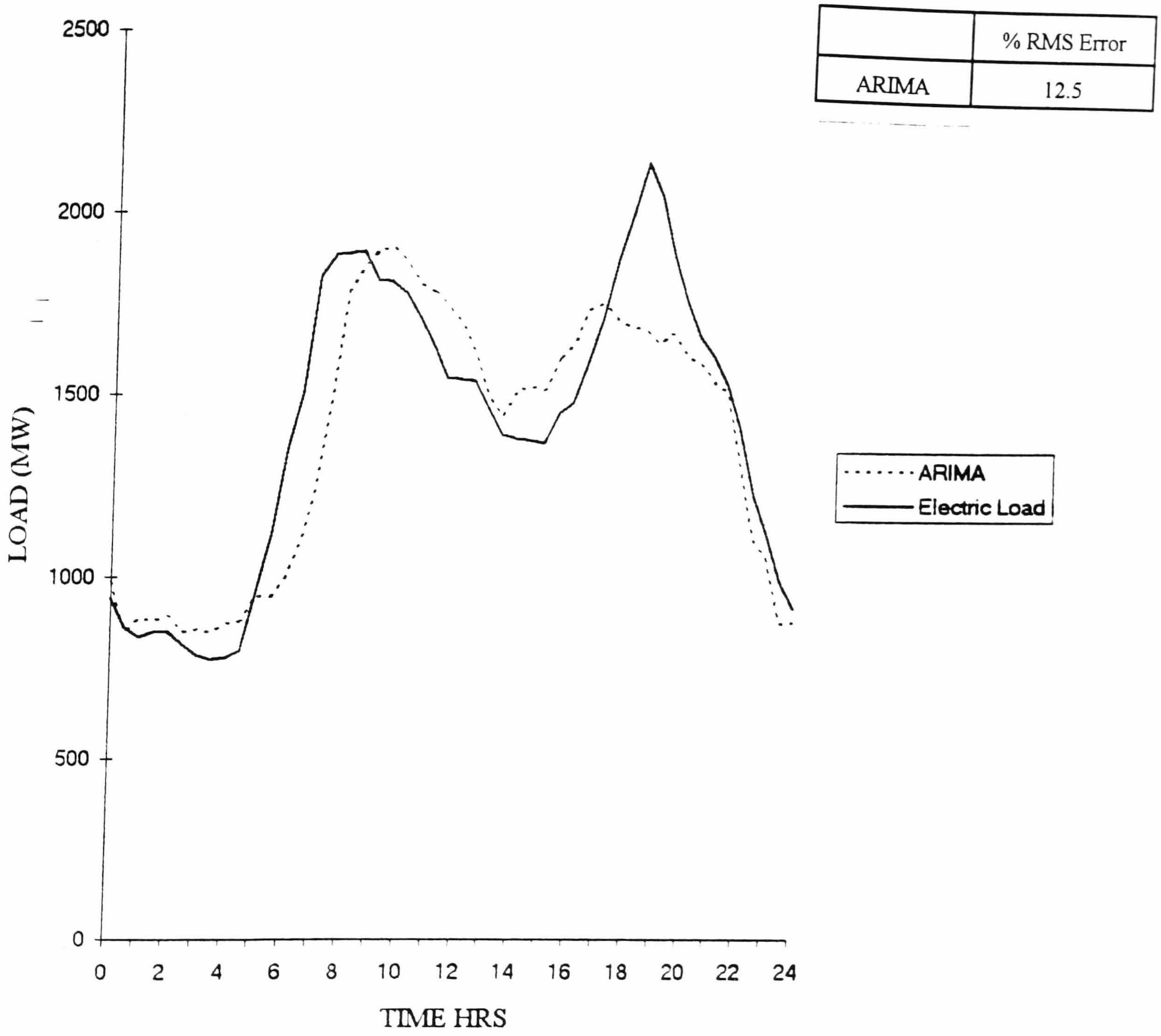
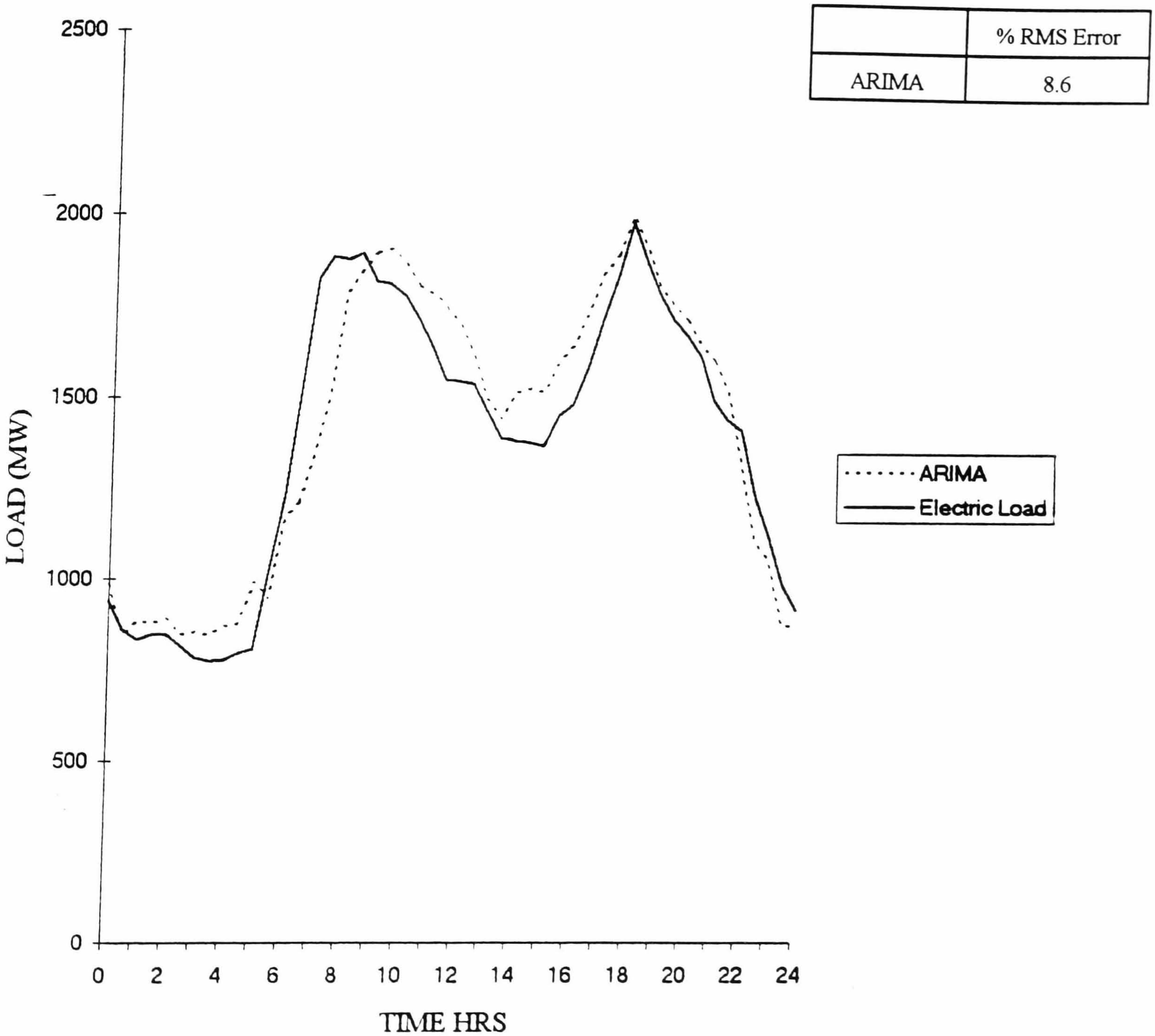


Figure 3.3.6 ARIMA and Actual Load 1 Week After Bank Holiday



variables (such as temperature, cloud cover, humidity etc.), the occasion of a special social or sporting event or the occurrence of public holidays. As can be seen from the profiles shown in figure 3.3.4, the presence of a distorting factor (in this case the prediction is for a bank holiday Monday) prevents the ARIMA model from being able to correctly predict the load. Large errors between the predicted and actual load mean any operational decisions that were based on the predicted level of load could be invalid in the light of the actual situation.

In addition to being unable to successfully predict the influences of the many non cyclic factors that influence the pattern of electrical consumption, the accuracy of the ARIMA predictions can be further compromised by the presence of distorted data in the time series used to generate the predictions. Examples of this effect can be seen in figures 3.3.5 and 3.3.6 which show the actual and predicted profiles for the Tuesday after a bank holiday and also the Tuesday one week later. It is clear that the presence of the typical bank holiday profile within the data submitted to the ARIMA algorithm has had an unwanted influence on the predicted profiles. The prototype described in the following section provides a method for correcting some of the above faults.

3.3.2 Water Network Data

Submitting the water consumption data from Thames Water to the ARIMA algorithm produced a similar set of results to those seen with the electrical load data. Acceptable accuracy was achieved by the ARIMA algorithm during periods where external influences, such as those due to meteorological variations, were at a minimum. A logic filter was used to remove the obvious bad data 'spikes' and 'troughs', but no further smoothing or filtering was performed upon the data in order to minimise the risk of erroneously removing fluctuations in the demand level that are the result of the consumption altering factors that this research was aiming to identify.

The results shown in figures 3.3.7 , 3.3.8 and 3.3.9 are for non holiday days with normal weather conditions for the time of year. It should be noted that compared to the examples of electrical load profiles, the water demand data displays a large noise component, this being due to the measuring apparatus used in the water industry.

The influence of variations in the weather conditions was found to have the most significant effect on the accuracy of the predictions produced by the ARIMA algorithm, this was especially evident in the spring period where rapid and dramatic changes in the prevailing weather cause obvious distortions to the consumption patterns. The examples in figures 3.3.10, 3.3.11 and 3.3.12 show the effect of increasing temperature and dryness upon the shape of the water demand profile, and the consequent decrease in the accuracy of the ARIMA forecasts.

Figure 3.3.7 Typical Weekday Water Demand Profile

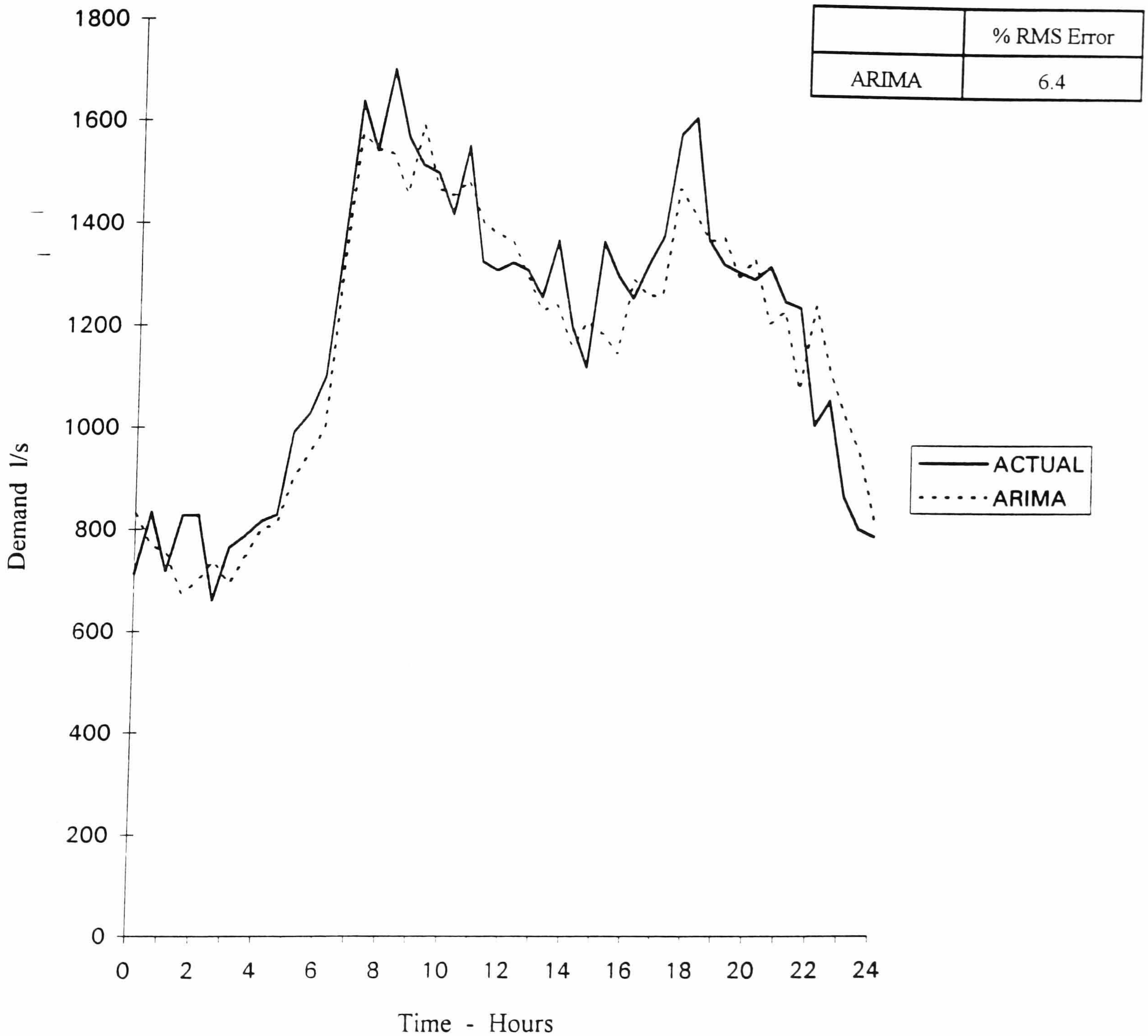
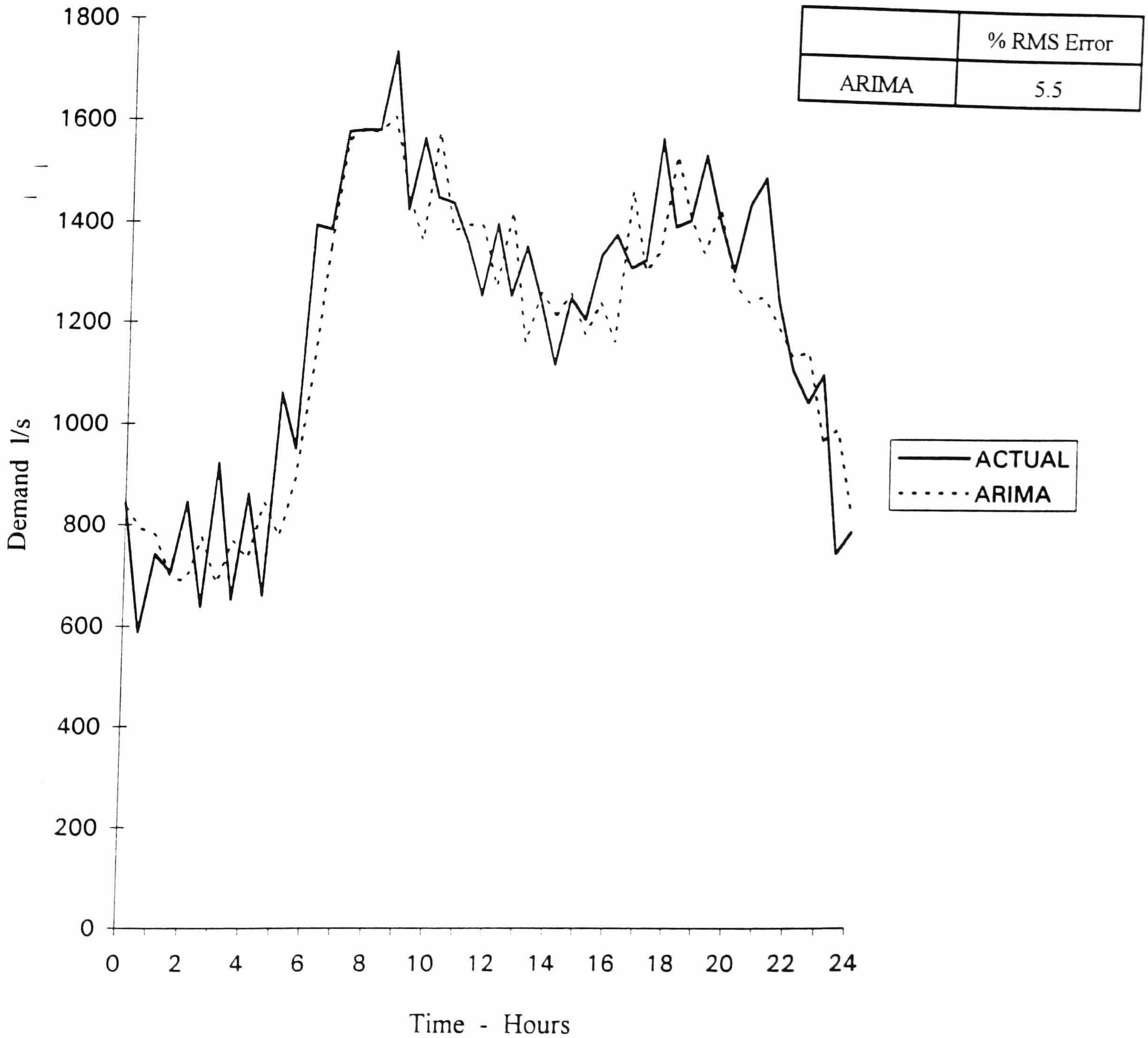


Figure 3.3.8 Typical Weekday Water Demand Profile 2



	% RMS Error
ARIMA	5.5

Figure 3.3.9 Typical Sunday Water Demand Profile

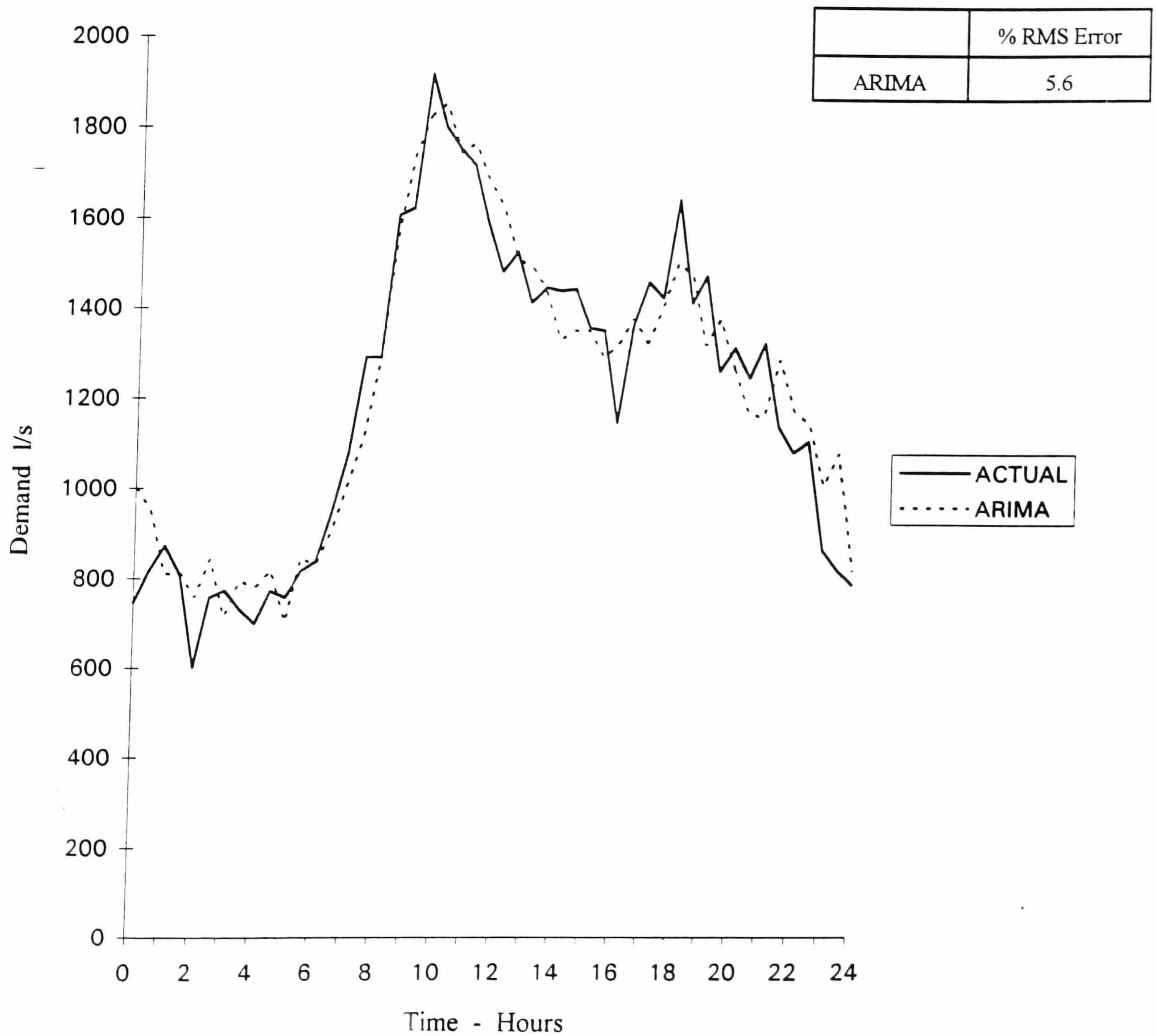
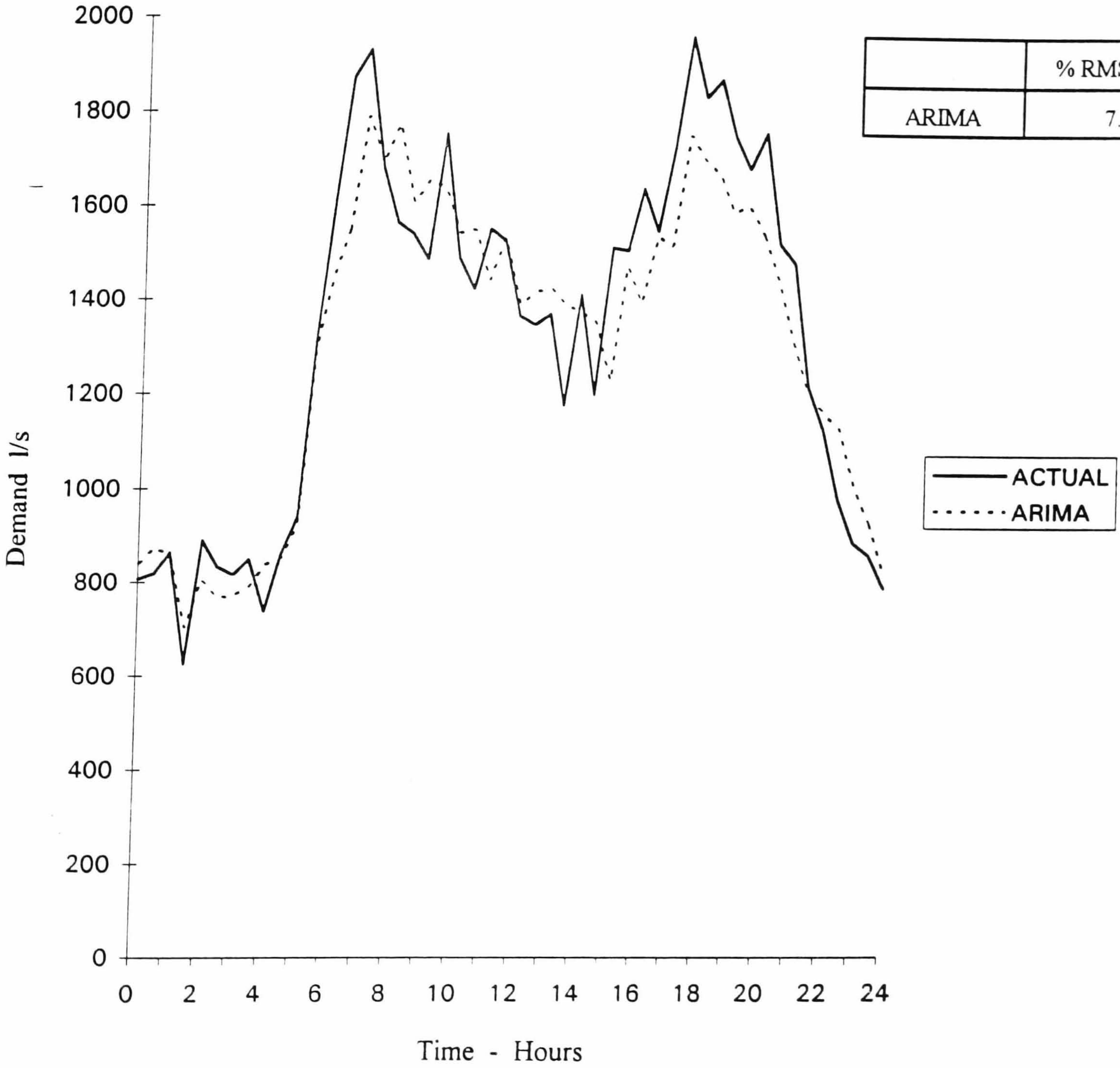


Figure 3.3.10 Effect of Increasing Temp/Dryness on Water Demand and Arima Prediction Accuracy



	% RMS Error
ARIMA	7.1

Figure 3.3.11 Effect of Increasing Temp/Dryness on Water Demand and Arima Prediction Accuracy

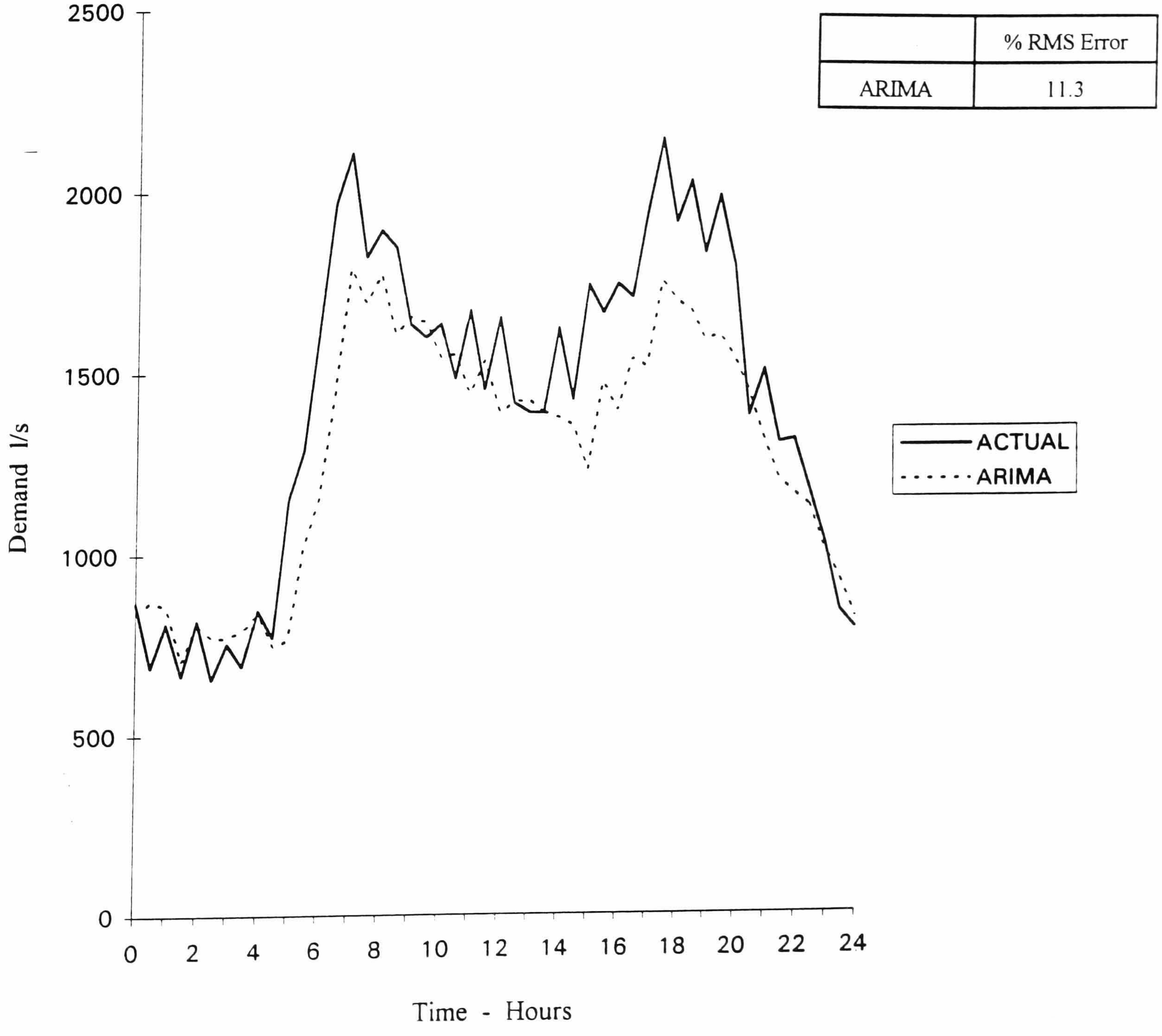
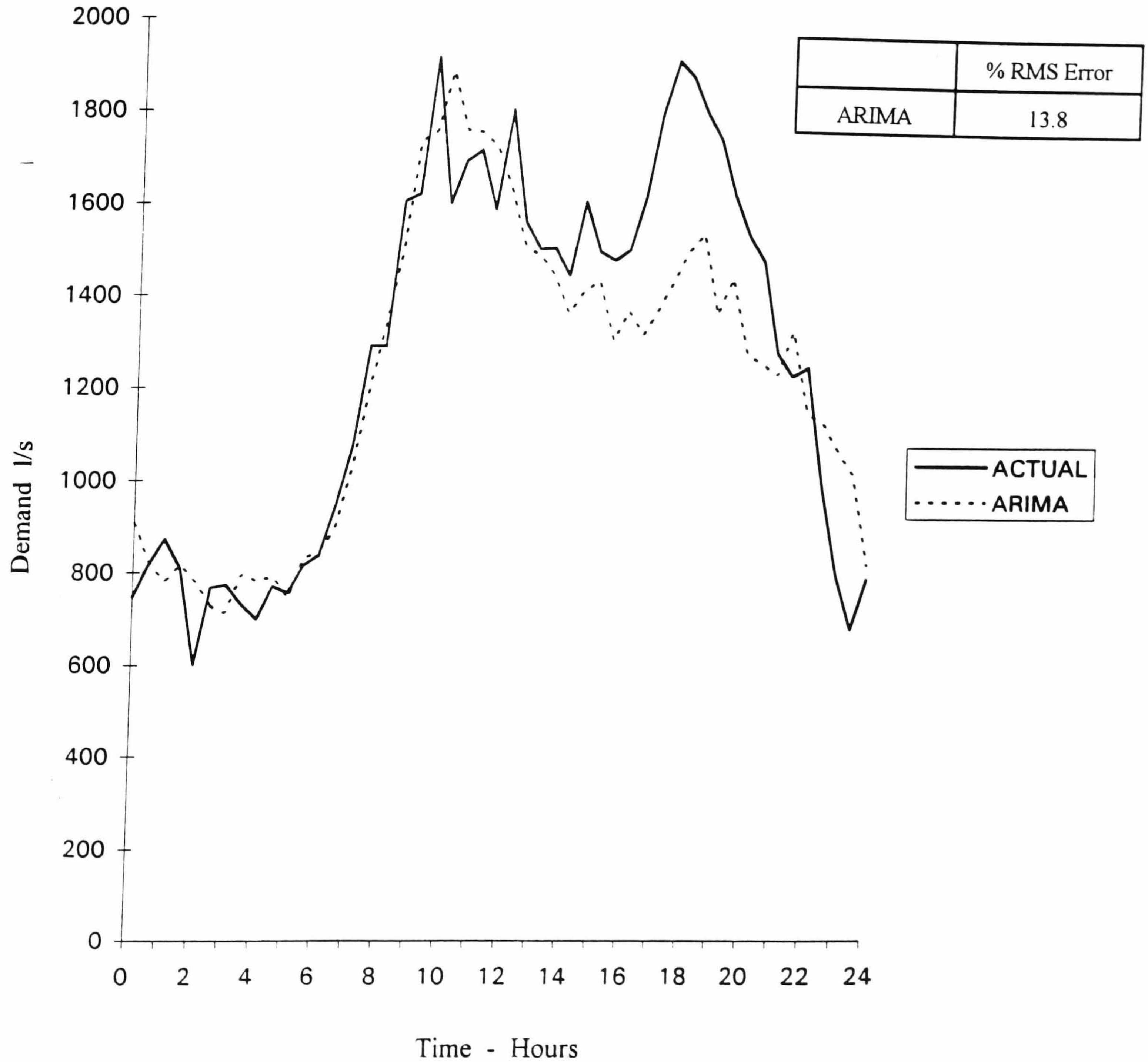


Figure 3.3.12 Effect of Increasing Temp/Dryness on Water Demand and Arima Prediction Accuracy



CHAPTER 4

KNOWLEDGE BASED SYSTEMS CONCEPTS, EXAMPLES AND APPLICATIONS

4.1 Introduction

The investigation of Knowledge Based Systems falls within an area of computing research known generally as Artificial Intelligence. The term Artificial Intelligence or AI, covers a broad spectrum of topics which include machine learning [104,161], natural language and speech recognition systems [82,13] and visual recognition systems [12,33,126]. The common central theme of each of these subjects is that of creating machines that can appear to behave in an manner that is analogous to human intelligence in their response to given stimuli. The degree to which this seemingly intelligent behaviour can truly be thought of as intelligence depends on the application involved. Although an expert system can appear to posses the problem solving capacity of one or more human experts, this appearance is wholly the result of the logical application of programmed rules and the controlling structures that govern the use of those rules. In contrast, the learning process that takes place within a neural network as it adapts its responses to the signals applied to it, can much more readily be envisaged as approaching the criteria many people would accept as indicating intelligence. However, even within the neural network, where the actual training and operation of the net are commonly treated in a ' black box' manner, the break is never made from the programmed algorithms that prescribe the way in which the net learns to adapt its behaviour. Hence, it is true to say that as yet, the field of AI has been concerned with the development of machines that mimic human reasoning and behaviour, rather than be the source of spontaneous thought and actions.

The reason for the investigation of the AI field in relation to the problems associated with the prediction of the patterns of water and electricity consumption, is that situations commonly occur in the operation of a water or electricity supply network that cause significant departures from the regular cyclic consumption patterns that are a fundamental requirement of wholly mathematical prediction systems. In such situations it is often the case that a potentially serious supply crisis is avoided by the engineer or operator overriding the information supplied by the prediction system and instead utilise their skill and knowledge built up over many years to correct the error. As a consequence of the drawbacks of relying on operator skill and judgement, such as lack of consistency, lack of availability and the longevity of the operator, there have been a number of attempts to harness the potential offered by AI techniques to provide method of capturing the knowledge necessary to mimic the actions of a highly experienced operator. In Chapter 2, as part of the review of previous work, several examples were given of the application of such knowledge based techniques in the field of electrical load prediction. In this chapter, the particular elements of AI that were used in the development of the combined demand forecaster that forms the basis of this thesis, will be introduced in more depth.

4.2 Knowledge Based Systems.

Knowledge based systems provide a means by which heuristic knowledge can be organised, structured and accessed in a manner that enables such knowledge to be incorporated into programs so that they provide a problem solving capacity. Many problems are by their nature either impractical for solution by algorithmic means or an algorithmic approach can provide only part of the answer. Prior to the development of knowledge based systems such problems were only solvable by the deployment of an expert in the particular field concerned. He would use his experience and intuition to arrive at a solution, though typically this solution would carry with it a degree of certainty/uncertainty dependent on the exact nature of the problem, the strength and quality of the evidence indicating the answer and the level of expertise of the expert himself. Knowledge based systems can provide the means by which some of the valuable knowledge possessed by one or more domain experts can be captured, formalised and structured so that it can be made available to non expert users.

4.2.1 What Is A Knowledge Based System.

There are three key elements that make up a typical expert system, these are briefly outlined below and are described in detail in section 4.3.

The working memory, this is where facts about the current situation are stored. In order to apply the correct rules in a given situation, the situation has to be defined i.e. what facts are known, what data is available and what goals are to be achieved. The working memory is highly volatile and acts as a scratch pad that is updated as the expert system progresses towards the desired goal. Initially the working memory will store the initial state of the available data and the goals to be satisfied, rules are then activated that act upon and where appropriate update the information stored in the working memory, sub-goals are created and where possible pursued to a solution. This proceeds until a solution to the original goal is found or the expert system is halted by lack of data thus preventing it from progressing any further towards a solution.

The production memory is a database of production rules that act upon the data stored in the working memory. These production rules are by far the most common format in which the expert knowledge in a particular domain is encoded, they consist of a condition part and a result part, the condition part requires to be satisfied by the data available to the system in order that the result part can be executed. The application of the appropriate rules enables the expert system to progress from the initial state described in the working memory towards the desired solution. The production rule database is built up during the construction of the expert system from interaction with one or more experts in the domain concerned. The rules remain largely static during the solution of a particular problem, but can be updated in the light of the overall system performance in order to improve that performance or enable a new situation to be incorporated.

The rule interpreter or inference engine provides the means by which the appropriate rules are selected for application to the data stored in the working memory. It also carries out the updating of data and goals in the working memory, in order to provide a means of progressing towards the desired solution. The rule interpreter also carries out conflict resolution to determine the order in which the rules which have been triggered by the data in the working memory are fired.

Knowledge based systems are commonly referred to by the term 'expert systems', however, this term is not popular with many of the researchers in the field who believe it is a misnomer. The reason for this is that for the majority of the knowledge based systems which have been developed, their performance is not comparable with that of the human experts in the domain to which they apply and it would therefore be wrong (and potentially dangerous) to treat the solutions derived from such systems with the same degree of reliance as solutions reached by the human expert. This thesis will therefore attempt to avoid the term expert system in discussion of this topic.

4.2.2 The Development of Knowledge Based Systems.

Applications to which knowledge based systems are most suited are those which concern a relatively specialised, narrow domain in which there are few highly trained and experienced personnel but for which there also exists is a wide demand for access to the information relating to the domain. By encoding the experts knowledge into a computer program, the knowledge can be disseminated to many more people than would otherwise be the case. The field of medical diagnosis is a very good example of the situation outlined above and indeed one of the most successful early knowledge based systems MYCIN [18] was developed to assist a physician in the prescription of disease specific drugs. MYCIN was designed to guide a user, via a series of questions concerning the results of a number of biological tests, to a set of possible candidate bacteria that could be the cause of a particular infection and based on this assessment the system suggested appropriate antibiotics. The system consisted of a rule base an inference engine and working memory as outlined in the previous section but also included an end user interface, an interface through which the expert could enter the knowledge to be utilised by the system and refinements such as an explanation system and uncertainty factors (these elements are described later in this chapter). The MYCIN design has been used as the basis of many similar knowledge based systems developed subsequently including PROSPECTOR [65] and DENDRAL [98].

Another early and sucessful knowledge based system that helped to stimulate commercial interest in the field, was the XCON system developed by Digital Equipment Corporation. This system was designed to provide assistance in the configuration of the components of mainframe computers and used a rule base of over 10,000 rules holding information on several hundred types of component. A similar system for the configuration of computer systems called R1 is described by McDermott in [102]. Such systems demonstrated a clear advantage over human experts in their ability to reliably and consistently manipulate such large volumes of detailed information.

The field of short term electrical load forecasting, which as described in Chapter 2 has close links with the prediction of water consumption, has seen the development of prediction systems that utilise knowledge based system and AI techniques. The combination of analytical and heuristic methods to arrive at a load forecast has been used in the ELFOS system described by Remior and Ayuso [127]. This uses a mathematical algorithm to predict the total demand for the coming 24 hours, the forecast being based on records of past daily demands, template daily demand profiles are then selected to arrive at an hourly forecast. Rules, which are held in context defined groups within the rule base, are then invoked which alter the derived daily profile in the light of known special characteristics for the day in question. Rahman and Bhatnagar [124] have also published work on the possibilities of using an knowledge based system to replace the role of the electricity system operator who relies on his experience and judgement to augment the load forecasts generated by mathematical prediction systems. They attempted to identify the variables and rules used by the system operators in estimating the likely system load as well as identify the criteria used in the decisions to apply specific rules in specific situations. Following the in depth analysis of the system operators skills a rule base containing the derived rules relating to the tasks carried out by the operators was constructed. This rule base was then used to adjust a base forecast generated by selecting similar days to the prediction day from a four week window.

In both of the above examples the work concentrates on the use of rule based knowledge based systems to account for the effects of weather conditions upon the daily load profile. However, there are a number of problems associated with using rules to hold knowledge about meteorological influences, the principle problem being that the relationship between prevailing weather conditions and resultant load variations is not stable. The weather related demand for electricity on any given day (or time of day) is the result of a general perception among the consumers, and this perception is based on a complex interaction of numerous individual factors such as temperature, humidity, cloud cover, day of the week, time of the year, past weather conditions etc. Because it is not practical to monitor and record every possible factor that has an effect upon the level of demand, systems such as the ones described above select a small number of key meteorological factors determined as being most influential. Unfortunately as a result it is therefore highly probable that on two days exhibiting the same values for the selected meteorological factors but separated in time, the load profiles will show significant differences. Figures 4.2.1 and 4.2.2 illustrate this.

This lack of consistency leads to problems of constantly trying to update rules in the light of their success in matching the actual observed load or can lead to a very unwieldy system caused by the attempt to generate a rule for every slightly different situation. Because this situation is mirrored in the link between weather conditions and the pattern of water consumption, the knowledge based system developed as part of the combined prediction system that forms the basis of this thesis does not concentrate on accounting for meteorological variations, instead this task is performed by a neural network.

The AI technique of pattern matching has been used in load forecasting with some success, the ALFA system [78] being a good example. This system initially generates a base load profile for each day of the year based on a 15 year database of past load data. The differences between this base load profile and the actual load observed for any given day is attributed to a weather dependent load. The values for this weather dependent load and the corresponding meteorological values for each day are held in a database spanning 10 years. The predicted values for the meteorological variables for the coming day are used as a basis for selecting the eight nearest matching days in the database. This search is augmented by weighting of the relative importance of individual meteorological factors at different times of the year and by restricting the search to similar days of the week i.e. weekdays, weekends, holidays etc. This system has been proved to be very effective in its application to the Upstate New York power distribution system, however, it is obviously dependent upon having available a large database of past data, this is very rarely the case with most utilities. If there is a shortage of past data on which to base

Figure 4.2.1 Profile for April 17th With Same Weather Conditions as May 25th

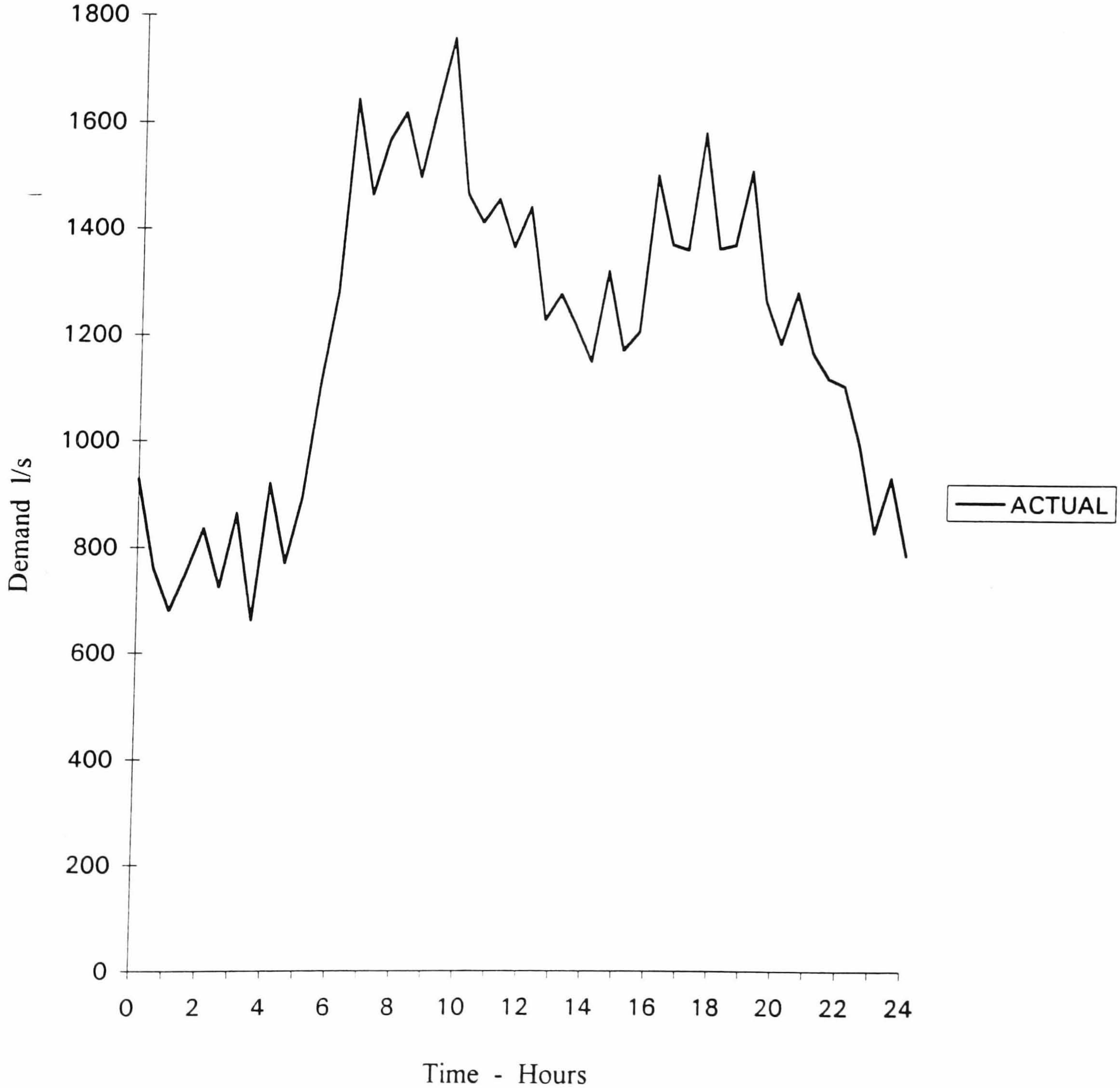
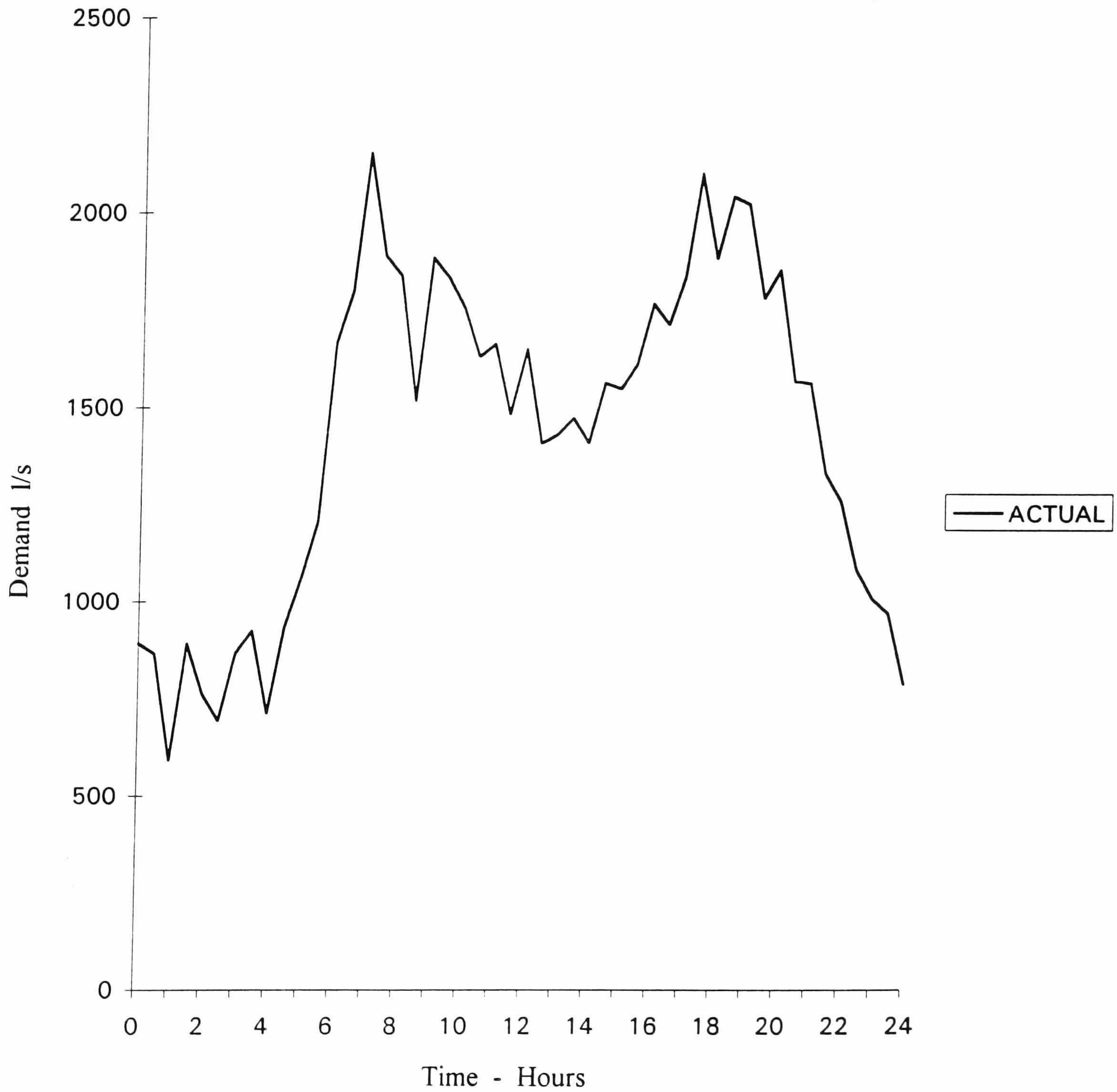


Figure 4.2.2 Profile for May 25th With Same Weather Conditions as April 17th



such a pattern matching system, the possibility arises of there being a single or very few examples of similar days to the conditions of the prediction day. This in turn would lead to encountering the problem raised previously of inconsistency between days apparently having identical meteorological conditions.

Although the use of knowledge based systems with in the water industry is not widespread at the present, there have been a number of applications developed and applied to specific areas. In the mid 1980's the Water Industry Knowledge Based Systems Club was set up under the ALVEY program to investigate the potential of knowledge based systems by building two prototype applications SERPES, and WADNES. SERPES was an knowledge based system designed to mimic the decision making process carried out by an sewer rehabilitation engineer and WADNES was a system designed to provide expert guidance to control room staff concerning emergency situations that could occur within a water distribution network. These particular areas of research were chosen for knowledge based system development because they were demand to meet criteria established at the outset of the project. These criteria were as follows:

The problems concerned should be ones where human expertise is at a premium and resolution involves significant judgement and experience.

The problems should be such that they contain well documented and closely bounded planning procedures based on detailed formalised knowledge.

That an improvement could be achieved in the levels of service attained and the standardisation of those levels of service as a result of the knowledge based system development/application.

The potential benefits foreseen as following on from knowledge based systems research included, an improvement in the distribution of valuable knowledge within the industry, the achievement of common standards of service and the improved utilisation of experienced staff. The SERPES system was based on the four planning stages of sewer rehabilitation as defined in the Sewer Rehabilitation Manual. This includes the ability to build and calibrate a WASSP [153] network model of the sewer layout from within the SERPES program as well as providing costed solutions to identified problems within the sewer network. The program contains approximately 800 rules which lead the user through the rehabilitation manual performing tasks such as the classification of sewers based on their importance to the functioning of the network. Much of the program is concerned with the provision of graphical tools designed to assist the sewer engineer at various stages of the rehabilitation process, including network displays, graph generation and sewer cross section generation. The WADNES program was intended to demonstrate the relevance of knowledge based system technology in an on line control room situation, a field that by its nature provides the possibility of a multiplicity of knowledge based applications. The modern water network control room is the receiving point for a vast amount of data arriving via a number of signal carrying media from remote sensors distributed throughout the geographic area of the network. This data needs to be categorised, interpreted and acted upon in order to provide optimal network control. The expertise, experience and data assimilating capabilities required by the staff that make the control room decisions are increasing rapidly and a method of avoiding the situation where the available staff are unable to meet the data processing needs of the network is required. Knowledge based systems are seen as providing the means by which the required volume of knowledge can be made available around the clock to control room operators, managers and engineers. WADNES itself was designed to provide advice to a network operator in the event of an abnormal or emergency situation occurring within the network. The program is relatively basic in that it will only handle four particular failure types, a major burst, a pumping station failure, a chlorine gas leak and a request for increased fire fighting capacity. However, the program was intended as a prototype that would demonstrate the capabilities of such a system rather than as a commercially viable product. It is, like SERPES, a rule based system that through a dialogue with the user, generates one or more courses of action to be followed to resolve the emergency situation. In addition, a template matching system of network parameters is used to locate the probable source of a problem within the network based on the effect the particular problem has upon the flows and pressures monitored by the control room telemetry system.

There is significant potential that can be exploited in the use of knowledge based systems within the water industry. However, given the inertia commonly experienced in the industry in the light of innovative developments and the fact that the two knowledge based systems themselves were only of a prototype nature, there is still some considerable distance to go before knowledge based systems become more widely accepted as being of real benefit.

4.3 Types of Knowledge Based System.

Within AI research there are a number of different types of knowledge based system which can be usefully categorised by the methods used to hold the domain knowledge. The most common types of system are rule based systems, this popularity is due largely to the fact that rules are a relatively intuitive form of knowledge representation thereby making the task of structuring the domain knowledge easier. In addition, the fact that a rule is usually a self contained item of knowledge makes the updating and modification of the rule base relatively straightforward. The components of such rule based systems are described in detail below. There are however other approaches to knowledge representation which have yielded significant results, a more object oriented approach is possible with frame based systems such as the IDEAS system described by Winstanley et al [10]. The object oriented view of problem solving is based on identifying the real world objects involved in a problem and the processes carried out by those objects. A simulation of objects and their processes and a means of linking such objects is provided by object oriented programming. An object in such a program can represent anything from an integer to a machine tool, all objects are treated uniformly, they can have a local memory associated with them, they can pass data between each other, they can execute subroutines or processes and they can inherit characteristics from ancestor objects.

4.3.1 Frame Based Systems.

Frames have been used to implement such object oriented programs [10,104], the frames themselves corresponding to objects and the attributes and relationships of these objects are represented by slots within the frames. Frames can be ordered in a hierarchical structure of superclasses, classes and subclasses of objects, so that attributes (slots) and where appropriate values (held within slots) can be inherited. This allows a centralisation of the knowledge held within the frames and avoids the need to specify all the attributes of a new object or individual when it is to be added to the system i.e. The slots in a new instance frame are determined by the class and superclass to which that instance belongs and the values for these slots can be passed down the inheritance tree. Figure 4.31 illustrates the principle of inheritance.

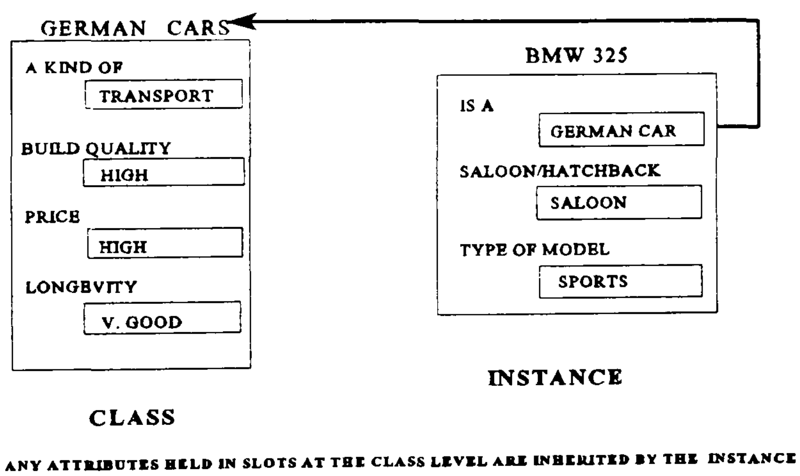


Figure 4.3.1 Inheritance in a frame based system.

Information can be retrieved from a frame based system by matching template frames with those stored in the knowledge base. It may be necessary to move down the hierarchy of frames to retrieve specific values via the use of pointers in frame slots, alternatively such frame slots can hold equations or instructions to trigger outside procedures to calculate the required information. The advantages of such hierarchical systems are that they are initially straightforward to construct once the relations between the objects involved have been established, they are also easier to update and modify in a consistent way when required and mistakes can be corrected again with consistency ensured by the hierarchical structure.

There have been applications where combinations of knowledge representations have been used in an attempt to exploit the relative strengths of each. Atkins [7] used a frame based system to hold the static information relating to a problem, while the more dynamic information was held in production rules which could alter the values held within the frame slots. The G2 system uses multiple knowledge representations, objects and their relationships are represented by a frame system, these frames can also contain production rules within their slots along with information on where and when to apply them, dynamic models can be stored and used to represent system behaviour over time and real time executable procedures can carry out tasks required by the system.

An extension of the idea of using multiple knowledge formats tailored to suit the data being stored and manipulated, is the concept of the Blackboard System [64]. The general methodology used by such systems is that of opportunistic reasoning, each knowledge source contributes what it can, when it can to an evolving solution held in a processing area termed the blackboard. The information on the circumstances in which a particular knowledge source can contribute its knowledge is held with the knowledge source itself but contributions are controlled by an overall monitoring system that activates the appropriate source at the appropriate time. In many respects the use of the ARIMA, neural network and rule base elements of the demand forecasting application described in this thesis can be viewed as a form of blackboard system. Each component was developed to provide the most suitable approach to the individual elements of the problem of deriving an accurate demand forecast, the ARIMA algorithm exploits the cyclic periodicity characteristic of water consumption patterns, the neural network provides a method of accounting for the highly non-linear effects of weather influences and the rule base is designed to provide a method of capturing the heuristic knowledge commonly used by operators to modify forecasts.

4.3.2 Rule Based Systems.

As outlined in section 4.2.1 rule based systems[19,31] have three basic components, the rule base itself where the domain knowledge is held, the working memory which holds the data which describes the current situation and inference engine or rule interpreter which controls the selection and firing of the appropriate rules.

The structure of the rules held within the rule base is commonly of the form shown overleaf.

RULE *N*

IF *ANTECEDENT1*

AND *ANTECEDENT2*

AND *ANTECEDENT3*

....

THEN *CONCEQUENT1*

AND *CONCEQUENT2*

AND *CONCEQUENT3*

....

Antecedents and concequents of rules can contain constants as identifiers, in which case a rule must be created for each instance likely to be encountered by the system, for example:

IF John lives next to Frank THEN John is a neighbour of Frank

The above rule is restricted in its application to the characters John and Frank and we would need additional rules to deduce if other persons that are likley to be encountered were neighbours. To avoid this problem rules can contain variables which allow then to have far more general application, as illustrated below:

IF x lives next to y THEN x is a neighbour of y

These variables initially have no value but acquire values as the antecedent patterns are matched to the information in the working memory, the binding of a variable with a particular value is termed instantiation. The patterns specified in the antecedents of a rule are matched by the rule interpreter to the assertions held in the working memory (an assertion in this context taking a form such as 'The current temperature is 23 Degrees Celsius') , if the match succeeds then the antecedent in question is said to be satisfied. If all the antecedents that make up the 'if' part of a rule are satisfied by successfully matching assertions in the working memory, then the rule is triggered. The concequents in the 'then' part of a triggered rule can either have the effect of establishing a new assertion in the working memory and hence add to the information available to enable progress towards a solution, in which case the rule based system is described as a deduction system i.e. the rules deduce facts from an existing pool of information and add the deduced facts back into the pool thereby moving incrementally towards a solution. Alternatively, the concequents of a triggered rule can carry out some action such as the adjustment to the values of a vector or the alteration of the concentrations in a chemical production process, such systems are termed rule based reaction systems. The actions taken by such systems can also include the activation/deactivation of specific rules or groups of rules as well as the addition or deletion of assertions. The demand forecasting application described in this chapter is a reaction system in that the rules carry out adjustments to the twenty four hour prediction.

In both rule based reaction systems and deduction systems, forward chaining is the process that moves the system from the initial state to the solution state. A forward chaining system is a data driven system, with the antecedents of each rule being compared to the assertions that exist in the working memory. Successful matches between assertions and a rules antecedents cause the triggering of that rule and an updating of the information in the working memory. An examination of the working memory reveals that either progress has been made towards a solution in which case further rules can now be invoked, or that the solution itself has been reached. If no progress has been made, then this indicates that the information we have at our disposal, in the form of assertions in the working memory, is insufficient to allow a solution to be deduced.

This contrasts with rule based systems which are goal driven, these systems are termed backward chaining systems. Such systems start with a initial hypothesis being made as to a possible solution and the rule base is examined to match any rules whose concequents would confirm the proposed hypothesis. If the search is successful, the rule in question is extracted and its antecedents matched against the assertions held in the working memory, if a positive match is found for each element of the list of antecedents, then the rule's concequents are confirmed and the original hypothesis shown to be correct. If an incomplete match is made between the rule's antecedents and the available data in the working memory, then the antecedent elements that were not matched are extracted to form sub goals which initiate a further search of the rule base to locate rules whose concequents will confirm the missing information. This process continues until all the information required is assembled and hence the hypothesis confirmed, or it proves impossible to match the required facts with the available data and the hypothesis fails.

The guiding factor in determining whether a data driven or goal driven approach is most appropriate to a specific problem is an examination of the way the rules relate facts to conclusions. When a typical set of available facts could lead to many plausible conclusions, of which only a small number will be of interest, then the rule system exhibits a high degree of 'fan out' i.e. there is potentially a large number of branches and sub branches which would have to be explored before arriving at a desired solution. Such a situation would indicate that the most suitable approach would be backward chaining from a solution known to be one of the small candidate set of solutions of interest. However, if the number of ways of reaching the particular conclusion in which you are interested is large, but the number of conclusions you are likley to reach given a typical set of facts is small, then the system is said to exhibit 'fan in' and forward chaining is indicated. Similarly, if the situation is such that you initially possess all the facts that are ever likley to get and you wish to know everything it is possible to conclude from these facts, then forward chaining is again indicated. This is the situation found in the demand forecasting application described in this chapter an hence a forward chaining rule based reaction system has been adopted.

Such forward chaining reaction systems require a method of conflict resolution to enable them to determine the order in which to implement the concequents of the list of rules that have been triggered by the information present in the working memory. The determination of the order in which a set of triggered rules are fired can have a very significant effect upon the results of a reaction system. For example, the demand forecasting application may have two rules, one of which boosts the demand forecast to account for the occurrence of very hot weather on a Bank Holiday and the other alters the basic 24 hour prediction profile to reflect the characteristically unique shape of a Bank Holiday profile. If both these rules are triggered on a particular hot Bank Holiday, then the order in which they are applied to the basic prediction profile has a significant effect upon the final prediction, this is illustrated by figures 4.3.2 and 4.3.3.

There are several methods for achieving conflict resolution some of which are listed below.

Rule Ordering - this is the simplest form of conflict resolution, the order that the rules occur within the rule base determines the firing order.

Context Limiting - here the rules are separated into groups of related rules and only some of the groups are active at any one time.

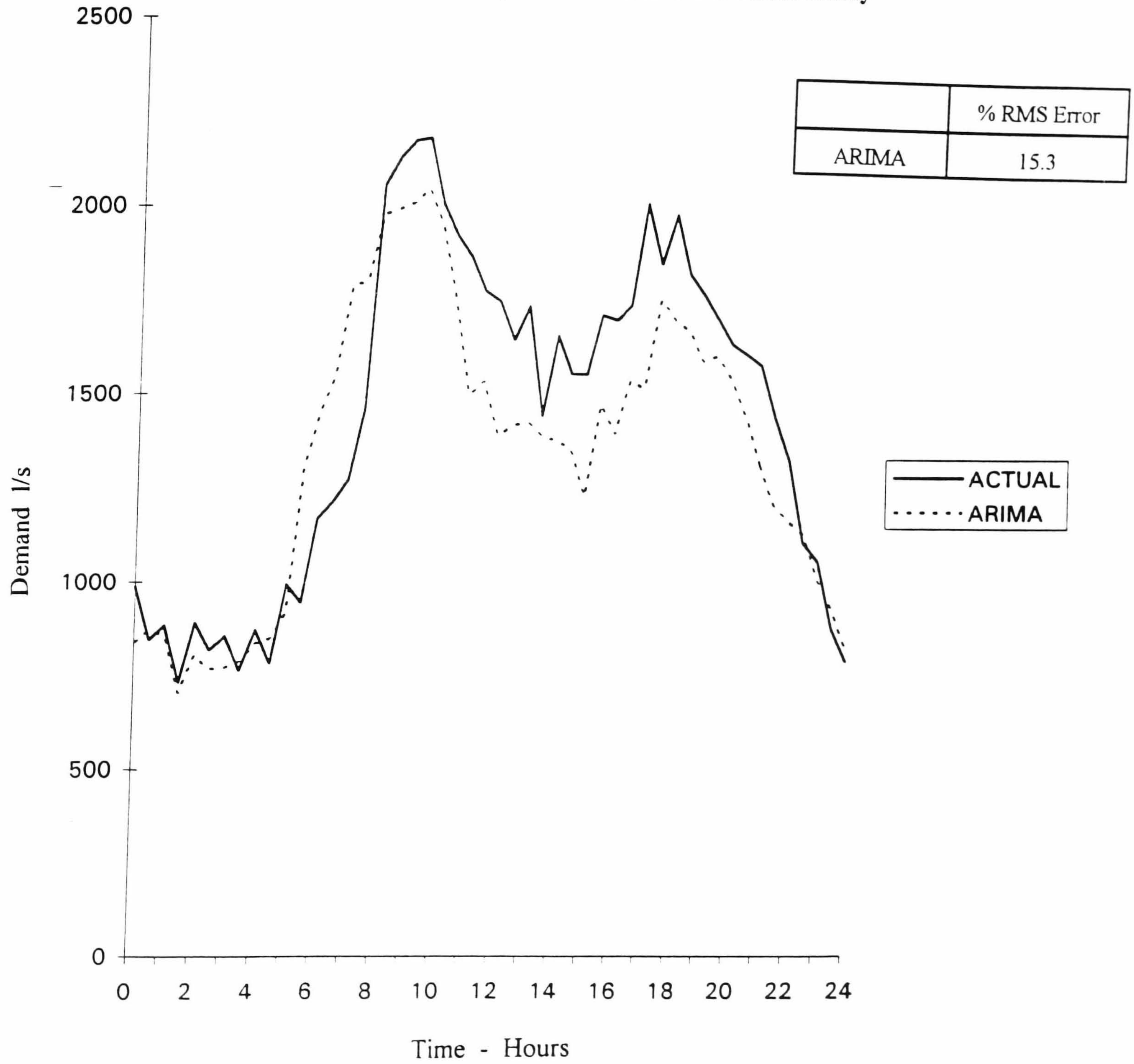
Specificity Ordering - when the conditions of one rule are a superset of the conditions of another, then use the superset rule as it is the more specific.

Input Data Ordering - the available facts are ranked in a prioritised order and the rule that uses the highest priority assertions in its 'if' part is fired first.

Rule Priority Ordering - each rule is give a numeric value which indicates its firing priority.

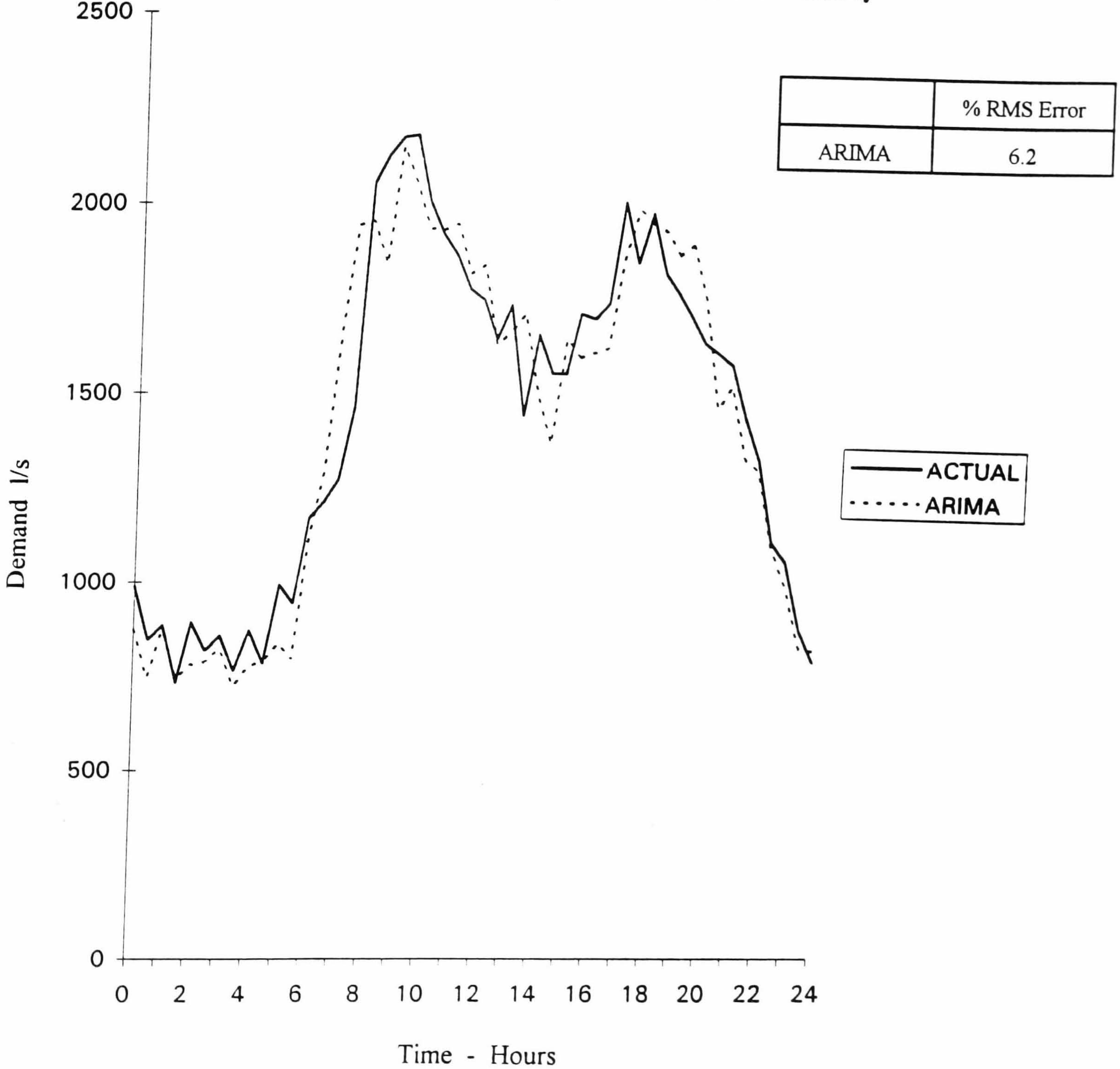
The demand forecasting rule interpreter uses an combination of context limiting and rule priority ordering in that the rules are divided into one of three categories, calendar related rules, network related rules and weather related rules and within those categories each rule is assigned a numeric priority.

Figure 4.3.1 Rule For Hot Day Fired Before Rule For Bank Holiday



	% RMS Error
ARIMA	15.3

Figure 4.3.2 Rule For Bank Holiday Fired Before Rule For Hot Day



4.4 Implementation of the Demand Forecasting Rule Base.

There are several programming environments and languages that have been developed specifically to provide a suitable platform for the construction of artificial intelligence programs. The demand forecasting application described here was implemented in the POPLOG environment using the POP-11 programming language [10]. The decision to implement the rule based elements of the demand forecaster in an AI environment/language was based on the wish to use the most appropriate tools for the nature of the tasks to be undertaken. Hence the mathematical calculation required by the ARIMA algorithm dictated that a numerical computation biased programming language such as FORTRAN be used for its execution, while the more heuristic nature of the task of implementing the rule base was best approached by making use of an environment specifically designed for such problems. There are several advantages that artificial intelligence environments and languages offer, for example, rapid prototyping is facilitated, programs can easily be created, modified, tested and extended, this is necessary because of the ill-defined nature of the problems commonly encountered in AI research. The environment provides a built in editor, on line help and a comprehensive set of debugging aids. Programs can be built up in a modular form, with each module able to be tested and run independently. There is also an extensive range of built in library functions available in POP-11 that are designed to aid the tasks commonly required in AI programs, of these functions, the pattern matching facilities are very powerful and because of their importance in the demand forecasting application, they are described in more detail in section 4.4.1.

In order to allow the use of the POPLOG environment for the implementation of the rule base, a means of communication between the host FORTRAN program and the POP-11 program needed to be developed. This was achieved by running the POP-11 program from within the FORTRAN program as a spawned sub-process and using the 'mailbox' facility provided by the VMS operating system to allow data transfer between the two programs (a mailbox is a virtual device that can be used to send data between VMS processes). Initially the mailbox is created by the FORTRAN program, a mailbox being a file type that can be written to and read from by more than one program type i.e. the FORTRAN program can place data in the mailbox and POP-11 can access that data, modify it and write it back into the mailbox. A system dependent process command in FORTRAN causes the spawning of the POP-11 program as a sub-process by triggering a DCL file that holds the commands that activate the spawned sub-process. FORTRAN uses the system input/output commands to write the data required by the POP-11 program to the mailbox, the required data being, the date of the prediction day, the array holding the 48 data points of the 24 hour demand prediction and the meteorological data for the prediction day in the form of maximum temperature, number of hours of sunshine, total rainfall, the number of antecedent dry days and the number of antecedent days where the temperature exceeds a threshold value. POP-11 uses its 'sysread' commands to extract the data from the mailbox and the rule interpreter initiates the search of the rule base to locate any rules that are applicable to the prediction day. The prediction is modified by the selected rules and then written back to the mailbox where it is available to the FORTRAN program for graphical display. Figure 4.4.1 illustrates this process.

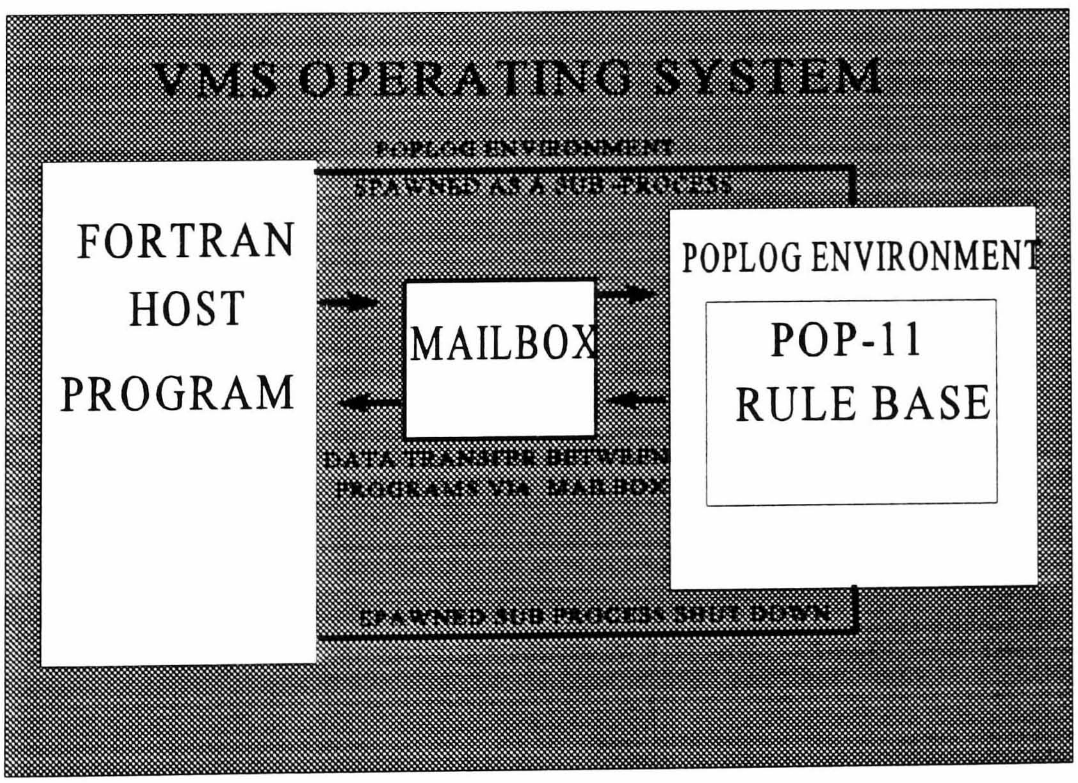


Figure 4.4.1 Communication between FORTRAN and POP-11.

Once the POP-11 program receives the prediction data from the FORTRAN program, a menu driven process is initiated to determine what rules, if any, are to be invoked to modify the raw prediction profile. The menu driven process allows the operator to control the selection of rules for application to the current demand forecast both in terms of the rules triggered as a result of matching the meteorological and calendar data associated with the current forecast, and the rules to be selected manually for application based on the operators experience and judgement. As stated previously the pattern matching facilities offered by POP-11 are highly important in both the above rule selection procedures, it is therefore worthwhile reviewing them in more detail in the following section.

4.4.1 Pattern Matching in POP-11.

POP-11 provides a powerful data structure known as a list, which can be used to represent and manipulate information. It is this data structure upon which POP-11's pattern matching capabilities are based. Lists are denoted in POP-11 by square brackets and can contain any of the POP-11 data types - numbers, words, strings, variables, other lists etc. Some examples are given below.

[a list of five words]	- A list containing five elements of data type word.
[1 2 3 456 78 9]	- A list of six numbers.
[1 2 3 cat dog mouse 4 5 6]	- A list containing a mixture of words and numbers.
[[1 2 3][cat dog][34.23 23.44]]	- A list of lists.
[]	- An empty list.
['a string within a list' {1 4 5 3 5 5}]	- A list containing a string and a vector.

For the demand forecasting application a list structure was used to store the rules that modify the imported prediction, the structure of the list used is shown below.

```
[ rule_no 'rule_name' rule_type manual_trigger priority 'date' [ day_type
 temp sun rain no_dry_days no_hot_days season ] {weight_vector} ]
```

The elements that make up the rule template are as follows:

rule_number - A unique identification number for each rule of the format 012C or 005W, where the letter following the three figure reference relates to the rule category C = Calendar, W = Weather and N = Network.

rule_name - A string of maximum length 30 characters that provides a name that indicates the purpose of the rule, for example 'bank holiday profile'. This string is utilised in providing the operator with a meaningful description of the rules action, both for rule base examination and in the provision of an explanatory list of the rules applied on a particular day.

rule_type - A numeric figure 1 - 3 that is used to identify the rule as calendar, weather or network type and is used in conjunction with the priority figure to determine the order of firing of those rules that have been triggered. More details of this process are given later in this chapter.

manual_trigger - A single number that can have the value 1 or 0 that is used to indicate those rules that have been selected by the operator for manual triggering on the current prediction day i.e. Although no match from the available information passed to the database would indicate triggering of the rule, the operator considers that from his experience the rule should be invoked.

date - A string holding the date or dates upon which the rule is applicable.

Meteorological data relating to the rule is held as a list within a list and is composed of the day type (1 to 4 as used by the neural network predictor - see Chapter 5), the maximum temperature, the number of sunshine hours, the total rainfall, the number of antecedent dry days, the number of antecedent hot days and a figure 1 to 4 to indicate the season. The season indictor is used to ensure rules operate during the particular season or seasons they were constructed for.

The above data in effect compromises the conditional 'if' part of each rule, while the vector described below forms the resultant 'then' part.

weight_vector - This is a vector of length 48 that contains the weight values used to alter each of the 48 half hourly data points of the raw prediction when the rule is fired. Each element of the weight vector is a numeric value that can be either posotive or negative and is applied by adding each vector element to the corresponding value of the 24 hour demand forecast.

In order to select the rules that are applicable on any given prediction day, a number of built in POP-11 matching procedures and functions are utilised. The built in POP-11 pattern matcher provides a means of checking the correspondence of a list with a pattern, it has a general syntax of:

```
< target list >  matches  < pattern >
```

A number of symbols can be used to allow complex patterns to be specified. The = and = = symbols are the basic descriptors of pattern shape and can be used to allow 'wild card' matching to a single element or a number of elements in the target list. For example, in order to determine if the date in a rule with a structure as outlined on the previous page matches the current prediction date, the following pattern specification could be used.

```
rule_1 matches [ = = TUE_12_MAY_1990 = = ]
```

The = = symbols either side of the date will match with any number of list elements, so in effect the above statement will return an answer of true if any element of rule_1 matches TUE_12_MAY_1990.

More complex patterns can be constructed that specify not only the linear shape of the of the target list but also its structural content. In the example given below the date TUE_12_MAY_1990 must be followed by a list (the meteorological data list in this case) whose second and fourth elements are 20 and 0.1 respectively.

```
rule_1 matches [ = = TUE_12_MAY_1990 [ = 20 = 0.1 = = ] = = ]
```

In order to allow variables to be used in pattern specification the ^ symbol is used to prefix the variable name to denote that the current value of that variable should be used in the pattern match instead of the literal values of the name itself. This can be illustrated by the following:

THU_14_MAY_1990 ~> pred_date;

The date is assigned to the variable pred_date.

rule_1 matches [= = pred_date = =]

This match fails.

rule_1 matches [= = ^pred_date = =]

This match succeeds.

The ^^ symbol acts in a similar manner to ^ , except that it is used to represent the elements of list variables.

As well as checking that a list conforms to a particular pattern, it is often necessary to retrieve values from the target list. The ? and ?? symbols are used to bind the variables whose names are preceded by ? or ?? to the values of elements of the target list. Hence, ?x means set the value of x to the value of the single element in the target list to which it matches and ??x means set the value of x to the list of elements in the target list to which it matches.

```
THU_14_MAY_1990 ~> pred_date;
rule_1 matches [ = = ^pred_date [ = ?get_temp ?get_sun ?get_rain = = ] = =]
```

In the above example the value of THU_14_MAY_1990 is assigned to the variable pred_date, if a successful match is made between this variable and the date in rule_1 then the variables get_temp, get_sun and get_rain would have their values set to the values of temp, sun and rain contained in rule_1.

It is possible to carry out restricted matching with the use of the : symbol in conjunction with the ? symbol, this means that the specified element(s) in the target must be a member of a restricted set of possible answers.

```
rule_1 matches [ == ?x:bank_holidays == ]
```

In the above example a match would be successful if the date element in rule_1 was one of the dates listed in the following previously defined procedure.

```
define bank_holidays ( date ) -> result;
   member ( date, [ MON_07_MAY_1990 MON_28_MAY_1990 ]) -> result
enddefine;
```

4.4.2 The POP-11 Database.

As an extension to the pattern matching facilities outlined above, POP-11 provides a built in database and numerous data manipulation procedures for this database. This database facility has been utilised in the construction of the demand forecasting application. The database provides a simple mechanism for storing the collection of rules (of the format previously specified) and retrieving the data in those rules on a pattern matching basis. The database takes the form of a list of lists and the various pattern matching procedures described in the previous section are utilised for adding extracting and examining data held within it, as well as a number of specialised database matching facilities.

Initially on entering the POP-11 program the database is empty and the existing rules of the format shown in section 4.4.1 are loaded into the database from a datafile.

```
datafile( 'rulebase.dat' ) > database;
```

The built in procedures add and alladd allow single items and multiple items respectively to be appended to the database, so that if a new rule is constructed and held in the variable 'rule_10', it could be added to the database by the command:

```
add ( ^ rule_10);
```

The list that comprises rule_10 is consequently added to the list of lists that forms the database. Similarly the procedures remove and allremove allow the deletion of single and multiple items from the database, hence if a rule became obsolete it could be removed from the database.

```
remove ( [ 006W = = ] );
```

The above command would remove the rule with a rule number that matched 006W.

Several procedures exist that allow the searching of the database to locate items that match specified patterns. The procedure present takes a pattern, matches it against every item in the database and returns either true or false as a result depending on the success or otherwise of the attempted matching. In addition, if a match is found the procedure places the matched item from the target list in the variable 'it'. A illustration of the use of the present procedure in the demand forecasting application is the extraction of a rule for alteration of the weight vector. In the program extract shown below a search is made of the database to locate a rule whose first element matches the value of the variable rule_no, if successful then the variable weight_vector is assigned the value of the rule weight vector and the matched rule is removed from the database. The procedure alter_weights is passed the extracted rule and the weight vector and returns the updated rule, this updated rule is then added to the database.

```
if present ( [ ^rule_no = = ?weight_vector ] ) then
        remove ( it );
        alter_weights ( it , weight_vector ) > updated_rule;
        add ( ^updated_rule );
endif;
```

Two other important procedures used extensively in the demand forecasting application for the extraction of data from the database are the foreach and forevery procedures. These procedures allow the retrieval from the database all items present that match a pattern or list of patterns. This is in contrast to the present procedure in that present will stop when the first occurrence of a match is found, whereas foreach and forevery continue to search the database until all occurrences of the required match have been extracted. The general syntax for these procedures is as follows:

```
foreach < pattern > do < action > endforeach
forevery < list of patterns > do < actions > endforevery
```

An illustration of the use of the foreach procedure in the demand forecasting application is in the searching of the database to locate any rules of type calendar whose values in the date element match the current prediction date.

```
1  > rule_type;
foreach [ ?rule_no ?rule_name ^rule_type = ?priority ^current_date [ = =] ?weights ] do
        [ ^rule_no ^rule_name ^rule_type ^priority ^weights ]:: trigg_rules  > trigg_rules;
        num_rules + 1   >  num_rules;
endforeach;
```

In the above program extract the rule type is set to 1 to indicate a search for calendar related rules, the foreach procedure then searches the database for all the rules that have a rule type of 1 and a date that matches the variable current_date. The ? symbols are used to prefix those variables we wish to extract values for, the extracted values are then appended to a list of triggered rules. Similar searches are made of the database using the appropriate patterns to locate rules of both network and weather type and so build up a comprehensive list of rules applicable to the current day.

4.4.3 The Operation of the Demand Forecasting Rule Based System.

The application can be divided into two key elements, the rule base which utilises the built in POP-11 database described in the previous section and the inference engine that controls the operation of the system from importation of the initial data to generation of the final result. The tasks carried out by the inference engine are described below.

The initial task is to import the data placed in the mailbox by the ARIMA and neural network prediction programs, this comprises the 24 hour prediction profiles generated by both predictors, the prediction day date the current meteorological data and a record of the past rainfall pattern and temperature variations. The built in POP-11 database is then loaded with the latest version of the rule base from the appropriate data file. The menu driven interface with the operator is then activated, a schematic showing the operation of this menu driven system is displayed in figure 4.4.2. The first menu prompts the operator to decide whether the rule base is to be used for the current prediction, if it is not then the POP-11 program returns control to the FORTRAN program, if it is then menu 2 is displayed. Menu 2 allows the operator to proceed with the search of the rule base for rules triggered by the current data relating to the prediction day, or to manually select rules that the operator considers should be activated on the prediction day, or the operator can select the examine rule base option to get an overview of the current state of the rule base. The manual selection option is particularly important for the Network rule category, the reason for this is related to the nature of the rules themselves. For example a rule may well have been constructed for accounting for the effect of a hosepipe ban being imposed during a long dry spell. However the operator will not know in advance when such a ban would be enforced so no date can be pre entered in the rule to trigger it automatically, the operator must instead manually select the rule for application at the time the ban is introduced.

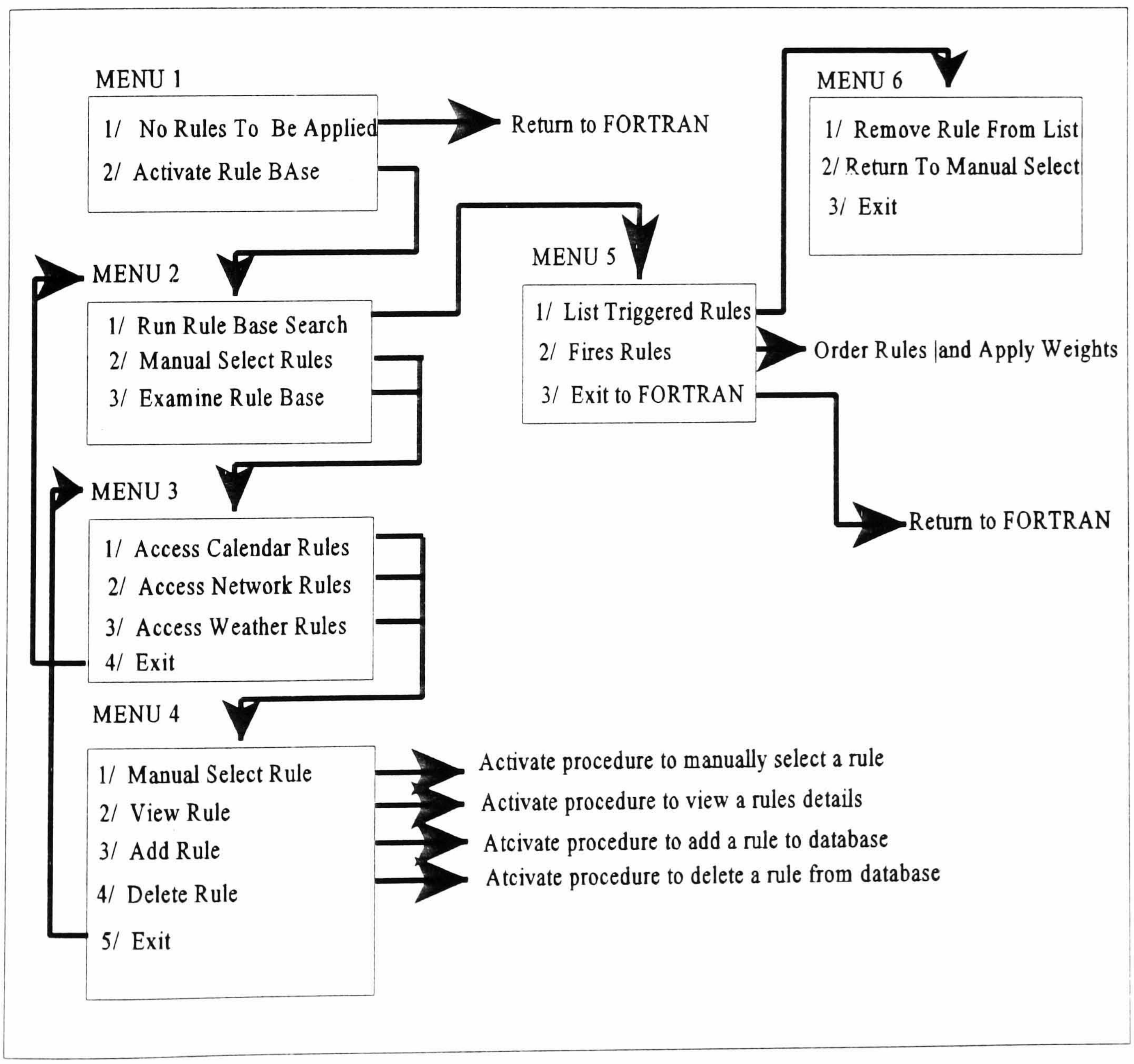
Selecting either the manual select or examine rule base options of Menu 2 activates Menu 3 which provides the choice of accessing the rules of type Calendar, Network or Weather. Menu 4 provides options which allow rules to be viewed, altered, added to and deleted from the rule base, in each case the operator is prompted via rule templates to enter the data required. The manual select option in this menu is either enabled or disabled depending on whether the manual select or examine rule base option was chosen by the operator in Menu 2. Once the desired changes have been completed via the options of Menu 4, control is returned to Menu 2 where the 'Run Rule Base Search' option can be selected.

The searching of the database is carried out for each rule type and for each combination of available data using the search techniques outlined in the previous sections. The Calendar and Network rules are searched for using primarily the date of the prediction day as the pattern to be matched while the Weather related rules are selected by a number of searches based on the current season and the meteorological data. An example of such a search based on the current day type (as passed from the neural prediction program) and the number of antecedent dry and hot days is given below.

```
3  > rule_type;
```

```
foreach [ ?rule_no ?rule_name ^rule_type = ?priority = [ ^current_day_type = =
        ^current_dry _days  ^current_hot_days =  ] ?weights ] do
      [ ^rule_no ^rule_name ^rule_type ^priority ^weights ]:: trigg_rules  > trigg_rules;
      num_rules + 1   >  num_rules;
endforeach;
```

Figure 4.4.2 Schematic to show the operation of menu driven interface.



Once the searching of the database for the triggered rules has been completed Menu 5 gives the operator the option to list the currently triggered rules by rule number and name and if desired he can elect to remove one or more rules from this list, alternatively he could return to manually select any rule he considers should have been present in the list. Once the operator is happy with the list of rules to be applied the 'Carry out profile modification' option issues the instruction to the inference engine to carry out the ordering of the triggered rule list and apply the weights to the raw prediction profile.

The importance of the correct ordering of the triggered rules was illustrated by figures 4.3.2 and 4.3.3 and this is carried out by the inference engine on the basis of the rule category and the rule priority figures associated with each rule. The nature of the categories of the rules dictate that the Calendar rules are fired first followed by the Network rules and lastly the Weather rules. The reasoning behind this is that the effects accounted for by the Calendar rules and to a lesser extent the Network rules, are ones that alter the type of profile displayed by the day in question. For example a Bank Holiday Monday has a profile that is peculiar to this type of day i.e. people get up later than on a normal Monday, commercial and industrial usage is reduced and activities such as washing are increased. We therefore need to arrive at the right basic profile for a Bank Holiday Monday before we start superimposing upon that profile a) the effects of Network related events such as hosepipe bans and b) the effects of the Bank Holiday weather. The rule priority figure allows for the ordering of rules within each of the three categories, for example the firing of a Network rule to account for the increase in consumption due to the adding to the system of a re-zoned area before the firing of another Network rule that accounts for a large burst in that area.

Once the rules have been ordered for firing their corresponding weight values are applied to the raw prediction profiles for both the ARIMA and neural network generated predictions. The modified predictions are then passed back to the VMS mailbox so that they can be accessed by the FORTRAN program for graphical display.

4.5 Results

The POP-11 rule base was tested on water consumption data for the Slough and High Wycombe areas covering a period from 1st April 1990 to 16 June 1990.

4.5.1 Calendar Related Effects

The calendar related effects cover events such as the following, Bank Holiday Mondays (3 occurrences during year), Bank Holiday weekends, the Easter Holiday (4 days), the Christmas and New Year holidays (10 days), the BST/GMT changeovers and in popular holiday areas there are significant effect during the peak school holiday periods. The available half hourly consumption data for the Slough and High Wycombe areas covered a period which contained a BST/GMT changeover, the Easter Holiday and the two Spring Bank Holidays, rules were therefore constructed to account for each of these events. Figures 4.5.1 and 4.5.2 show the 24 hour profiles for the two Spring Bank Holiday Mondays, in each case the three profiles shown are 1) the actual consumption on the Bank Holiday 2) an unaltered ARIMA based prediction for the day and 3) a neural network based prediction which has been modified by the rule base. The ARIMA profiles are in effect profiles for a normal Monday and show a considerable difference in shape to both the actual consumption profile which is much closer to the profile updated by the rule base (in this case the same rule was applied for each occurrence of the Bank Holiday). Figures 4.5.3 - 4.5.6 show the profiles for the four days covering the Easter Holiday i.e. Good Friday, Easter Saturday, Easter Sunday and Easter Monday. Again the actual consumptions are compared to unaltered ARIMA prediction profiles and neural network generated profiles that have been modified by the rule base. Figure 4.5.7 shows the effect of the BST/GMT changeover on March 25 1990, the actual consumption is shifted by one hour compared to the unaltered ARIMA profile. In this case the rule triggered by the occurrence of the changeover does not apply weights to the prediction profile but instead triggers a FORTRAN routine that shifts the data used to generate the prediction by one hour and then re-runs the prediction to arrive at a correct profile.

Figure 4.5.1 Demand Profiles for Bank Holiday 7th May 1990

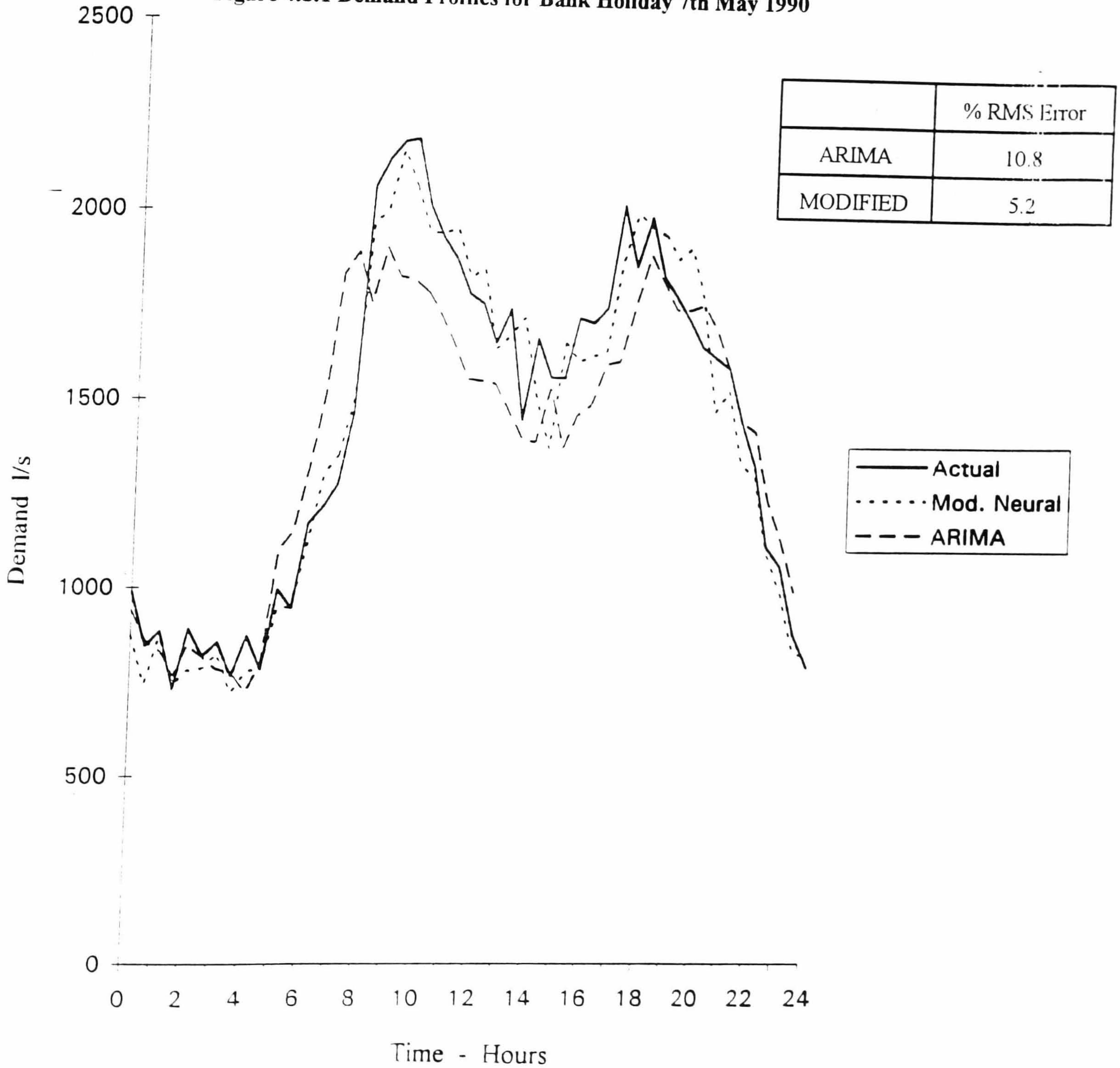
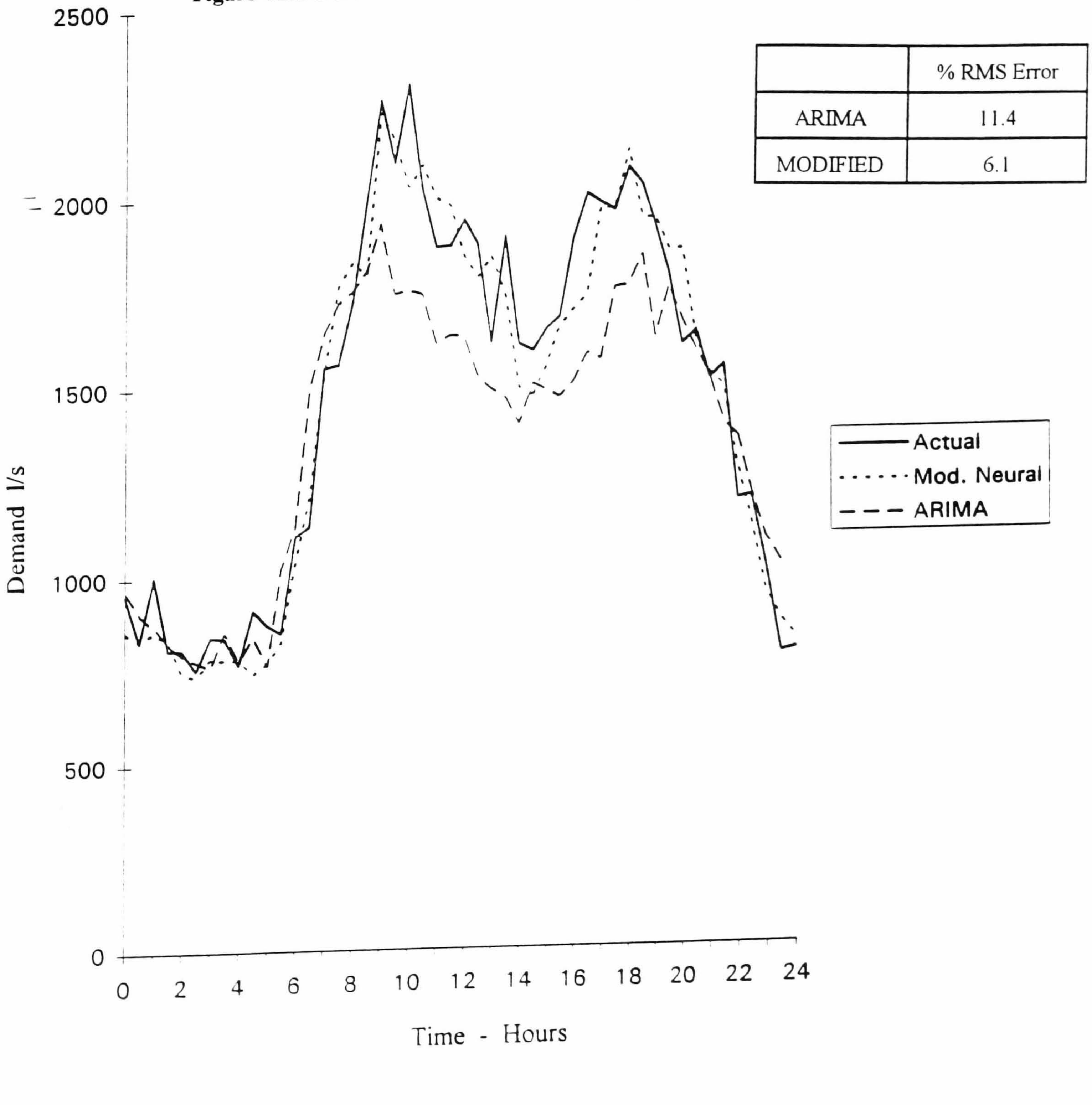
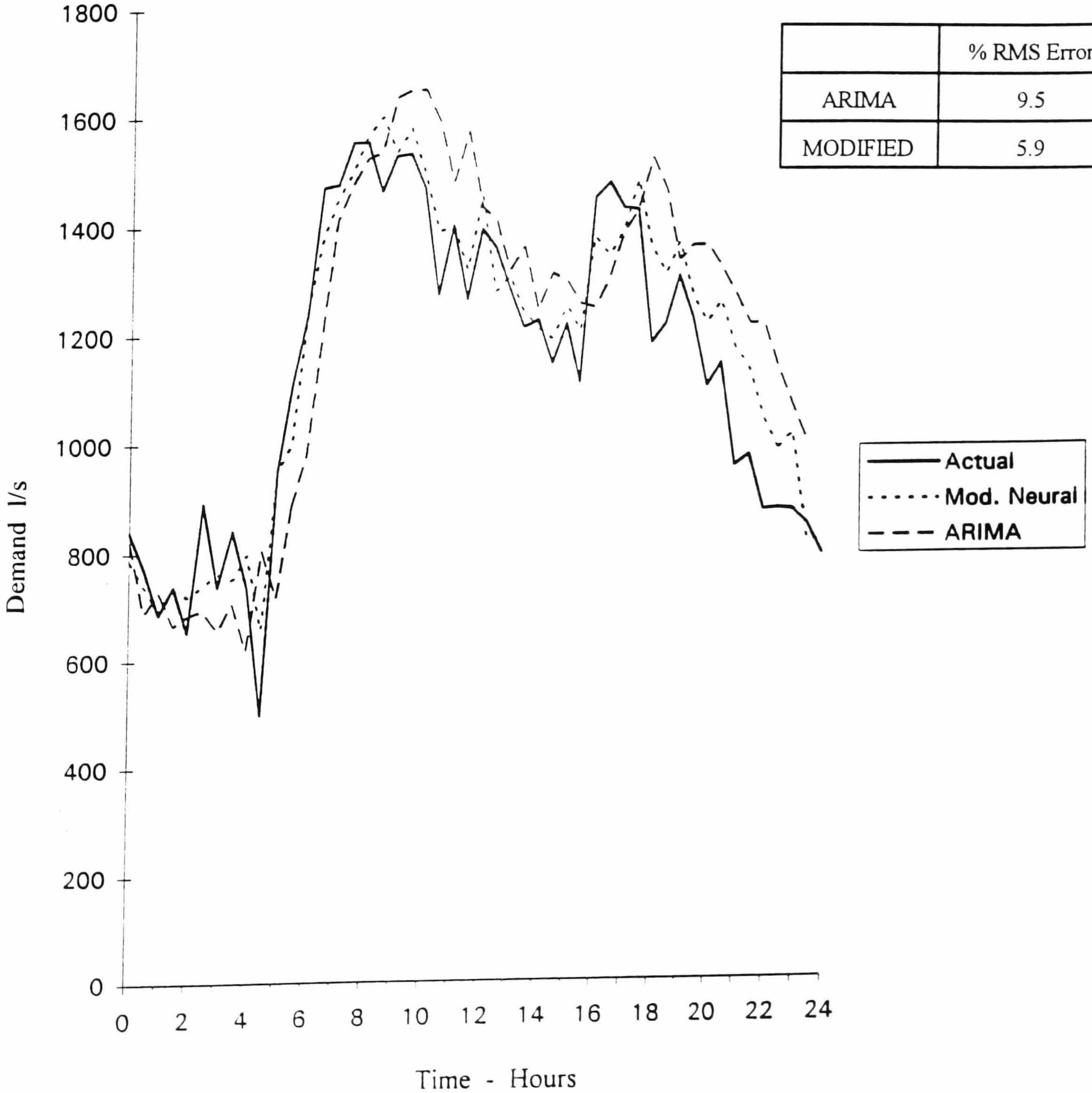


Figure 4.5.2 Demand Profiles for Bank Holiday 28th May 1990



	% RMS Error
ARIMA	11.4
MODIFIED	6.1

Figure 4.5.3 Demand Profiles for Good Friday 1990



	% RMS Error
ARIMA	9.5
MODIFIED	5.9

Figure 4.5.4 Demand Profiles for Easter Saturday 1990

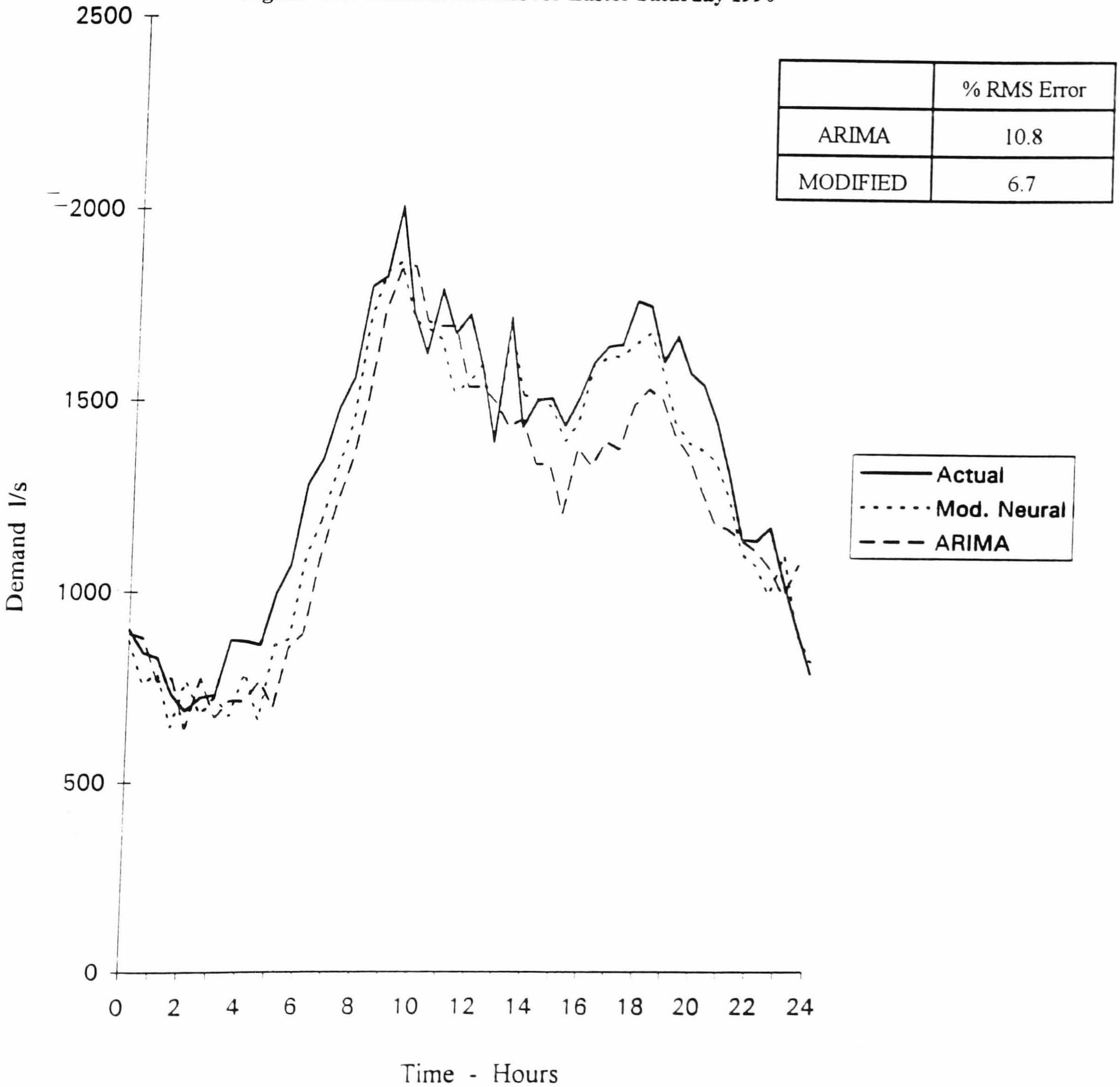


Figure 4.5.5 Demand Profiles for Easter Sunday 1990

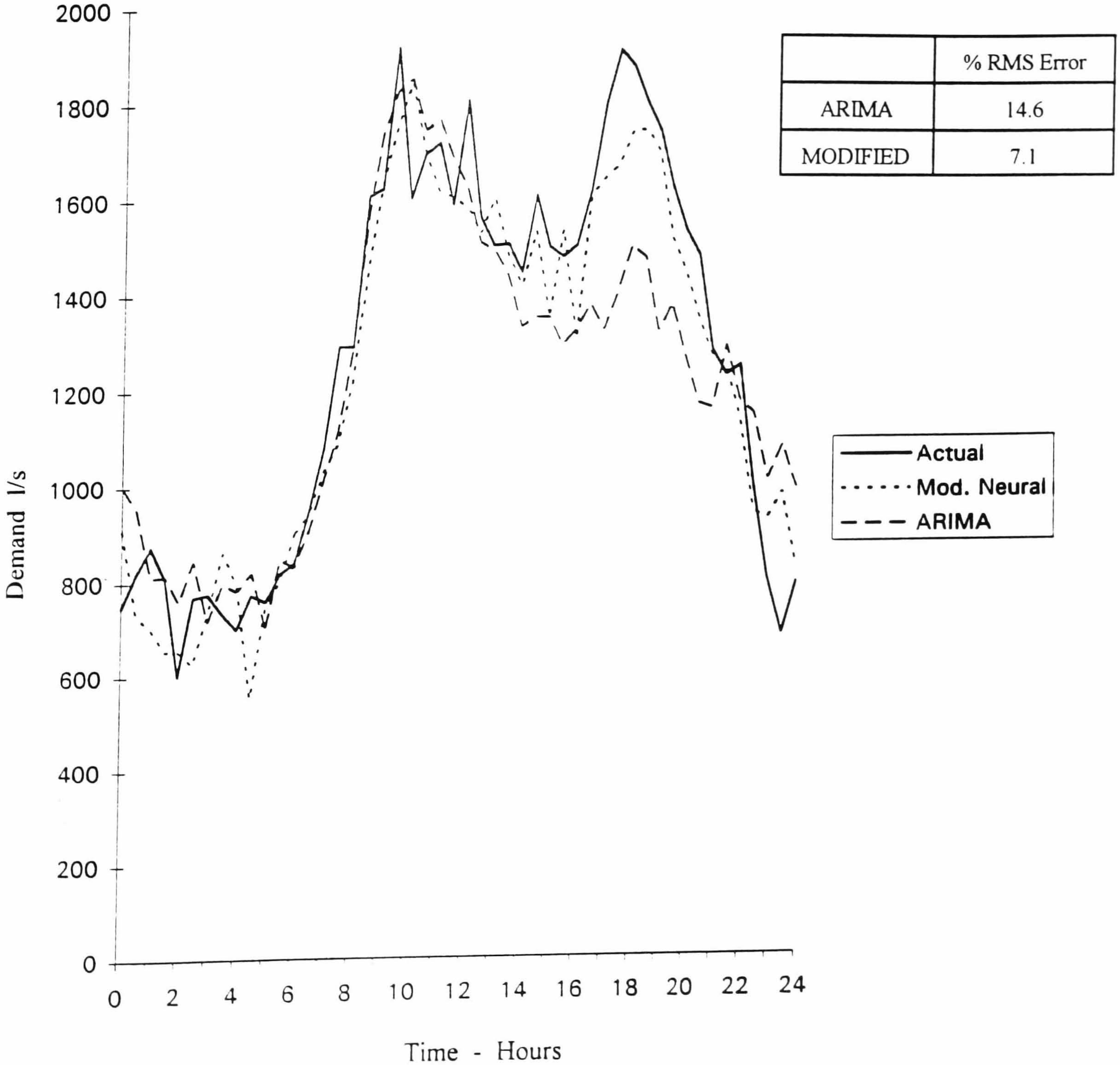


Figure 4.5.6 Demand Profiles for Easter Monday 1990

	% RMS Error
ARIMA	9.2
MODIFIED	6.8

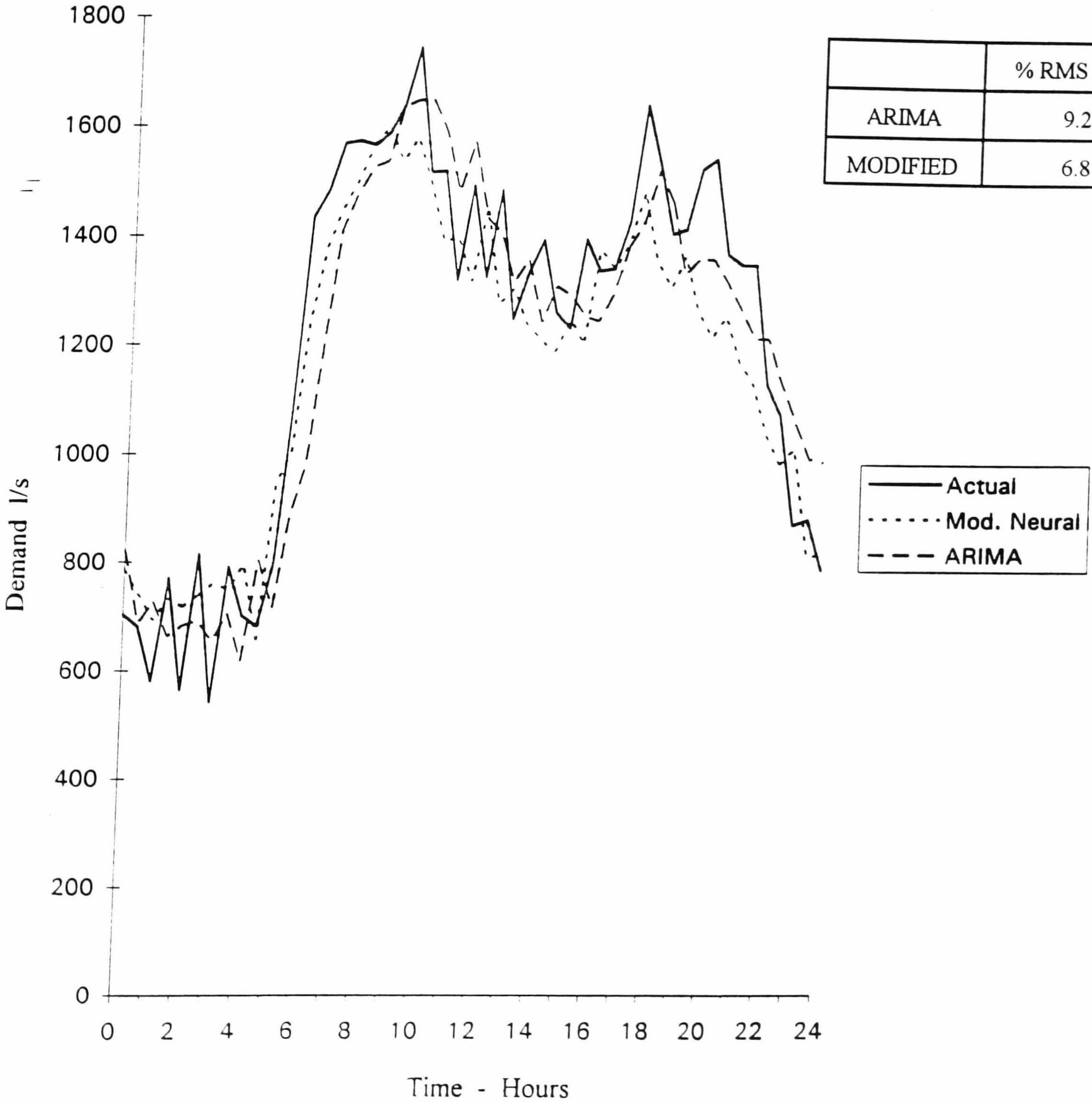
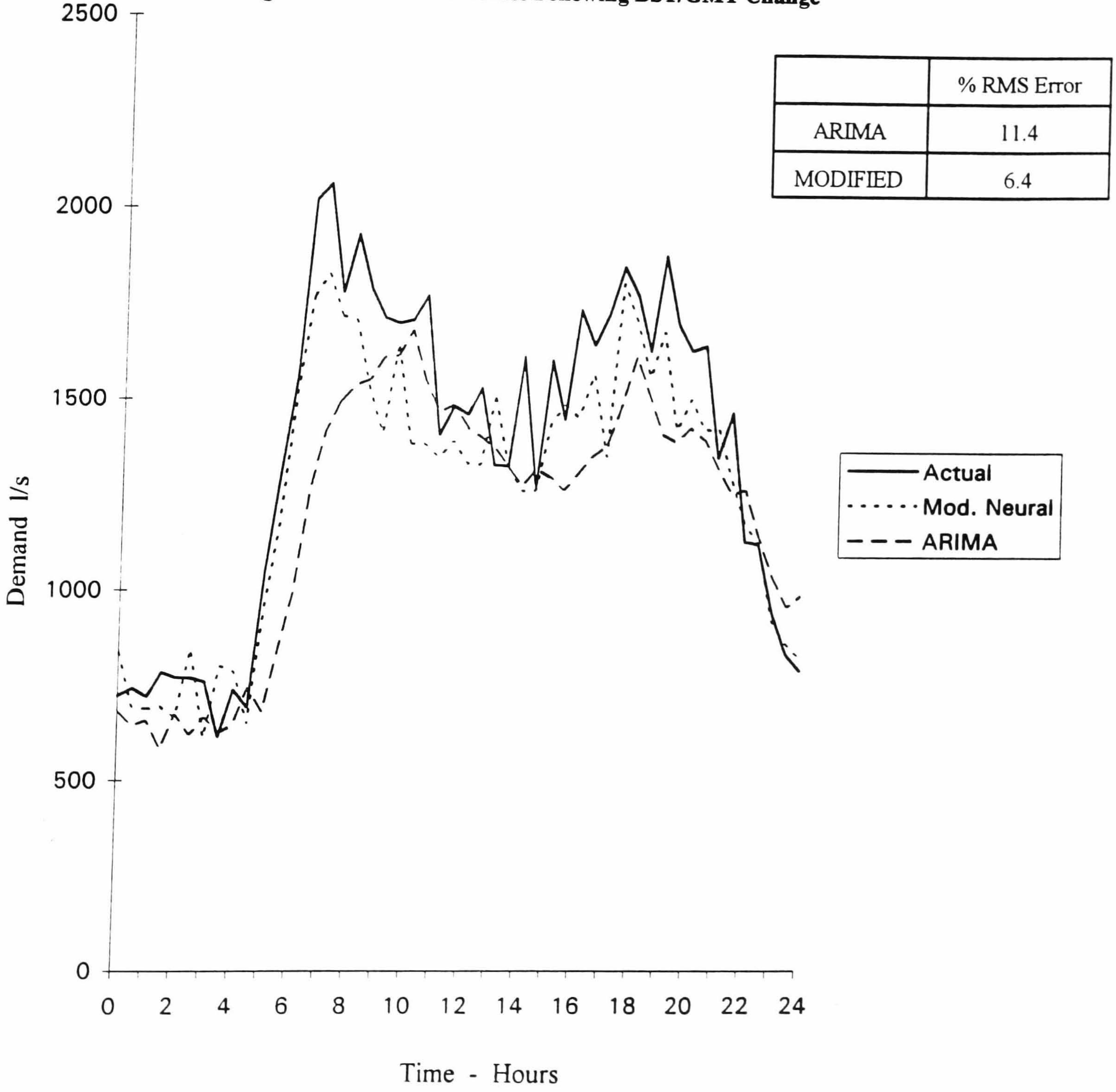


Figure 4.5.7 Demand Profiles Following BST/GMT Change



	% RMS Error
ARIMA	11.4
MODIFIED	6.4

4.5.2 Network Related Effects.

The only Network related effect that is present in the available consumption data is the opening of a major export to the Aylesbury area that occurs for three days in April of 1990 as a reservoir was being filled. This has the effect of shifting consumption upwards by an almost constant amount for the duration of the filling exercise. Figure 4.5.8 shows the actual, unaltered ARIMA and modified neural prediction profiles for one of the days during this period.

4.5.3 Weather Related Effects.

As will be shown in the following chapter, the neural network demand forecasting approach was developed to account for the majority of the effects of the meteorological conditions upon the consumption profile. However, there are a number of extreme or exceeding sudden effects caused by weather conditions that need the application of heuristic rules to fully account for their influence. Examples of such effects are the sudden halting of evening garden watering caused by heavy rain after a long period of dry weather, this effect is demonstrated in Figure 4.5.9, where the neural prediction profile based on data gathered during a long preceding period of almost totally dry weather does not totally account for the fall off in evening consumption. Another example of where heuristic rules are required to augment the neural network predictions is for the occurrence of the first hot weekend of the year. Consumptions are significantly increased on such weekends but because the neural prediction does not yet have any examples of water consumption on hot dry weekends its predictions are too low, consumption boosting rules are required in this instance. Figure 4.5.10 and 4.5.11 show this effect for the weekend of 24th and 25th April 1990 where the unaltered neural network profile is too low and requires modifying via the rule base.

Figure 4.5.8 Demand Profiles During Reservoir Filling

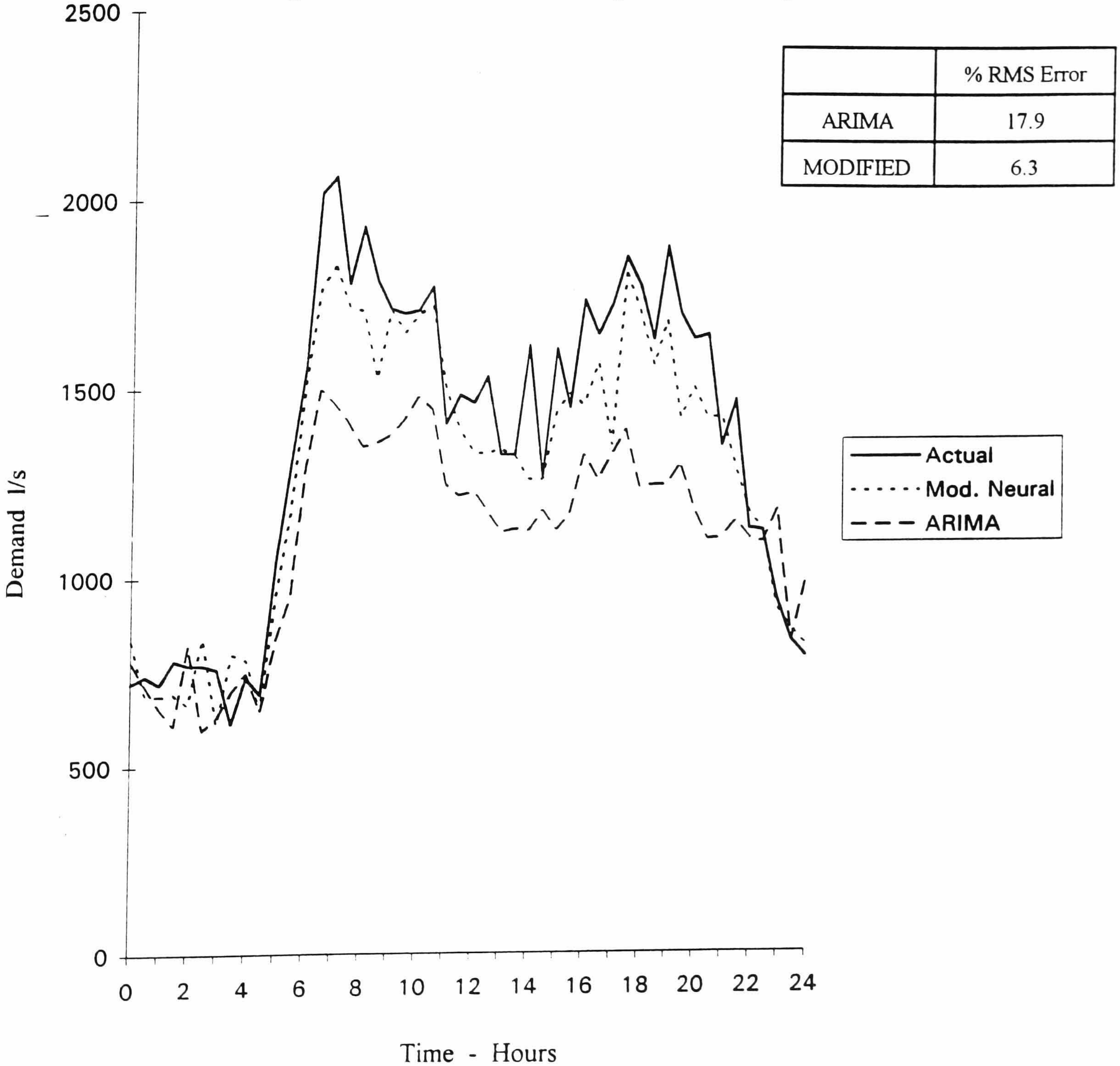
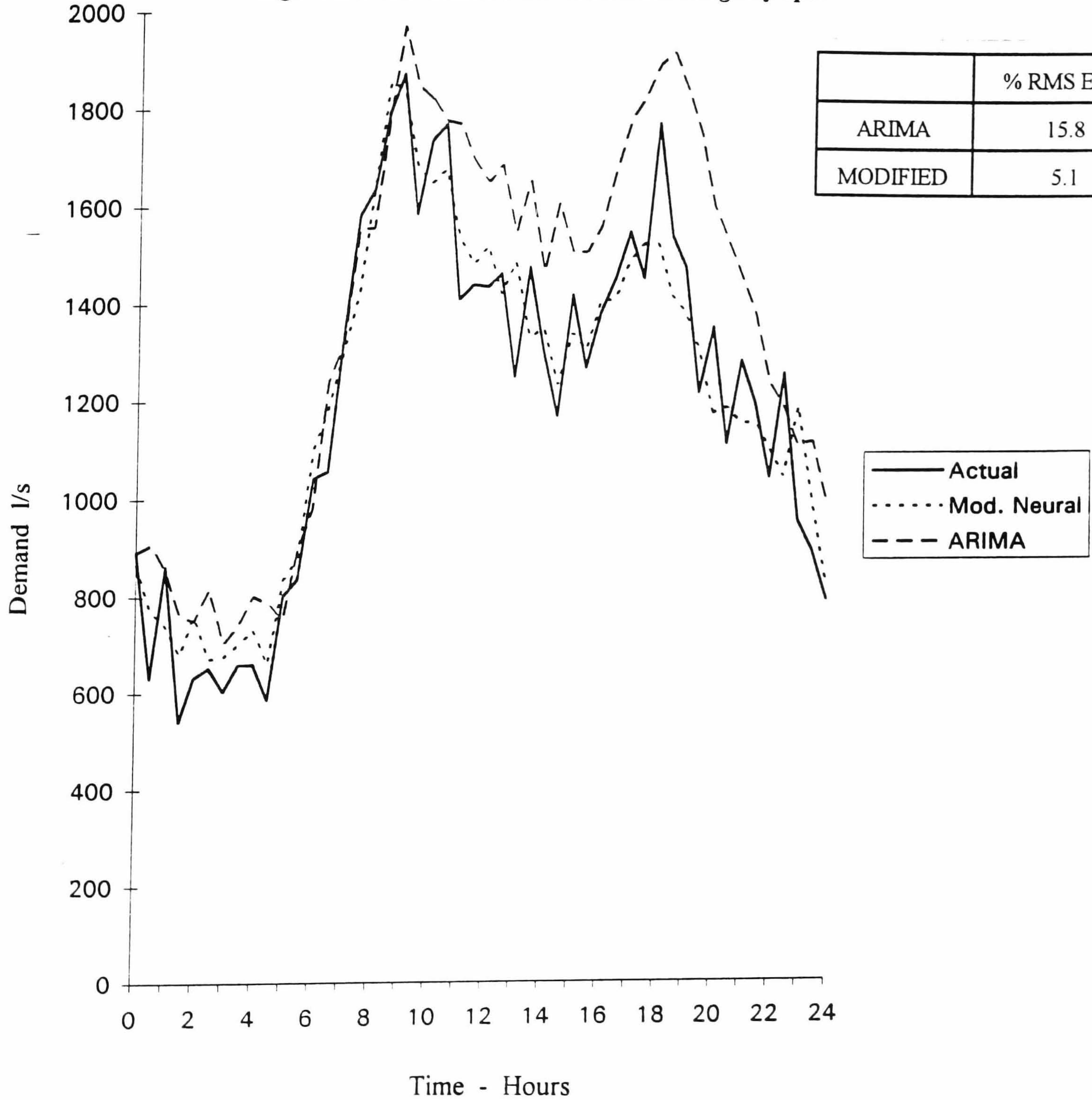
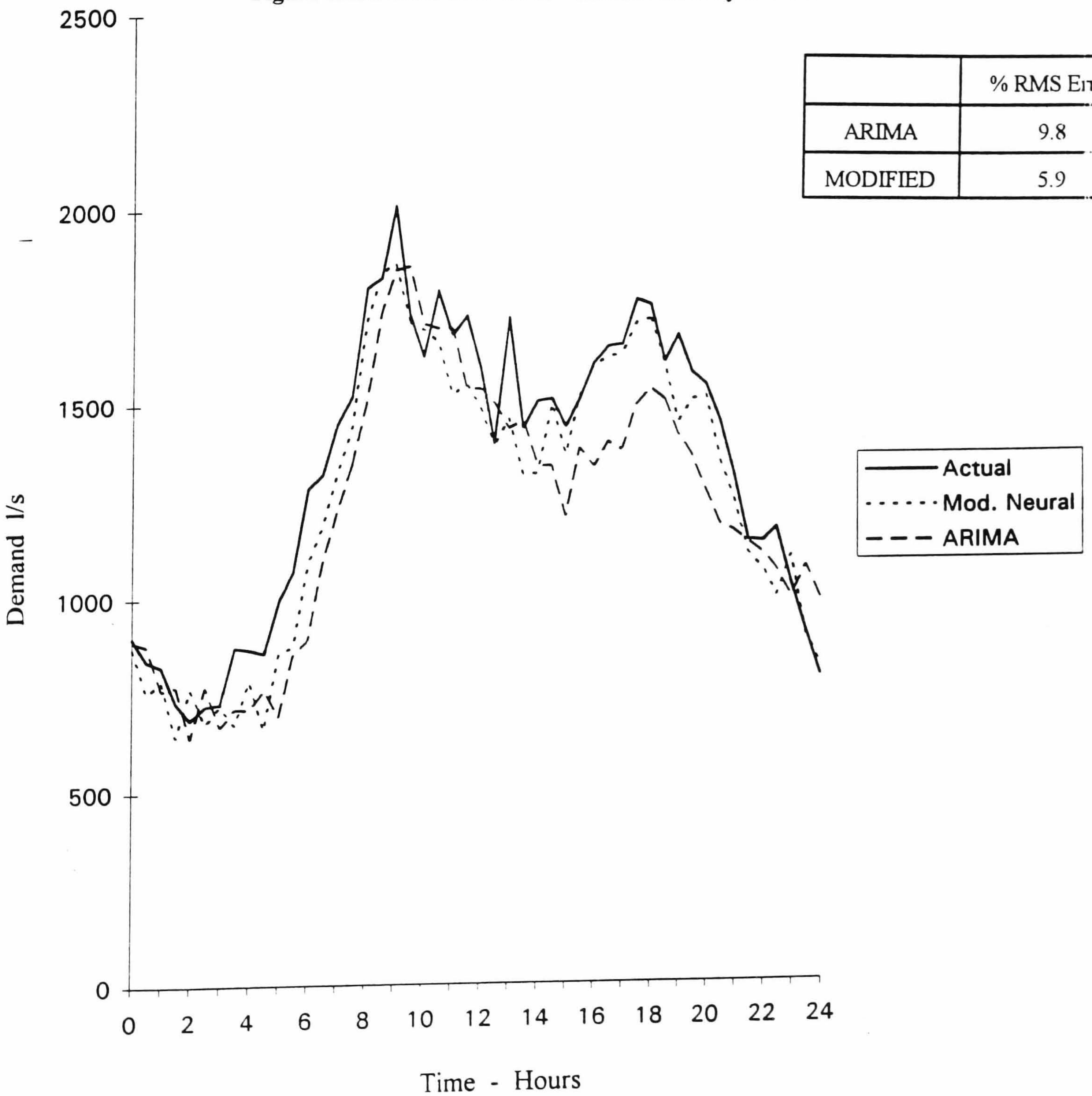


Figure 4.5.9 Demand Profiles - Rain After Long Dry Spell



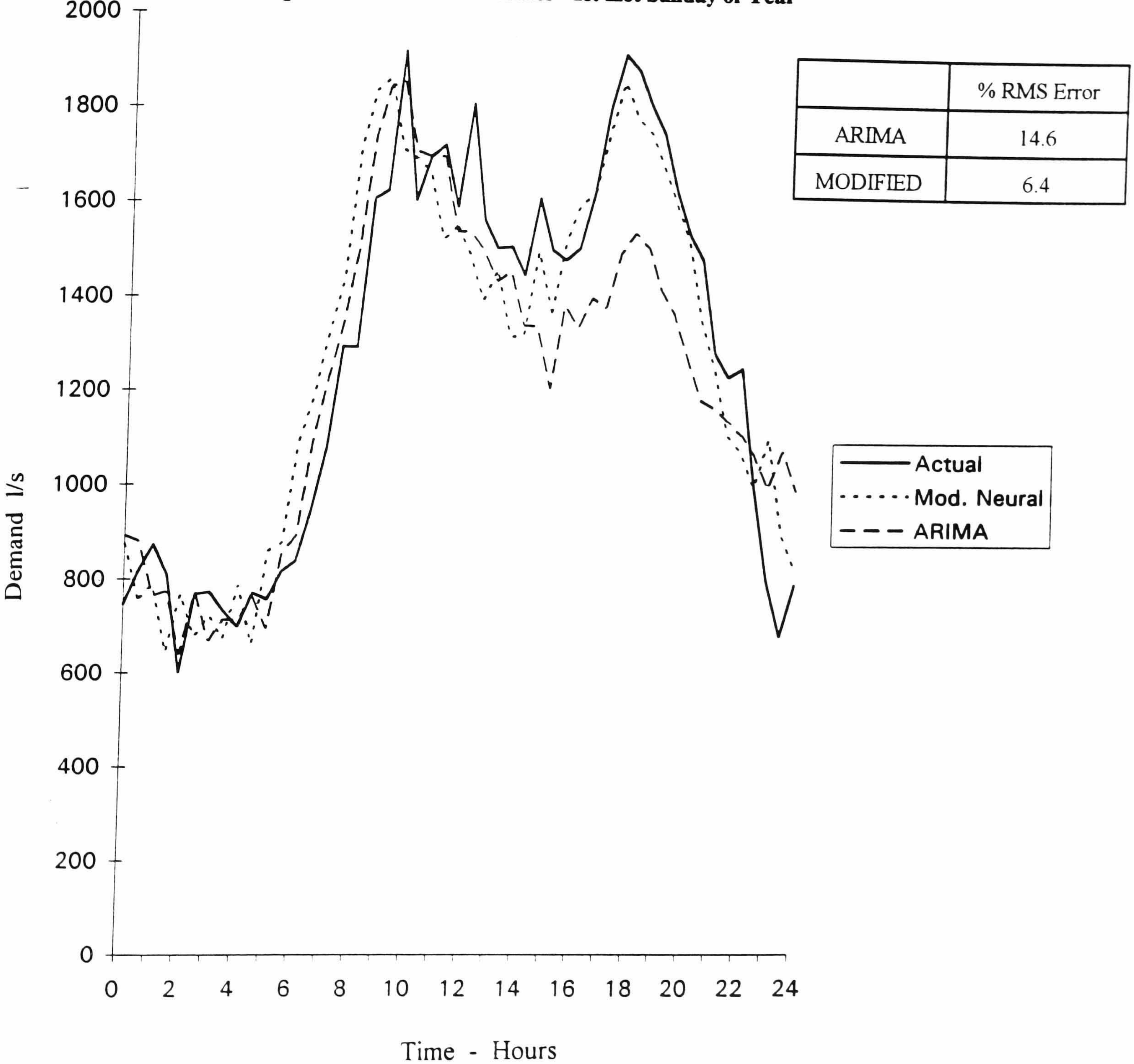
	% RMS Error
ARIMA	15.8
MODIFIED	5.1

Figure 4.5.10 Demand Profiles - 1st Hot Saturday of Year



	% RMS Error
ARIMA	9.8
MODIFIED	5.9

Figure 4.5.11 Demand Profiles - 1st Hot Sunday of Year



CHAPTER 5

A NEURAL NETWORK DEMAND FORECASTING APPLICATION

5.1 Introduction.

Neural networks can be described as directed graphs, they are composed of individual processing units termed neurons which are arranged in a layered form and are connected to other neurons in the same and/or different layers by weighted connections. Signals are applied to the input layer of a network and are propagated to the subsequent layers via the connections between layers. As a signal passes through the network it is altered by the weights associated with each of the connections through which it passes and by the transfer functions of each of the individual neurons to which it provides an input. Once the propagated signal arrives at the final layer of the network it is processed by the neurons in that layer and forms the output signal of the network.

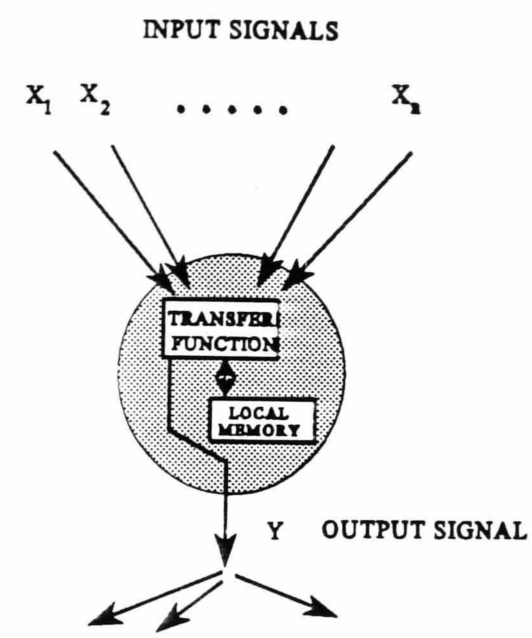
The interest generated in neural networks is centred on their ability to learn by producing a mapping between a given input signal and a desired output signal. Through the development of advanced learning algorithms [53,54,113,117,130] and network architectures, highly complex relationships have successfully been modelled by neural networks where conventional mathematical approaches have failed to provide adequate solutions. This is particularly the case where there exists a complex interplay between a number of influencing factors which interact to generate a particular end result, an example of such a relationship is the influence of weather conditions upon the level of water demand. Neural networks allow a 'black box' approach to be applied to such relationships so avoiding the need to explicitly define the exact interrelation between each of the influencing factors and their specific influence on the final result.

5.2 The Basic Elements of a Neural Network

There have been a number of attempts to link the study of neural networks directly with the thought processes in operation within the human brain [29,66]. However, these have been largely unsuccessful; in part this is due to the lack of a definite understanding of the exact learning processes of the brain and also to the fact that the networks that have been developed to date provide only a simplification of those thought processes that are known to occur. It is therefore not intended to cover this aspect of neural networks other than to highlight that the link exists between some of the theories that have been developed to explain the neurophysiological operation of the brain and many of the ideas that have been the basis for significant steps forward in neural network development.

5.2.1 Neurons

The fundamental building block for a neural network is the neuron or processing unit, a typical example of which is shown in figure 5.2.1. A processing unit receives signals from other processing units or from inputs to the network and generates an output signal which is passed to either other neurons or comprises an element of the output of the network as a whole. A processing unit can have any number of incoming connections and any number of outgoing connections.



- Figure 5.2.1 A Neuron.

Neurons are arranged within the network in layers with each layer consisting of one or more individual neurons. The connections between neurons can link neurons to each other within a layer, to other neurons in different layers or to external network inputs and outputs. The architecture of a typical two layer neural net is shown in figure 5.2.2.

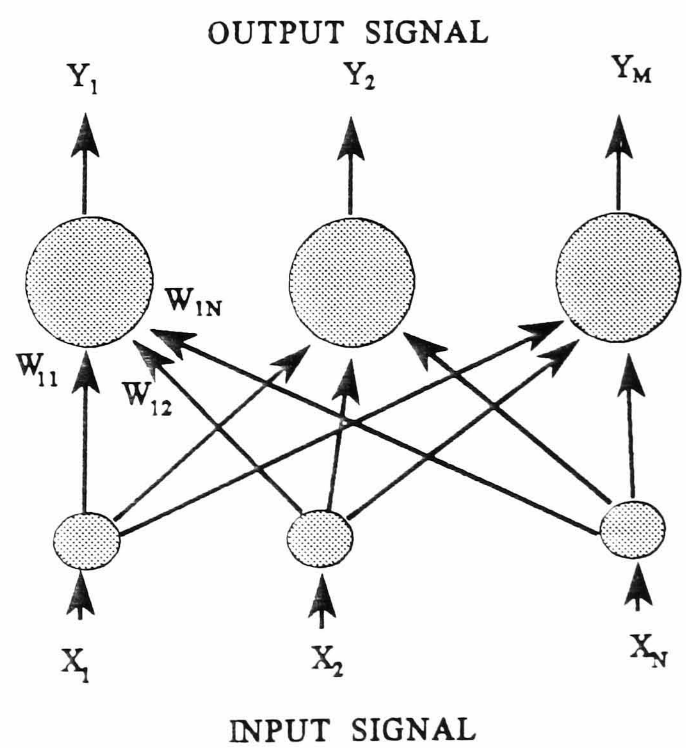


Figure 5.2.2 Structure of a Simple Two Layer Network.

Input signals to a particular neuron can be of any mathematical data type, however, the signals arriving at neurons in the same layer at the same time are conventionally all of the same data type. The output signal generated by a neuron is the same for all its outgoing connections.

Each neuron commonly possesses a transfer function and a local memory. The transfer function generates the neurons output signal from the values it receives as input signals and any values stored in the local memory (if present). This transfer function can be a simple summation of the input signal values, or it can be a more complex function such as a linear, sigmoidal, ramp or step function. Commonly it takes the form of a threshold value which must be exceeded by the incoming signals before the neuron will generate an output other than zero.

5.2.2 Connections Between Neurons

In most neural network structures the connections between processing units have weight values associated with them, these represent the strength of the particular connection between processing unit A and processing unit B. These weight values are commonly stored in a local memory array within the processing unit to which they provide input. It is these connection weight values, together with the processing unit transfer functions, that determine the nature of the output signal that is generated by a network in response to an applied input stimulus. As an input signal applied to the network propagates from one layer of the network to the next, the weight values determine the signal that is submitted to any particular processing unit and the transfer function of that unit determines the signal that is passed on to the next layer.

In addition to the weight values and transfer functions of a network, the network connection topology is also important in determining its operation. Not all networks have the completely connected architecture displayed by the example network in Figure 5.2.2. Complex linking of groups of neurons in separate layers and within the same layer can be implemented and a way of defining these architectures is via the use of fascicles and input classes [67]. Fascicles are used to describe the way in which particular groups of source neurons are linked to particular groups of target neurons by identifying for each target neuron all the source neurons that supply signals to it of a particular class. Input classes are required because target neurons can receive signals from more than one set of source neurons (more than one fascicle) and the transfer function may need to treat each set of incoming signals in a different manner, input classes allow the target neuron to identify to which fascicle an incoming signal belongs. Figure 5.2.3 shows the concept of fascicles and figure 5.2.4 shows signals of different input classes arriving at a single neuron. All connections belonging to the same input class must be of the same mathematical data type.

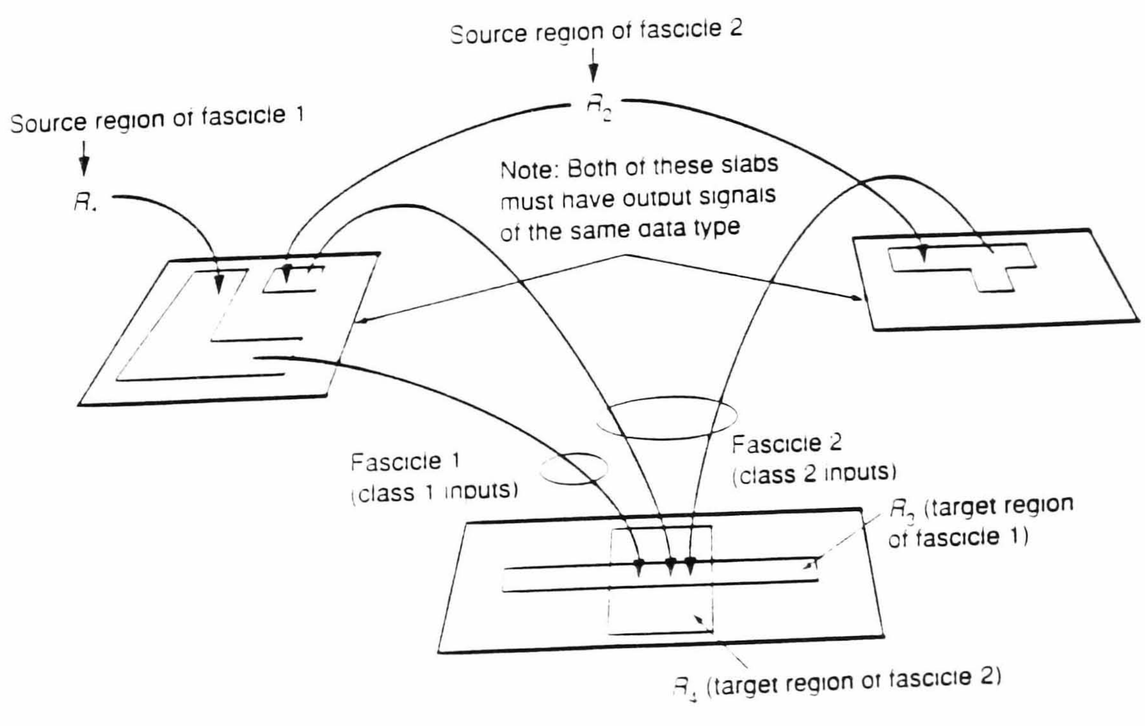


Figure 5.2.3 The Concept of Fascicles.

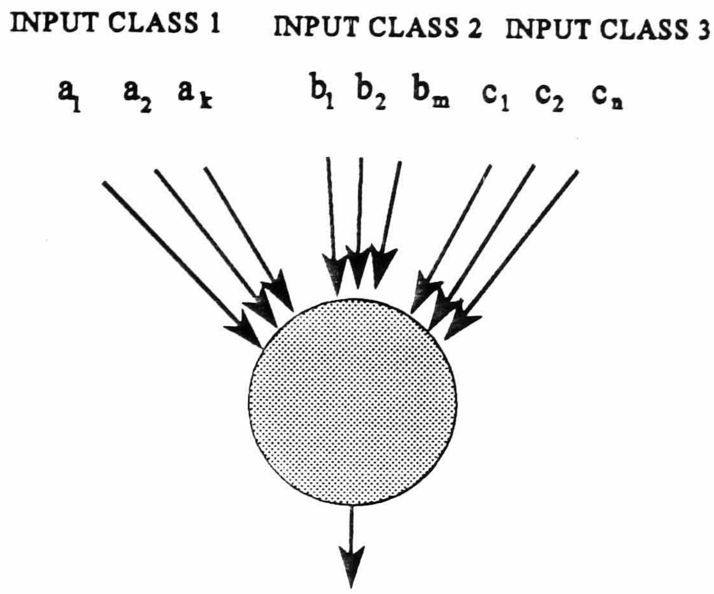


Figure 5.2.4 Signals of Different Input Classes Arriving at a Neuron.

Whilst fascicles and input classes allow highly complex network architectures to be defined, there are a number of simple network topologies which are very common, such as the fully connected network where each neuron in one layer is connected to each neuron in the adjacent layers, and the randomly connected network where the presence or absence of a connection between any two given neurons within the network is determined at random. The fully connected architecture is used in the application described later in this chapter.

5.2.3 Learning Algorithms

The process of training a network so that it performs a desired function involves the application of one of the many learning algorithms that have been developed for neural network applications. The general aim of the learning process is to observe the networks performance in response to particular input stimuli and use this information to modify and improve that performance. The predominant method for achieving this improvement in performance is by the adjustment of the weight values associated with the connections between individual processing units within the network. Other methods for altering the network operation include the addition and removal of connections and/or processing units [6].

In order to provide a basis for the discussion of the topic of network connection weight adjustment, the following section provides a mathematical description of the network weight matrix $\mathbf{W}$.

The network weight matrix $\mathbf{W}$ is formed by concatenating all of the weights of all of the individual processing units of the network being described. For a network with N units each possessing n weights, this can be written as:

$$\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N) \qquad (5.2.3.1)$$

Where the vectors $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N$ are the weight vectors of the individual processing units of the network. these are defined as:

$$\mathbf{w}_1 = \begin{bmatrix} w_{11} \\ w_{12} \\ \vdots \\ w_{1n} \end{bmatrix}$$

$$\mathbf{w}_2 = \begin{bmatrix} w_{21} \\ w_{22} \\ \vdots \\ w_{2n} \end{bmatrix}$$

$$\mathbf{w}_N = \begin{bmatrix} w_{N1} \\ w_{N2} \\ \vdots \\ w_{Nn} \end{bmatrix}$$

The set that contains all possible values of the network weight matrix $\mathbf{W}$ determines the set of all possible information processing configurations of a particular network. This implies that if the desired information processing capability is to be realised by a given network, then it will be found at some particular value of the matrix $\mathbf{W}$. The aim of a learning algorithm is to efficiently guide each network weight vector to the location in the solution space such that it yields the desired network performance.

Learning algorithms can usefully be divided into; performance learning laws that seek to minimise or maximise some particular global network cost or performance function (e.g. the mean squared error) and filter learning laws that do not attempt to optimise a specific cost function but instead have a goal that can be expressed in behavioural or mathematical terms. Many different learning algorithms have been developed for neural network applications and examples of learning algorithms that are of particular relevance to the work discussed in this thesis are described later in this Chapter and Chapter 6.

5.2.4 Training

The application of a learning algorithm in order to adjust the values of the network weight matrix is achieved by the implementation of a training regime. Such training regimes can be divided into three categories; supervised training, reinforcement training and self organisation.

Supervised training is applicable to situations where the neural network in question is to behave purely as an input/output system i.e. an input vector $\mathbf{x}$ is applied to the network and a resulting output vector $\mathbf{y}'$ is generated. The training process for such a network involves the submission to the network of multiple example input/output vector pairs $(\mathbf{x}_1,\mathbf{y}_1),(\mathbf{x}_2,\mathbf{y}_2),\ldots,(\mathbf{x}_L,\mathbf{y}_L)$ where $\mathbf{x}_k$ is an applied input vector, $\mathbf{y}_k$ is the desired output vector and L is the number of example vector pairs in the training set. The particular learning algorithm selected for the network carries out adjustments to the network weight values such that the differences between the ideal output vector examples $\mathbf{y}_k$ and the network generated output vectors $\mathbf{y'}_k$ are minimised.

Reinforcement training is similar to supervised training except that instead of each individual network output vector being compared to an ideal output vector, the network weights are adjusted according to a score that represents the performance of the network over a number of input/output passes. The 'score' is a cost function that represents the overall ability of the network to achieve a specific goal. Such a training methodology is applicable to problems where it is difficult or impossible to identify precicely what the value of an individual output vector should be as a result of a single input/output pass. An example of such an application is the broom balancing neural network developed by Barto, Sutton and Anderson [11] where it is not possible to state the precise ideal location of the broom at any one time, only that the overall goal to be achieved is that it should remain upright.

Self organising network training does not utilise example ideal outputs or graded score values to update the network weights, instead random weight changes are made and an assessment carried out of the change in network performance. If the overall performance has been improved then the random weight change is kept, whereas if the performance has been decreased then a probability distribution is applied in order to determine if the change to the weights is kept or discarded. In this way local minima are avoided. This process is similar to simulated annealing [103].

5.3 Learning Algorithm Examples and Applications

5.3.1 Hebbian Learning and the Linear Associator Network

Hebbian learning is based in the work of Donald Hebb, who in 1949 proposed a theory [66] on the mechanism by which the learning process takes place in the brain at cellular level. The mechanism proposed was that when a neuron within the brain is repeatedly involved in contributing to the firing of another adjacent neuron, then a change takes place in the connecting synapse between the two neurons in question such that the efficiency of the firing mechanism is improved. Although the continuing research into the functioning of the brain has since shown this theory to be a gross simplification of the actual learning process, it is thought to be correct in general terms. In order to provide a basis for illustrating Hebbian learning, an example of a network architecture to which Hebbian learning can be applied is introduced below.

The linear associator network which is shown in figure 5.3.1 is a two layer feed forward network i.e. it consists of an input layer that distributes the applied input vector to the processing units of the output layer and the propagation of signals through the network takes place in a forward only direction. The linear associator network was introduced by Anderson [4] in 1972 using Hebbian learning and has since had many refinements and variations [89].

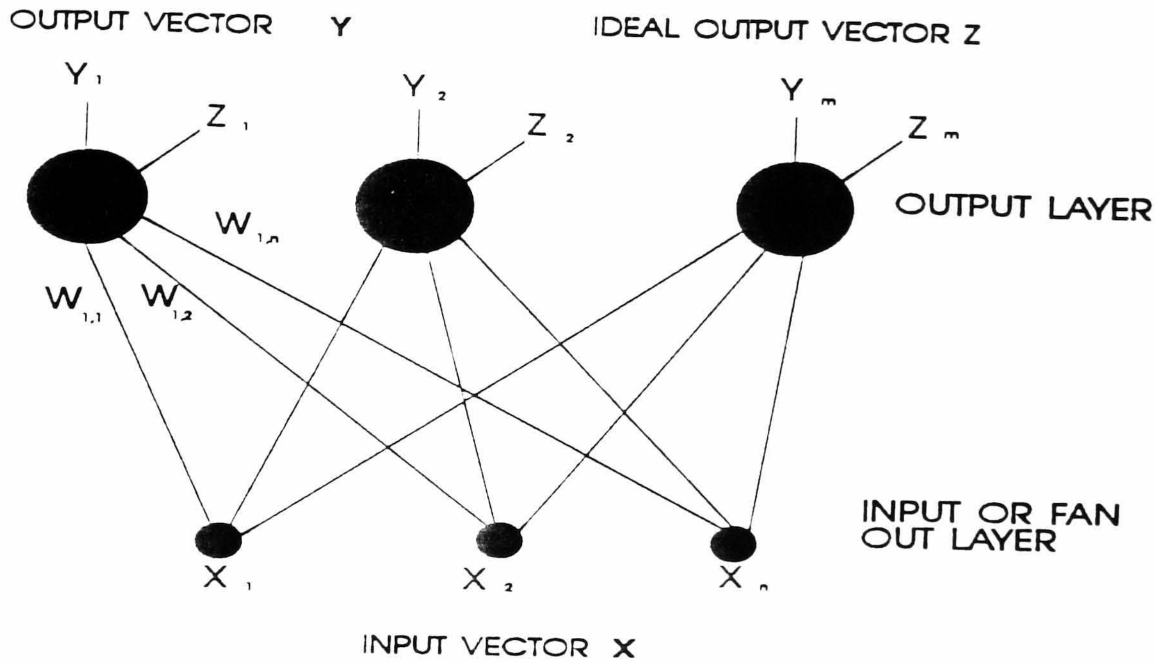


Figure 5.3.1. Linear Associator Network

The above diagram shows that the input layer of the network is composed of n input or 'fan out' units, these carry out no internal processing function and merely distribute the signals applied to them. The number of input units corresponds to the number of components of the network input vector $\mathbf{x}$. The output layer consists of m processing units that correspond to the number of components of the network output vector $\mathbf{y'}$. The network is fully connected. Associated with each connection to the processing units in the output layer is a weighting value $w_{11}, w_{12}, \ldots, w_{mn}$. A signal arriving as an input to a processing unit in the network output layer is multiplied by its corresponding weight value and summed with all the other weighted inputs to that processing unit to derive the unit's output signal. The output vector $\mathbf{y'}$ is composed of the output signals from each of the processing units in the network output layer. This output vector is derived from an input vector $\mathbf{x}$ by the application of the network weights:

$$\mathbf{y}' = \mathbf{W}\mathbf{x}$$

Where $\mathbf{W}$ is the network weight matrix described in section 5.2 and is of size $m \times n$. The goal of training the network is to adjust the weights in matrix $\mathbf{W}$ such that the network 'learns' L pairs of input/output mappings $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots, (\mathbf{x}_L, \mathbf{y}_L)$. Once trained the network should generate the desired output in response to a given input i.e. if input vector $\mathbf{x}_k$ is applied to the network, the network generates output vector $\mathbf{y'}_k$, which is equal to the desired output vector $\mathbf{y}_k$.

Hebbian learning can be used to achieve this vector mapping (providing certain restricting conditions outlined later in this section are met). The Hebbian learning formula can be stated as:

$$w_{ij}^{new} = w_{ij}^{old} + y_i x_j \qquad (5.3.1.1)$$

Where w_{ij} is an individual weight value and x_j and y_i are the j^{th} and i^{th} components of the training vectors $\mathbf{x}_k$ and $\mathbf{y}_k$ respectively. This is expressed in matrix/vector form as:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mathbf{y}_k \mathbf{x}_k^{\mathrm{T}} \qquad (5.3.1.2)$$

The initial values of the weights in matrix $\mathbf{W}$ are set to zero at the start of the training process. Therefore, during a training process consisting of L example input/output vector pairs, the weight matrix changes from its initial zero state to its final trained state simply by summing all the incremental weight changes caused by the submission of the L training examples. The end state of the matrix $\mathbf{W}$ is given by:

$$\mathbf{W} = \mathbf{y}_1 \mathbf{x}_1^{\mathrm{T}} + \mathbf{y}_2 \mathbf{x}_2^{\mathrm{T}} + \ldots + \mathbf{y}_L \mathbf{x}_L^{\mathrm{T}} \qquad (5.3.1.3)$$

However as reference [67] has shown, Hebbian learning can only guarantee the correct vector mapping performance from a network if all input vectors $\mathbf{x}_k$ are normalised to unit length and are mutually orthonormal. This has the effect that the network cannot successfully learn more than n vector mappings where n is the number of network input units.

5.3.2 Widrow Learning and the Linear Associator Network

In order to overcome the restriction Hebbian learning places upon the number of input/output pairs that can be learnt by the linear associator, a learning algorithm known as Widrow or Least Mean Square (LMS) learning can be applied [158].

The LMS algorithm adjusts the networks weight matrix $\mathbf{W}$ so as to minimise the least mean square error between the network generated output vector $\mathbf{y}'$ and the desired output $\mathbf{y}$. The network training procedure associated with the LMS algorithm and implemented in a linear associator network with n input units and m output units is given below:

1) Assign random initial values in the range [-1,+1] to the units of the $n \times m$ network weight matrix $\mathbf{W}$.

2) For each example input/output vector pair $(\mathbf{x}_k, \mathbf{y}_k)$ to be submitted to the network carry out the following:

i) Apply the input vector $\mathbf{x}_k$ components $(x_1, x_2 \ldots x_n)$ to the n network input units.

ii) Propagate the applied input signal via the weighted connections to the output layer processing units and for each unit calculate output signal y'_j for $j = 1 \ldots m$

$$y'_j = \sum_{i=1}^{n} w_{ij} x_i$$

iii) Compute the error between the network generated output values y'_j and the desired network output values y_j for each output unit.

$$e_j = y_j - y'_j \qquad \text{for } j = 1....m$$

iv) Calculate the adjustment for each connection weight using the equation

$$\Delta w_{ij} = \alpha x_i e_j \qquad \text{for } i = 1...n \text{ and } j = 1...m$$

Where Δw_{ij} is the change in the value of unit ij of weight matrix $\mathbf{W}$ and α is a small positive constant termed the learning rate.

3) Repeat step 2) until the error correction values e_j for all output units $j = 1...m$ and all training vector pairs $k = 1...L$ fall below a specified threshold value.

The weight updating procedure can be expressed in terms of the network weight matrix as:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \alpha(\mathbf{y}_k - \mathbf{W}_k \mathbf{x}_k)\mathbf{x}_k^{\mathrm{T}} \qquad (5.3.2.1)$$

The proof that the LMS learning algorithm will converge to a global minimum is given through analysis of the mean squared error, this is shown below .

5.3.2.1 Proof of Convergence

Given a network generated output vector $\mathbf{y}'$ and an example vector pair $(\mathbf{x}_k, \mathbf{y}_k)$ with units $x_1, x_2, ... x_n$ and $y_1, y_2, ... y_m$, the error e between the desired and actual output for the j^{th} processing unit is found using the equation:

$$e_j = y_j - y'_j$$

$$e_j = y_j - \sum_{i=1}^{n} w_{ij} x_i$$

This is written in vector form as:

$$e_j = y_j - \mathbf{w}_j^{\mathrm{T}} \mathbf{x}_k \tag{5.3.2.2}$$

Where $\mathbf{w}_j$ is the vector of weights associated with the connections to processing unit j. The squared error is therefore:

$$e_j^2 = (y_j)^2 - 2y_j \mathbf{x}_k \mathbf{w}_j^{\mathrm{T}} + [\mathbf{x}_k \mathbf{w}_j^{\mathrm{T}}]^2$$

$$e_j^2 = (y_j)^2 - 2y_j \mathbf{x}_k \mathbf{w}_j^{\mathrm{T}} + \mathbf{w}_j \mathbf{x}_k^{\mathrm{T}} \mathbf{x}_k \mathbf{w}_j^{\mathrm{T}} \tag{5.3.2.3}$$

Assuming a stationary input environment, the input and output variables are replaced with their means, yielding the mean squared error equation:

$$E\{e_j^2\} = E\{y_j^{\ 2}\} - 2\mathbf{w}_j^{\mathrm{T}} E\{y_j \mathbf{x}_k\} + \mathbf{w}_j^{\mathrm{T}} E\{\mathbf{x}_k \mathbf{x}_k^{\mathrm{T}}\} \mathbf{w}_j$$

(5.3.2.4)

Where E represents the expectation operator. The above equation is simplified by making the substitutions:

$$\mathbf{P} = E\{y_j \mathbf{x}_k\} \tag{5.3.2.5}$$

Where P is the input-correlation vector of the input values and the desired network response and:

$$\mathbf{R} = E\{\mathbf{x}_k \mathbf{x}_k^{\mathrm{T}}\} \tag{5.3.2.6}$$

Where $\mathbf{R}$ is the autocorrelation matrix of the input layer processing unit values. Substituting in equation (5.3.2.3) gives:

$$E\{e_j^2\} = E\{y_j^{\ 2}\} - 2\mathbf{w}_j^{\mathrm{T}}\mathbf{P} + \mathbf{w}_j^{\mathrm{T}}\mathbf{R}\mathbf{w}_j$$

(5.3.2.7)

Minimising the mean squared error is performed by calculating the change in the estimated mean squared error with respect to the change in the weight vector $\mathbf{w}_j$ calculated using the equation:

$$\frac{\partial}{\partial \mathbf{w}_j} E\{(y_j)^2\} = 0 - 2\mathbf{P} + 2\mathbf{R}\mathbf{w}_j \qquad (5.3.2.8)$$

Setting the result equal to 0 and solving for $\mathbf{w}_j$ results in:

$$\mathbf{P} = \mathbf{w}_j\mathbf{R} \qquad (5.3.2.9)$$

$$\mathbf{w}_j = \mathbf{P}\mathbf{R}^{-1} \qquad (5.3.2.10)$$

This is the matrix form of the Wiener-Hopf equation and proves that the LMS algorithm will find the optimal weight vector, in a least squares sense, providing the inverse of $\mathbf{R}$ exists.

5.4 An Application of the Linear Associator Network in Demand Forecasting

5.4.1 Introduction

As has been outlined in previous chapters, the problems encountered in generating accurate short term predictions of water demand are chiefly related to the effects of non cyclic events and influences. These were classified in Chapter 3 into three categories: calendar related effects that influence the level of demand on specific dates or at specific times of the year, network related events that are caused by some change that has occurred in the water network itself and weather related effects caused by meteorological changes. Chapter 4 described a methodology which, through the use of a rule based approach, could successfully account for many of the calendar and network related effects.

The problem of accounting for the weather related influences upon water consumption patterns is a highly complex one and does not readily lend itself to solution by the application of traditional mathematical modelling techniques. This is due in part to the interaction of diverse meteorological variables with a combined influence that results in a particular level of water demand. The relative influence of each individual variable changes with both the time of the year and geographical location. Hence, no consistant mathematical relationship between these variables and observed water consumption has sucessfully been derived. In addition, the perception of the meteorological situation by consumers and their consequent water usage, is highly subjective and is therefore very difficult to account for directly. Finally, there is the problem of the availability of accurate forecasts of the meteorological variables themselves. Predicted values of temperature, rainfall totals etc. may be in error and consequently have an adverse effect upon the accuracy of a demand prediction based upon them. The neural network demand forecasting program described below has been developed with the aim of providing a solution to the problems associated with accounting for weather influences.

A neural network approach was selected as being appropriate for investigation into the possibilities of generating demand predictions that could account for weather related influences. The principal reason for this is the ability of neural networks to model complex interrelationships between numerous factors which can not be easily defined by traditional mathematical techniques. Initially, investigations were carried out to determine the meteorological variables that have the greatest influence in determining the levels of water consumption. These investigations took the form of statistical examination of the available Thames Water data and Heathrow meteorological data, to identify the correlation between consumption and individual weather variables. In addition, a review of previous related work was undertaken to extract any useful information on the relative importance of various weather variables [24,44,46,120,142]. Figures 5.4.1 and 5.4.2 show plots of weather variables and their correlation with demand. From these statistical relationships it was shown that the principle meteorological factors involved in influencing water demand for the particular data examined, are air temperature, the rainfall pattern (expressed in No. of antecedent dry days) and the number of hours of sunshine each day. The reasons for the above meteorological factors having an influence upon water consumption are linked to consumer activities such as garden watering, irrigation and washing. Given the nature of these activities, such influences will be most apparent during the spring and summer months and much less important during the winter, hence a successful prediction system must be able to track these changes in influence with time.

Figure 5.4.1 Total Daily Demand With Temperature and Sun Hours

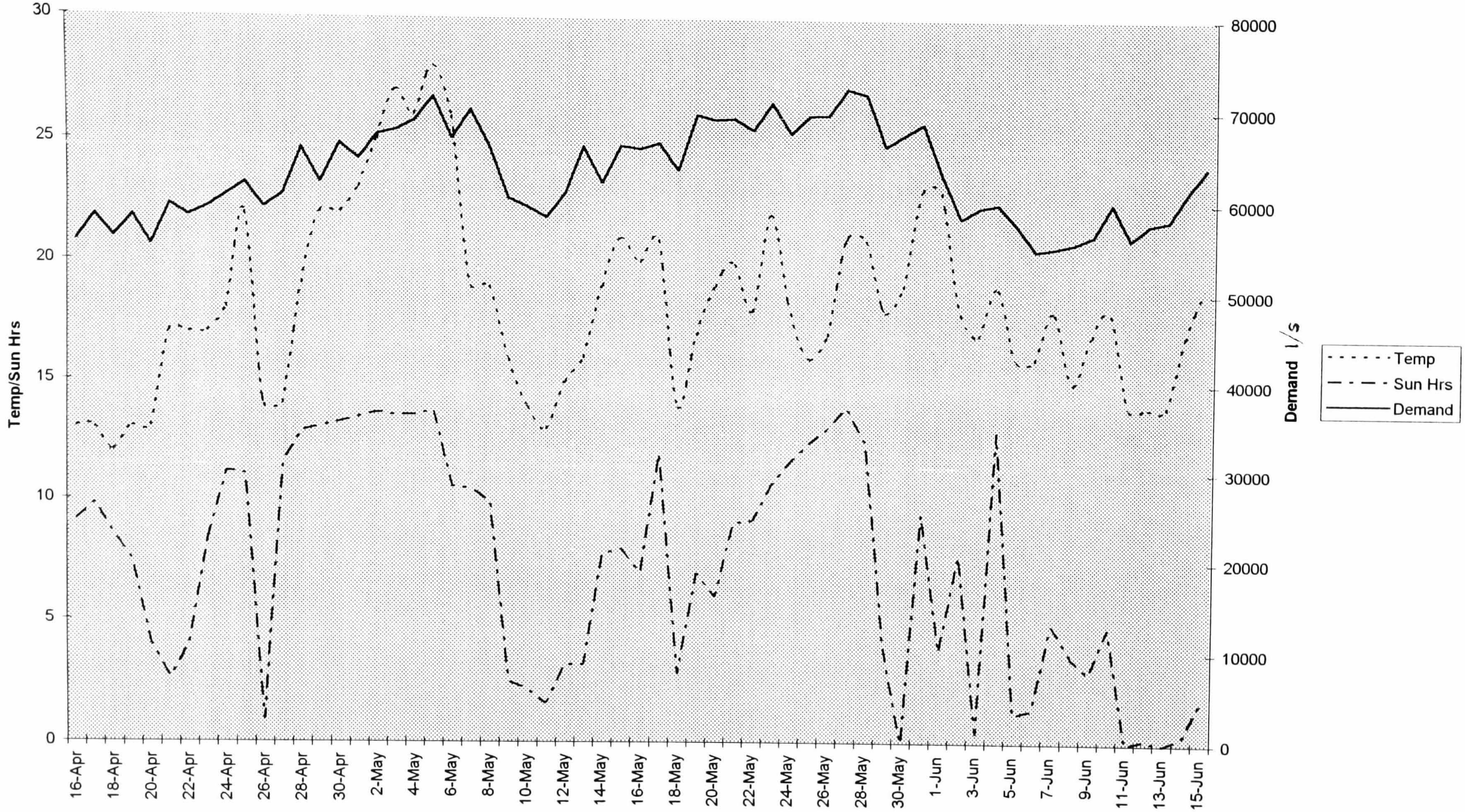
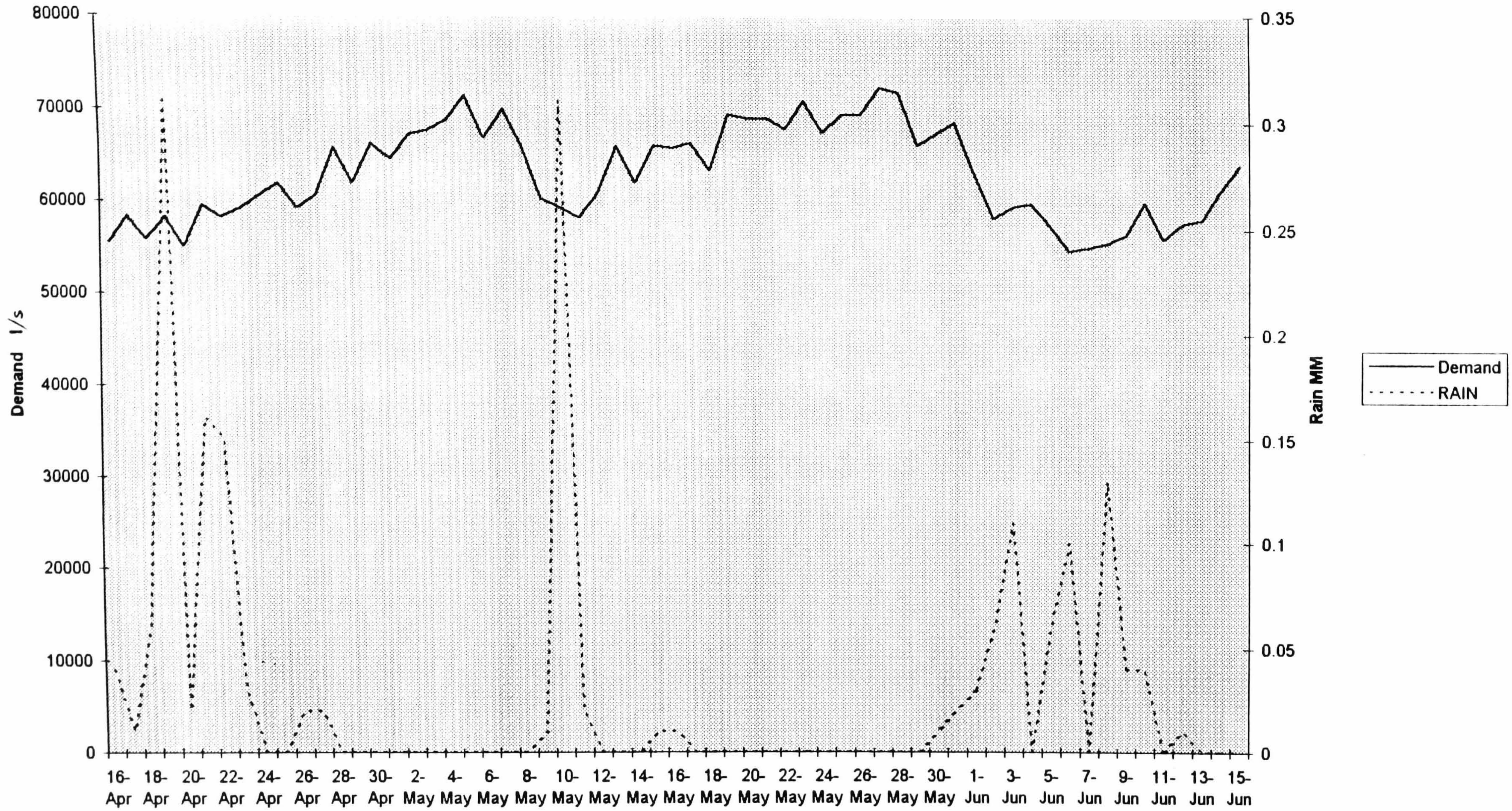


Figure 5.4.2 Total Demand Variation With Rainfall



As has been stated, there is a potential source of error in using predictions of individual meteorological variables as a basis for generating demand forecasts. In order to minimise the possibility of weather forecast errors having a detrimental effect upon the demand prediction accuracy, a day type classification system was developed. This provided a means of incorporating meteorological data without requiring precise predictions of individual meteorological variables.

The day type classifier established four basic day types ranging from type 1 which represents normal meteorological conditions for the time of year of the prediction day, to type 4 which represents extremely hot and dry conditions, with day type 2 and day type 3 as graduations between these limits. This fairly coarse division of the range of possible meteorological conditions has the advantage of avoiding the need for precise predicted values of temperature, rainfall totals and sunshine hours. A general weather forecast for the prediction day (for the appropriate geographical area) is sufficient to determine which of the four day type categories is most applicable. Potential errors in the predicted values of temperature, sunshine hours etc. are likely to be absorbed within the broad banding of the 4 day types and hence not have a significant adverse effect upon the final demand prediction. The task for the operator of the system is also made more simple, instead of looking up and entering exact meteorological data, only the single decision as to the appropriate day type needs to be made and entered.

5.4.2 Past Data Requirements

As described in Chapter 4, the water consumption data used in the testing and assessment of the neural network predictor was from the Slough and High Wycombe areas of Thames Water's Chilterns Division. This consumption data covered the period from March to July 1990 and consisted of half hourly consumption totals. Meteorological data covering the same time period was obtained from readings at Heathrow Airport which is geographically close to the Chilterns area. The weather data values extracted were Maximum Temperature in degrees Celsius, Total Daily Rainfall in mm and Total Sunshine Hours. This data was stored in the format shown in Table 5.4.3.

Date	Maximum Temperature	Total Daily Sunshine Hours	Total Daily Rainfall
WED 16 MAR 1990	12	4.5	0.02

Consumption Values Every Half Hour in l/s

890	990	876	765	753	802	874	832	844	917	970	1000
1134	1261	1456	1563	1487	1588	1690	1720	1534	1522	1455	
1487	1377	1345	1534	1465	1569	1253	1354	1444	1389	1355	
1542	1327	1645	1534	1335	1218	1125	1088	1025	988	885	

THU 17 MAR 1990	14	4.0	0.01

860	988	836	788	753	833	856	878	847	927	976	1023
1123	1251	1456	1563	1487	1570	1688	1700	1584	1572	1435	
1458	1377	1345	1534	1465	1569	1253	1354	1444	1389	1433	
1544	1327	1645	1534	1335	1218	1125	1088	1005	913	876	

Table 5.4.3. Example of past data used to generate demand predictions, demand values are in litres per second.

5.4.3 Classification into Day Types

In order to create the required input/output example vector pairs with which to train the neural network, a program was implemented in FORTRAN that automatically carried out the classification of the past data into the correct day type categories. For each day in the past data file the date, the 24 hours of consumption data and the corresponding weather variables are extracted. The date identifies the extracted day as a weekday, a Saturday or a Sunday, because of the significant differences in the daily demand profiles exhibited by weekdays, Saturdays and Sundays, it is necessary to create separate network training data sets for these three categories of day. Hence, an extracted 24 hour consumption profile is only passed on to be used as training data if it matches the category of the prediction day i.e. if the prediction day is a Saturday, only Saturday profiles are compiled into a training set, the weekdays and Sundays are read from the data file but not used for training.

For each day of past data extracted from the data file, a rolling record is kept of the number of consecutive dry days and hot days (days where the maximum temperature exceeds a defined threshold value). For each day that is to be used as network training data ,the dry day and hot day totals and the relevant meteorological variables are passed to the day type classification subroutine. This subroutine applies a number of rules to the meteorological data which determine the day type that will be assigned to the particular day being processed. Examples of the rules are shown below.

```
IF (TEMP .GT. 20) AND (HOT_DAYS .GE. 4) AND (DRY_DAYS .GT. 3) THEN
        DAY_TYPE = 3

ENDIF
```

Once the classification has been made for a particular day then the consumption data is entered into one of four data holding arrays depending on the day type assigned to it. The program then returns to the past data file and reads in the next 24 hours of data and the above process is repeated for each day until the prediction day is reached.

Four arrays are used to store the consumption data, one associated with each day type. The dimensions of the arrays used are 7 x 48 (i.e. they store 7 days data, each day consisting of 48 data points) and they are designed to operate as stacks. As each day of data is read and classified, the consumption data is placed on the top of the appropriate stack and the existing data within that stack is moved down. If the stack is full then the oldest 24 hours of data at the base of the stack is discarded. This ensures that only the most recent examples of a particular day type are used in the training process.

A count is kept of the number of examples of each day type that have been read in from the past data. In the event of there being no examples of a particular day type in the available past data, then the consumption data from the day type adjacent to the missing day type is used. For example if no day type 4 examples exist in the past data set used, the day type 3 consumption values are copied to the day type 4 array.

5.4.4 Training the Network

Each of the four arrays of classified past consumption data are passed in turn to the network training subroutine. This subroutine contains a FORTRAN implementation of the linear associator network, this network is illustrated in Figure 5.4.4. It consists of 4 input units which are fully connected to 48 output processing units. The training process is achieved by repeatedly submitting example input/output vector pairs $(\mathbf{x}_k, \mathbf{y}_k)$ to the network and adjusting the network weights associated with the connections between the input and output layers using equation (5.3.2.1). In this application the input vector $\mathbf{x}_k$ is a Boolean vector of length 4, the 4 components of this vector corresponding to the 4 possible day types. The input vector component relating to the day type of the current training example is set to 1 and the remaining components are set to 0.

The output vector consists of a 48 element vector which represents the 48 half hourly data points in a 24 hour demand forecast.

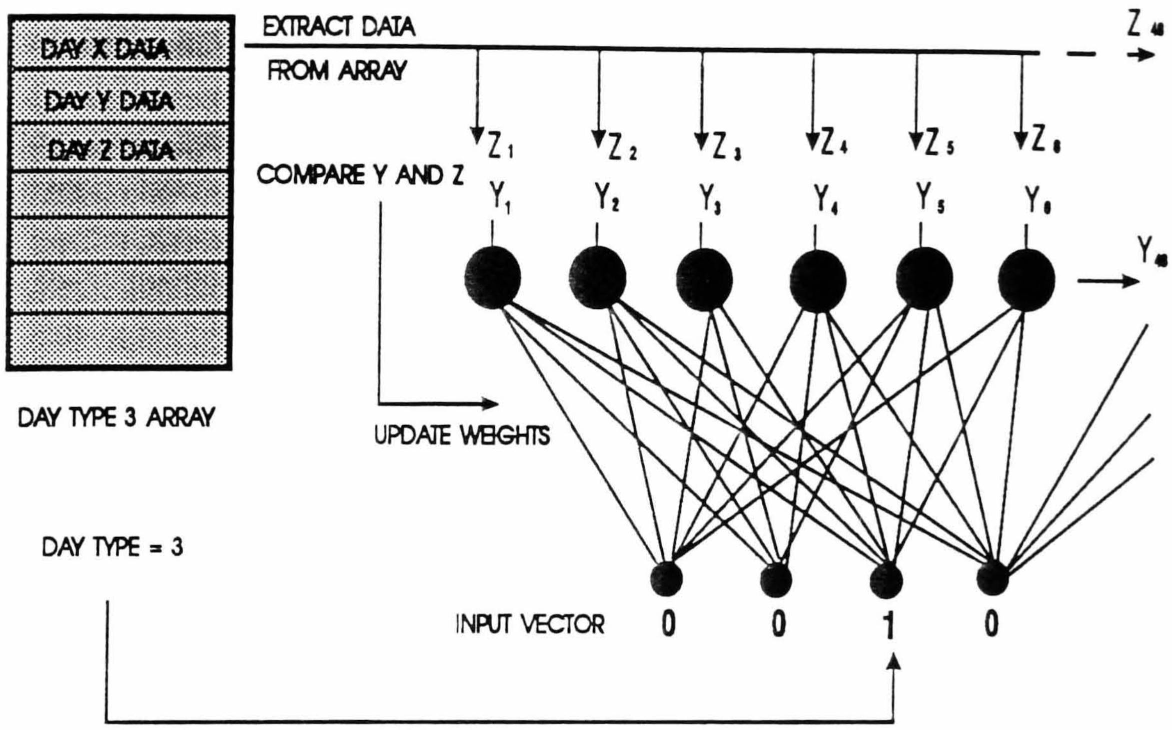


Figure 5.4.4 Training of the Linear Associator.

The weights associated with the connections between the input and output layers of the network are stored in a 4×48 matrix. These weights are set initially to random values in the range $0 \rightarrow 1$.

Training for an individual pair of input/output vectors $(\mathbf{x}_k, \mathbf{y}_k)$ consists of the following steps:

1) Extract the training pair from the array of example data passed to the training subroutine.

2) Apply the input vector $\mathbf{x}_k$ to the network input units.

3) Propagate the signal from the input units to the output processing units via the weighted connections. For each processing unit, apply the appropriate weights to the incoming signals and sum the results to derive the output signal of the unit.

4) The output signal of each output unit is compared to the corresponding unit of the example output vector $\mathbf{y}_k$.

5) Update the network weights by applying equation (5.3.2.1)

The number of times a particular pair of training vectors $(\mathbf{x}_k, \mathbf{y}_k)$ is sent round the above loop is dependent on how recent the data is that comprises the vector pair i.e. how near the top of the data holding stacks described earlier. The most recent data (that near the top of the stack) is submitted for network training up to 7 more times than the data extracted from further down the stack. This ensures a bias in the final configuration of the network weights towards data that is chronologically closest to the prediction day. The presence of this bias was found to have a significant effect upon the accuracy of the resulting demand predictions and this is illustrated in figures 5.4.5 and 5.4.6.

A training run for a single day type is complete once all pairs of example data present in the relevant stack have been submitted to the network a sufficient number of times, such that the differences between successive values of the network weights all fall below a threshold value i.e. the network has stabilised at a set of weight values that yield the correct output results for each training vector pair.

Training for the next day type then commences and the steps outlined above are repeated. Once training for all day types has been completed the network weight matrix contains the values it requires to carry out the mapping between a selected day type and the corresponding 24 hour demand prediction. The day type that has been entered by the operator as being the day type of the prediction day is applied to the

Figure 5.4.5 Effect of Multiple Submissions of Most Recent Data

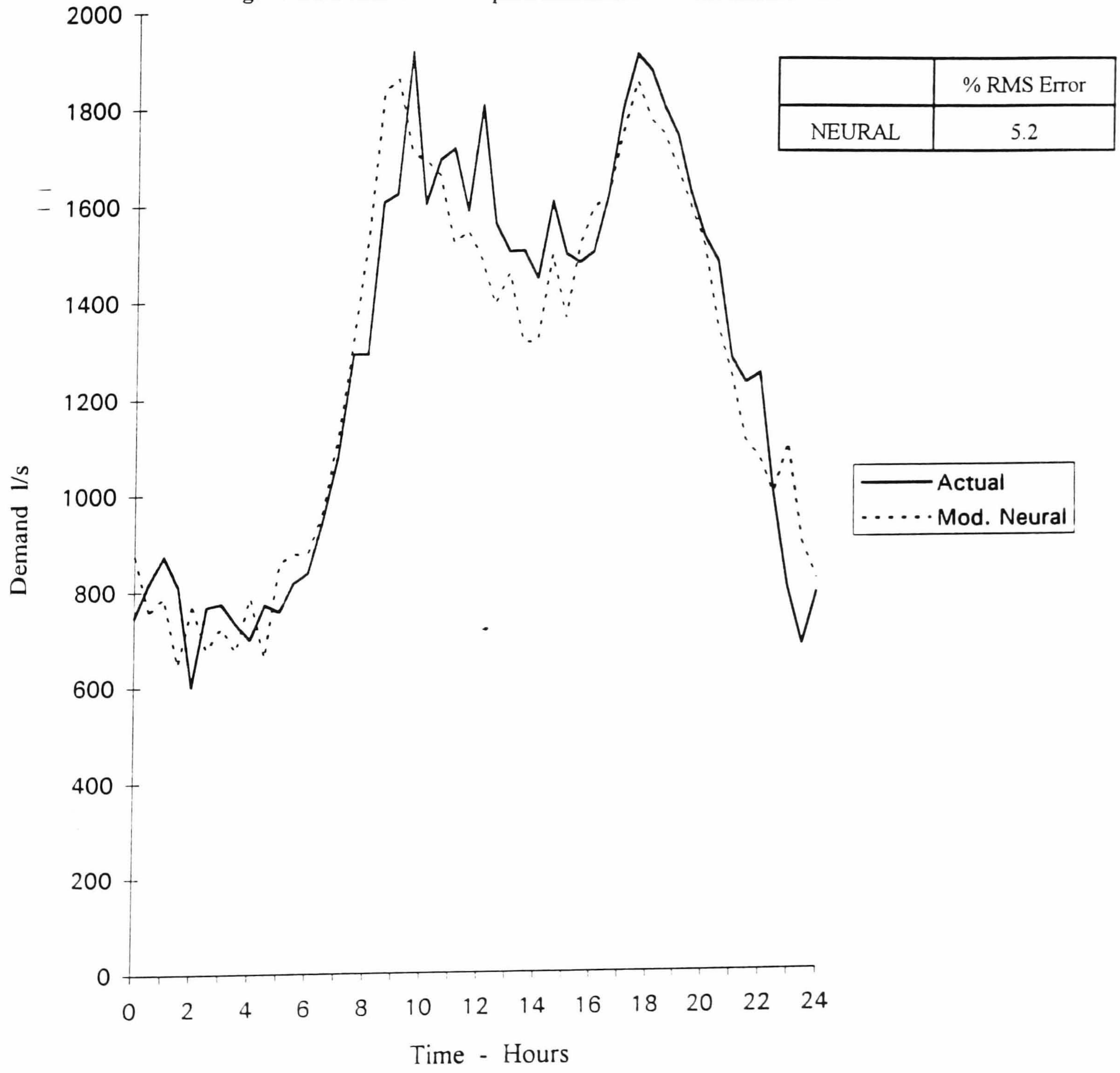
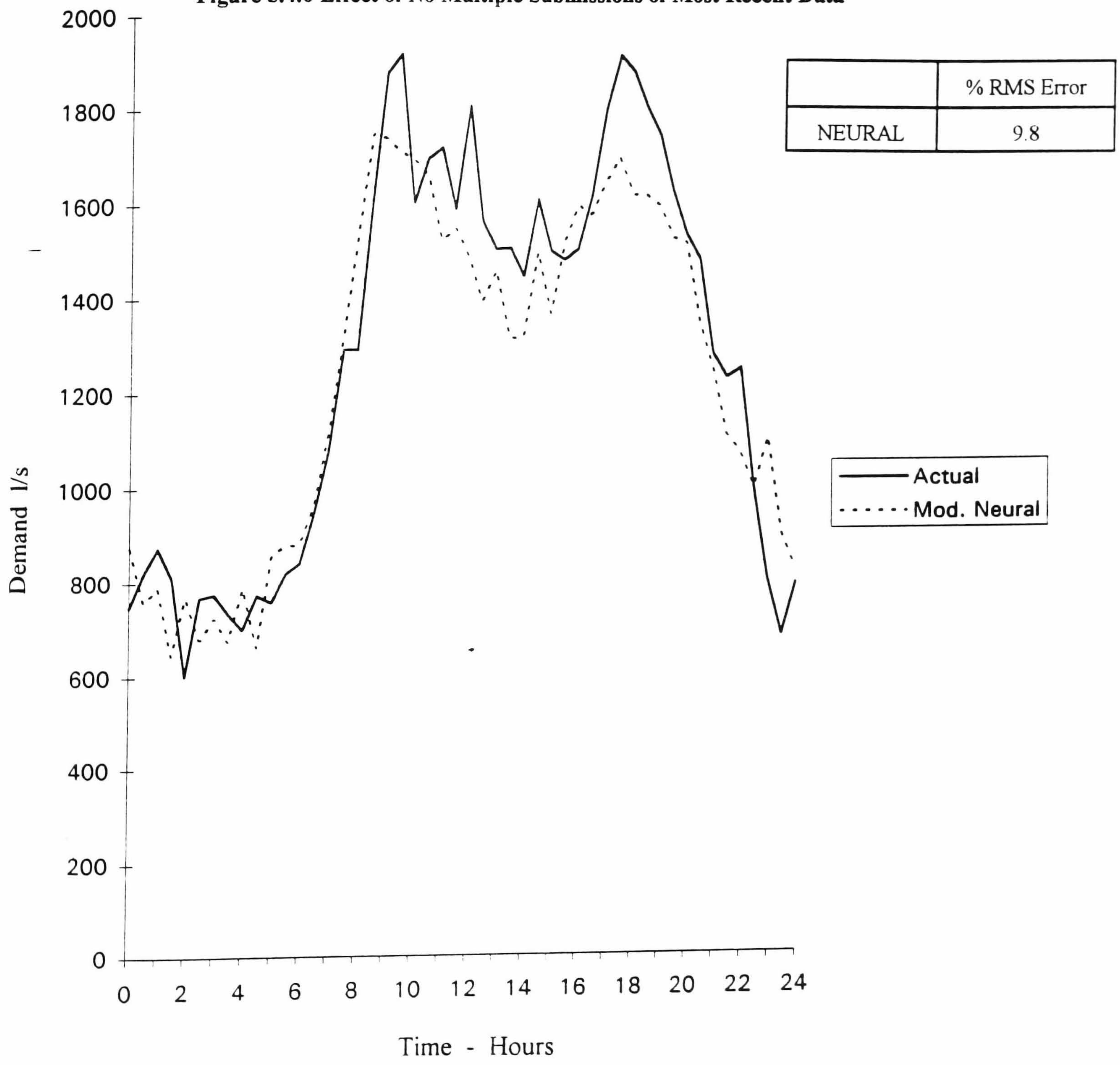


Figure 5.4.6 Effect of No Multiple Submissions of Most Recent Data



network containing the trained weight matrix, the resulting output from the network forms the prediction of the consumption for each half hourly interval for the next 24 hours.

5.5 Results

In the following data plots, the actual consumption for the day is shown in comparison with the prediction profile generated by the linear associator neural network. Figures 5.5.1, 5.5.2 and 5.5.3 show profiles for weekdays with significantly different meteorological conditions. Figure 5.5.1 shows the actual and predicted profile for a dull, normal temperature day, figure 5.5.2 shows the profiles for a sunny fairly hot day and figure 5.5.3 shows the profiles for a hot sunny day during a dry spell.

Figures 5.5.4 and 5.5.5 show actual and predicted profiles for Sundays with contrasting weather conditions.

In the above example it can be seen that the neural predictor achieves an acceptable degree of prediction accuracy over a range of varying meteorological conditions. In contrast to this, Figures 5.5.6, 5.5.7 and 5.5.8 include the prediction profiles generated by the ARIMA predictor as well as the actual and neural network profiles. These plots demonstrate that in changing meteorological conditions the mathematical algorithm fails to react sufficiently rapidly to the changes in demand pattern.

Table 5.5.9 shows the relative prediction performance of the neural network and the ARIMA algorithm over 65 days from April to June 1990.

Table 5.5.9 Results For 65 Days Spring 1990		
	Neural Network	ARIMA
Average Daily Error	7.0	10.5
No. of Days Error >15%	0	8
No. of Days Error >10%	2	17
No. of Days Error >8%	12	30

Table 5.5.9. Prediction performance over 65 days.

Figure 5.5.1 Actual and Neural Net Prediction for Normal Day

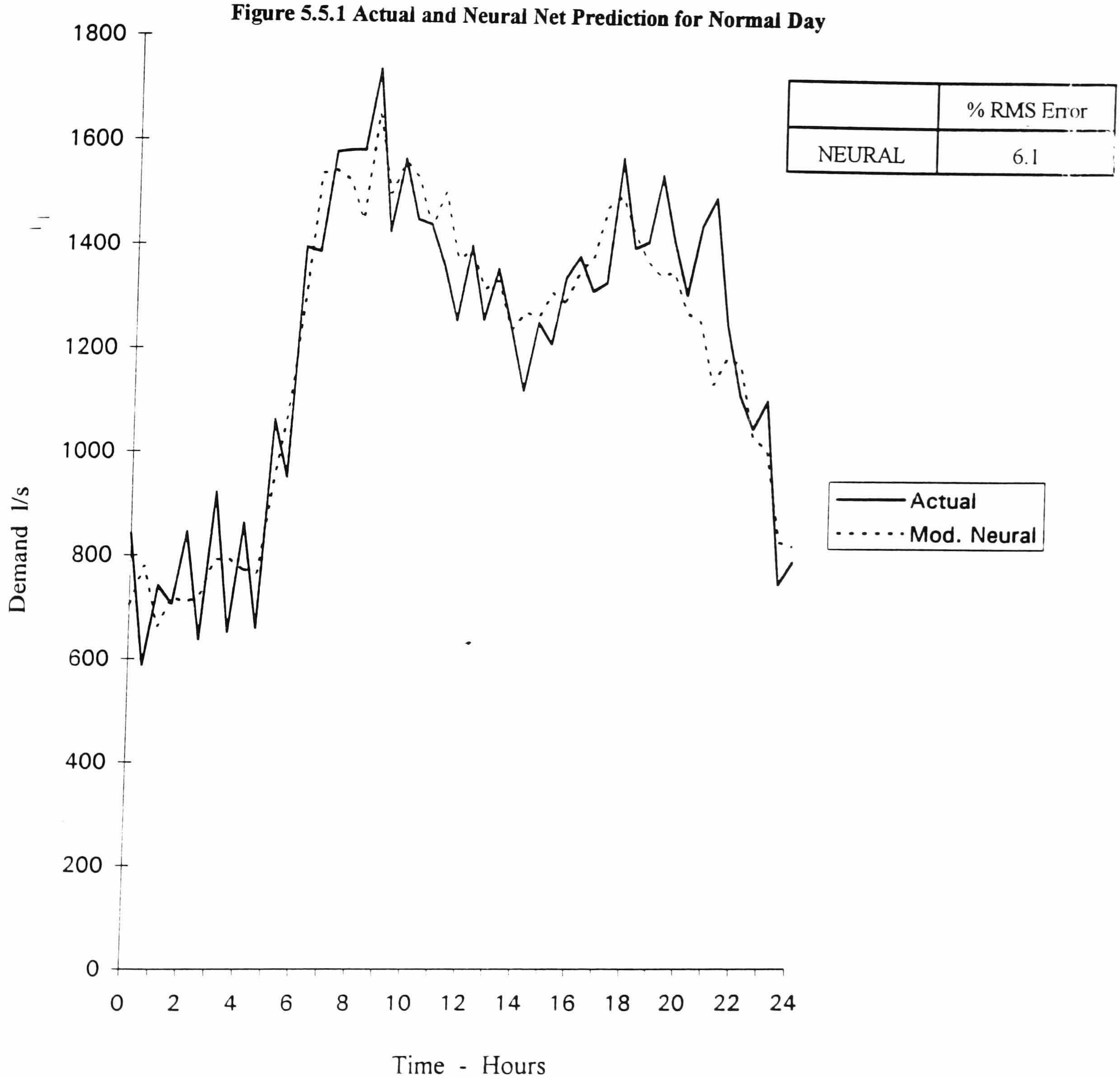


Figure 5.5.2 Actual and Neural Net Prediction for a Moderately Hot Day

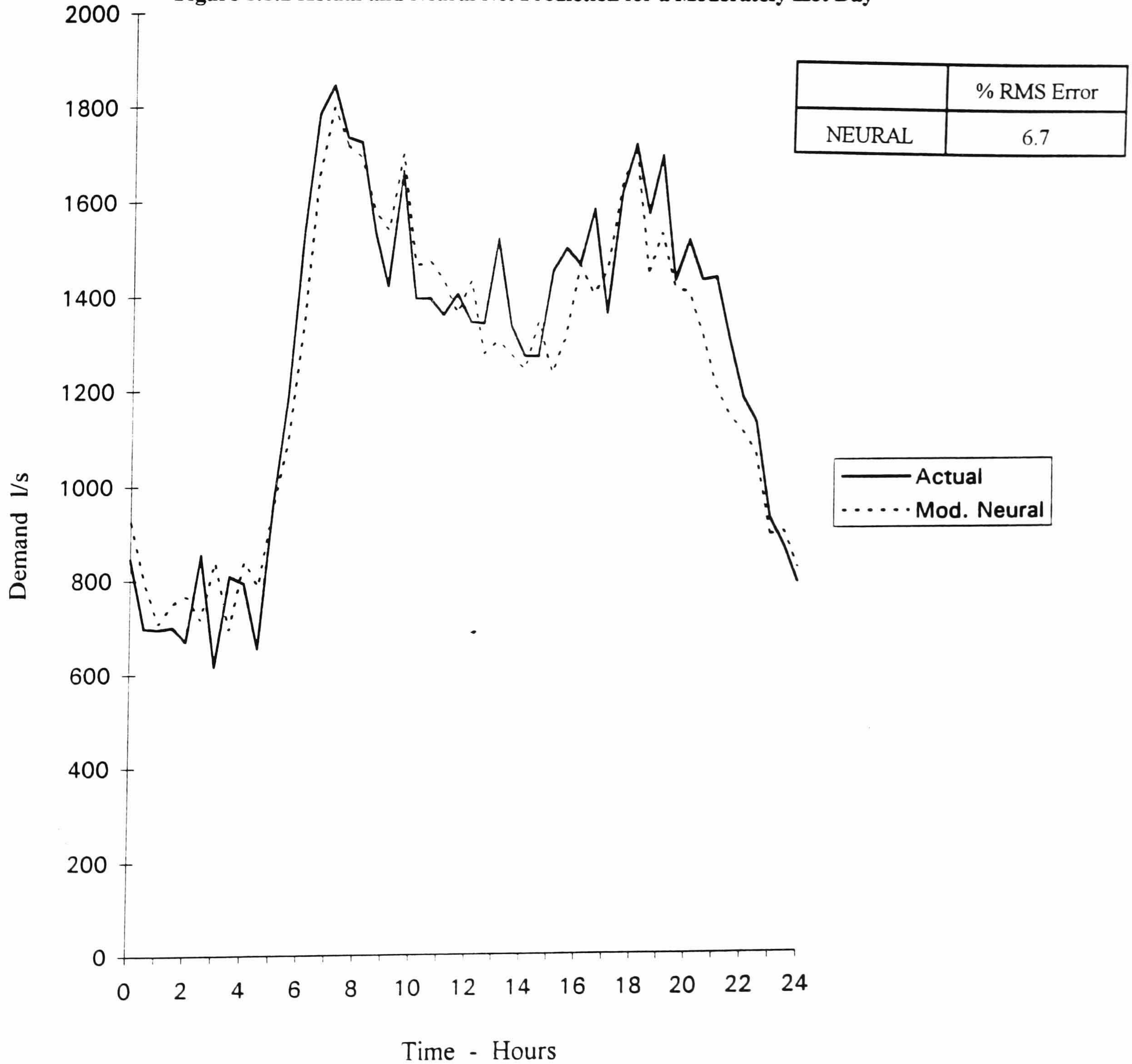


Figure 5.5.3 Actual and Neural Net Prediction for a Very Hot Dry Day

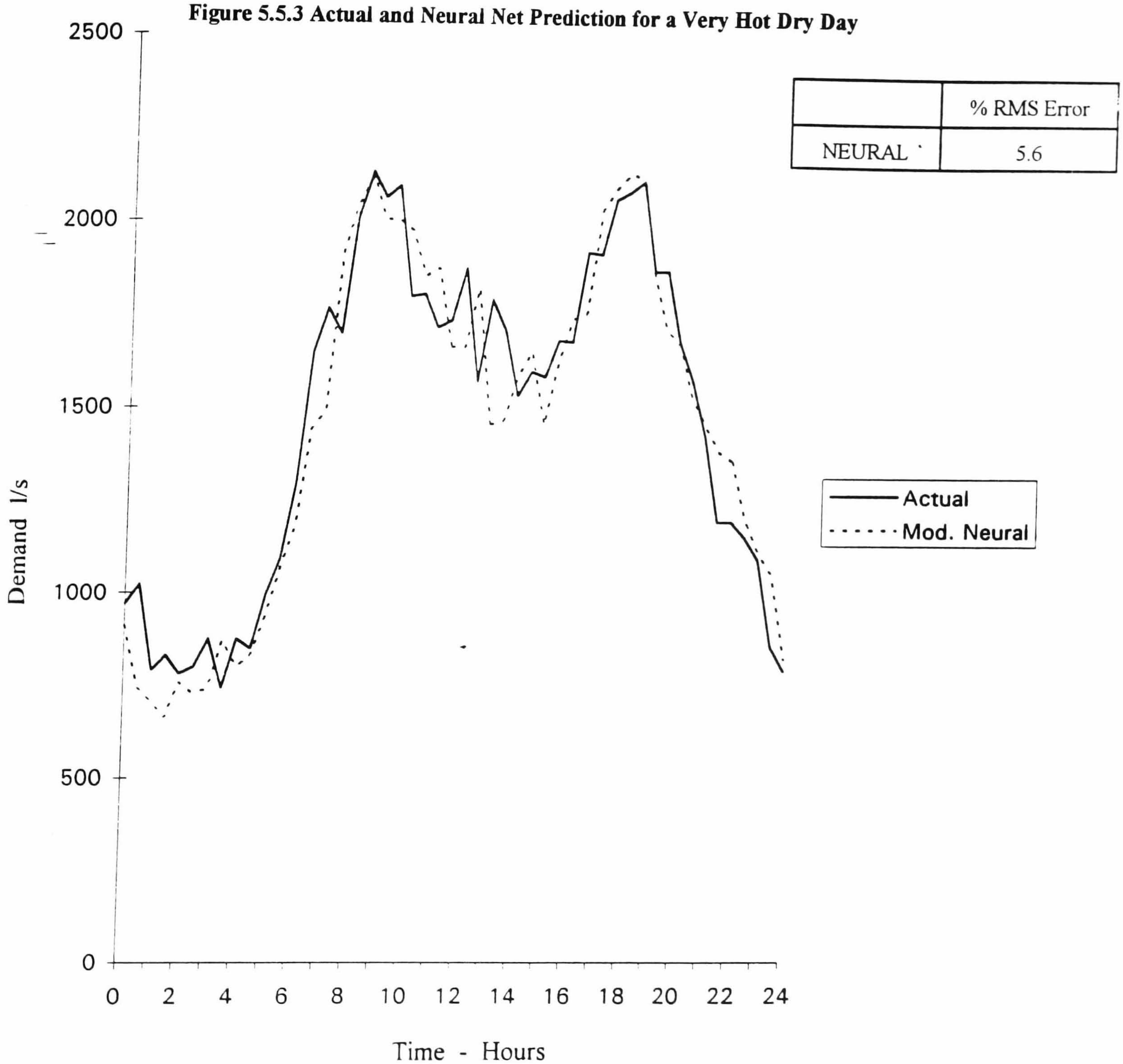
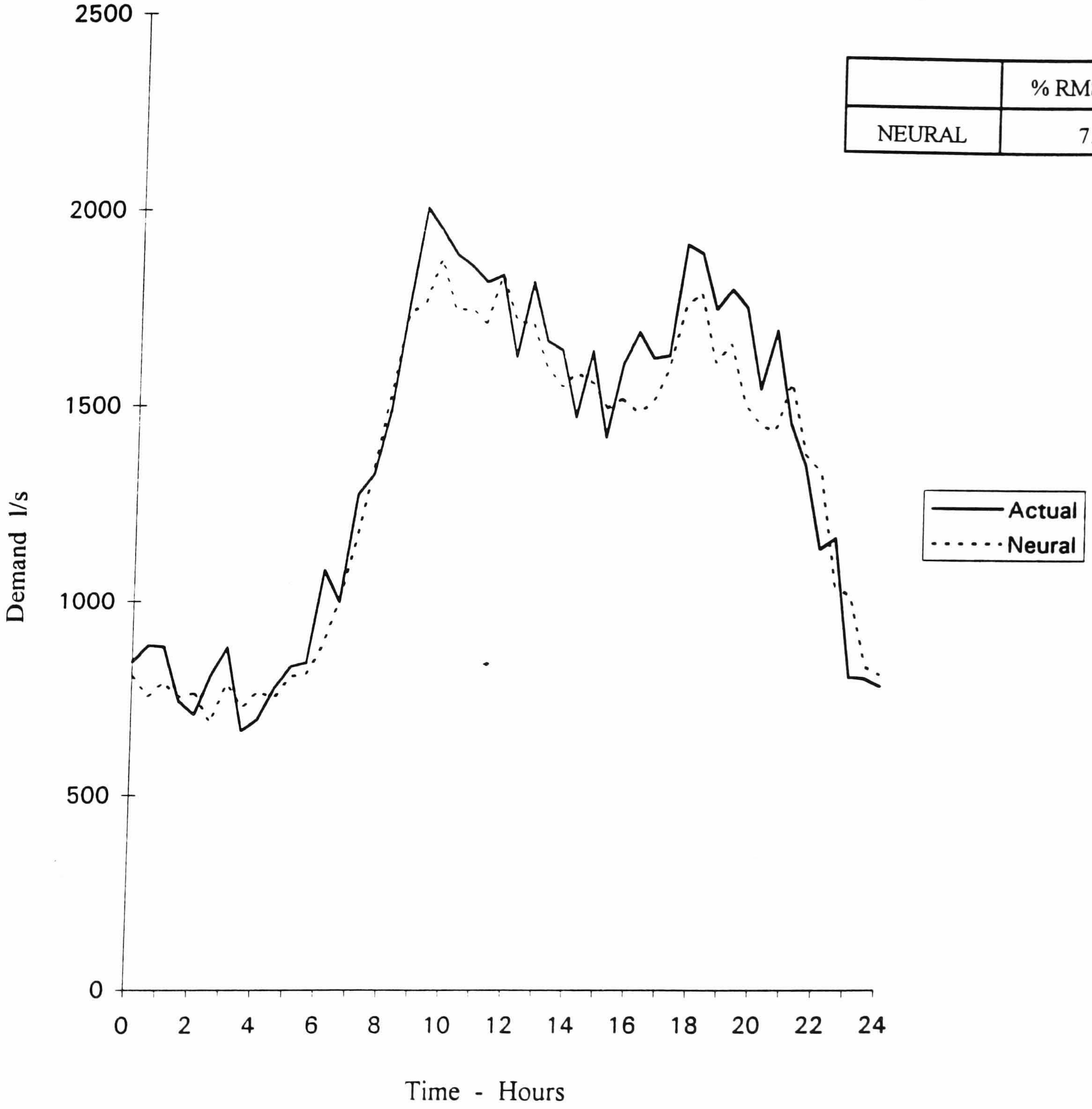
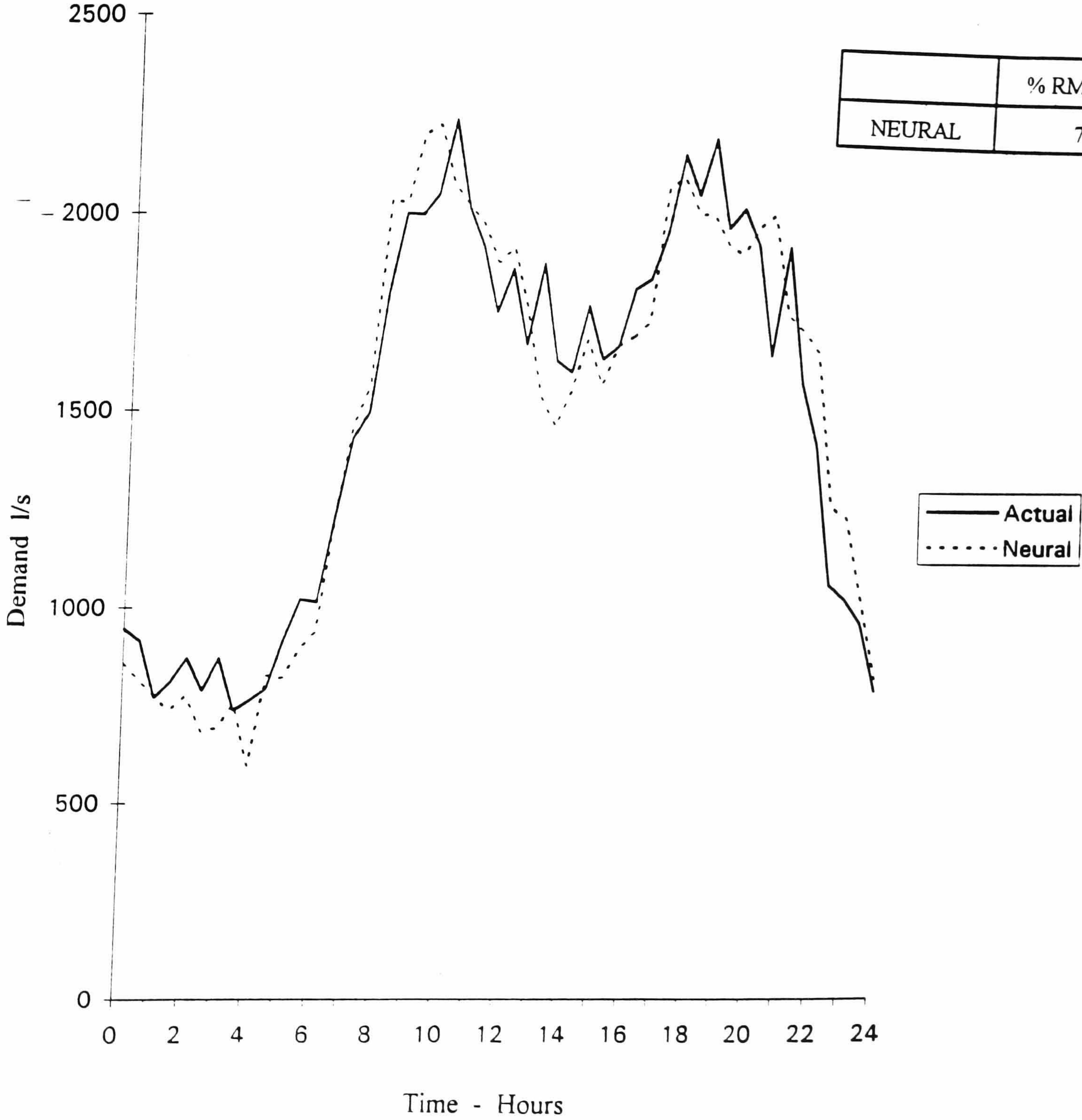


Figure 5.5.4 Actual and Neural Prediction for a Normal Sunday



	% RMS Error
NEURAL	7.3

Figure 5.5.5 Actual and Neural Prediction for Hot Dry Sunday



	% RMS Error
NEURAL	7.1

Figure 5.5.6 Actual, Neural and ARIMA Predictions for a Above Average Temperature Day

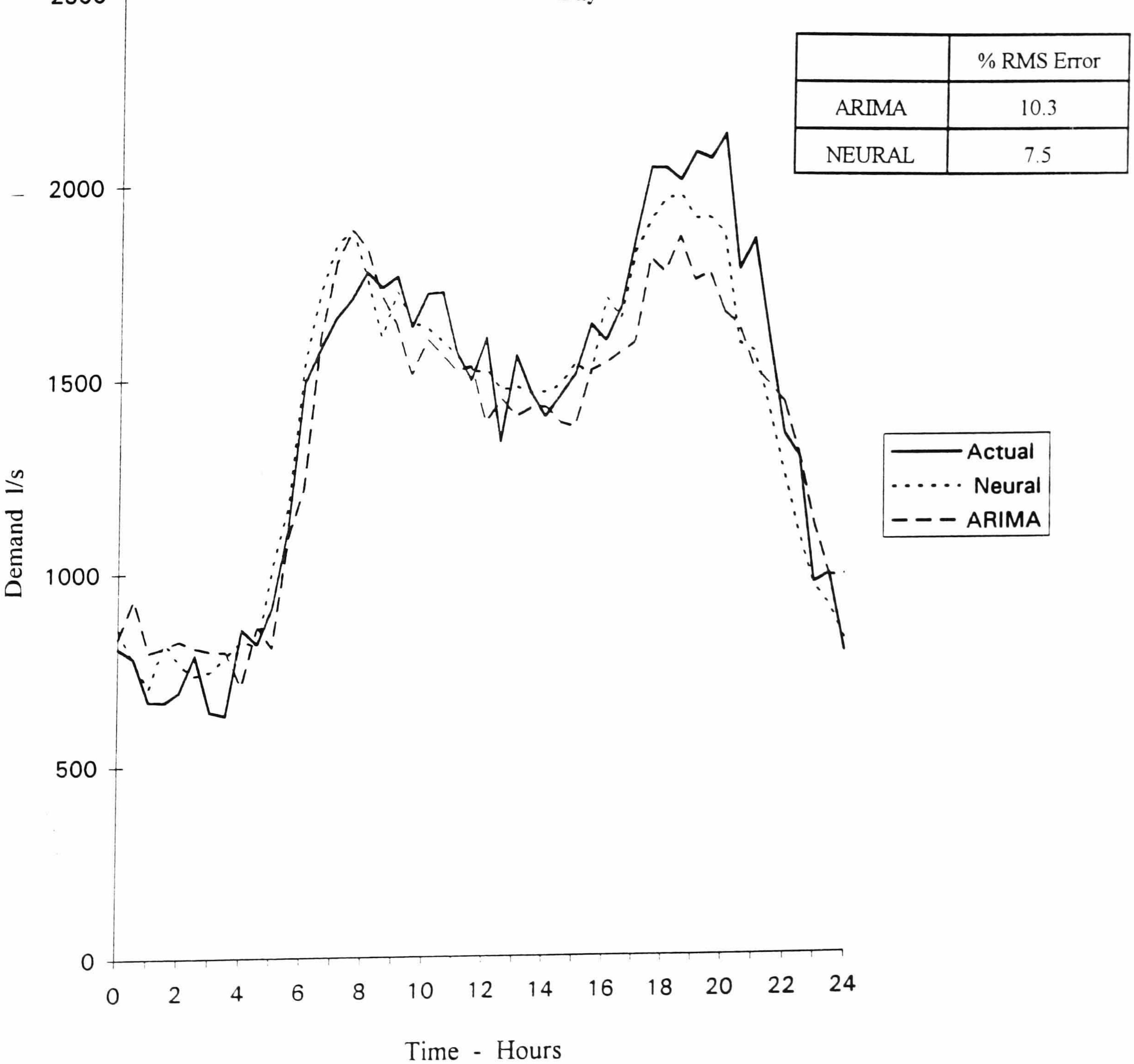
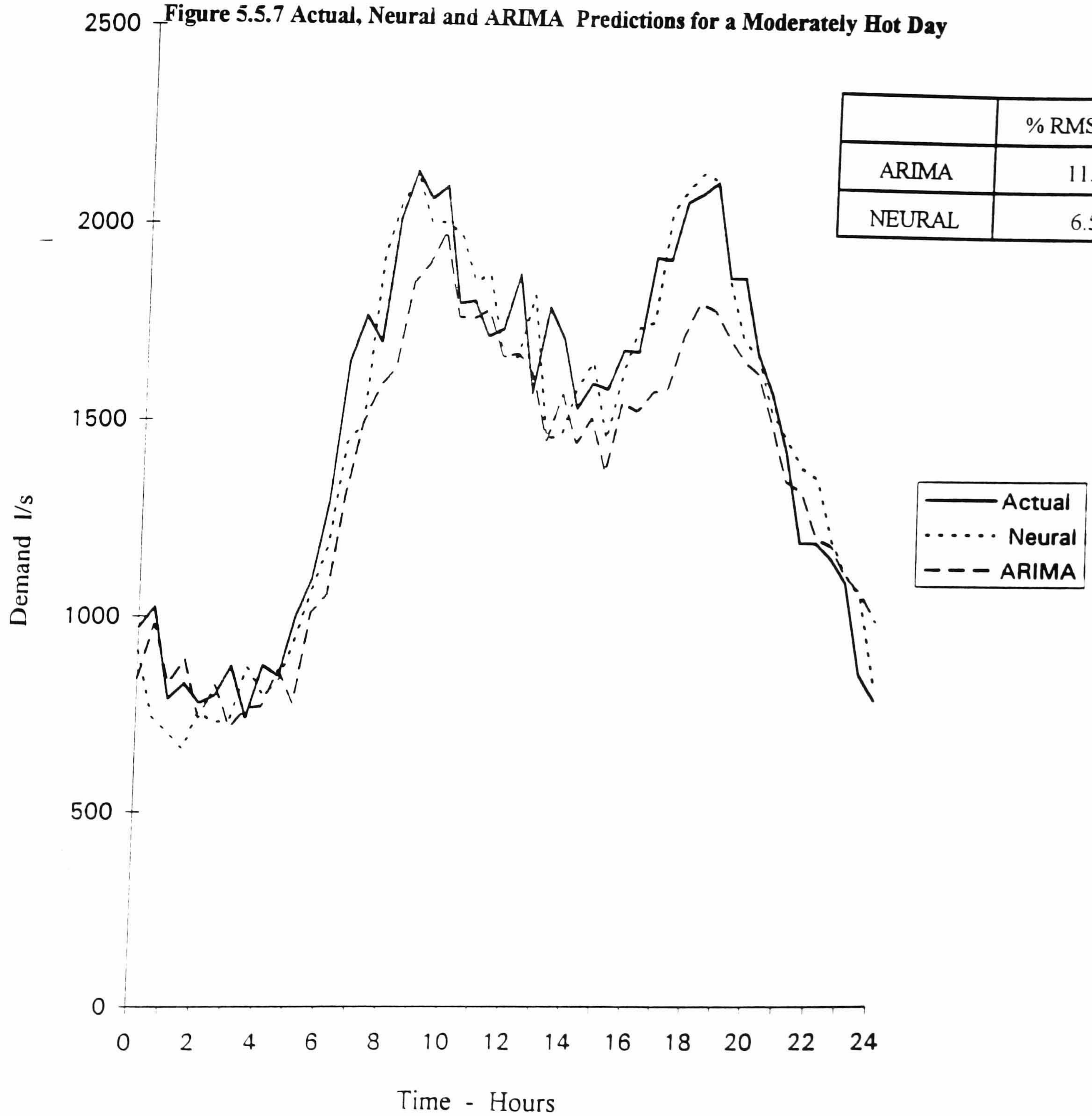
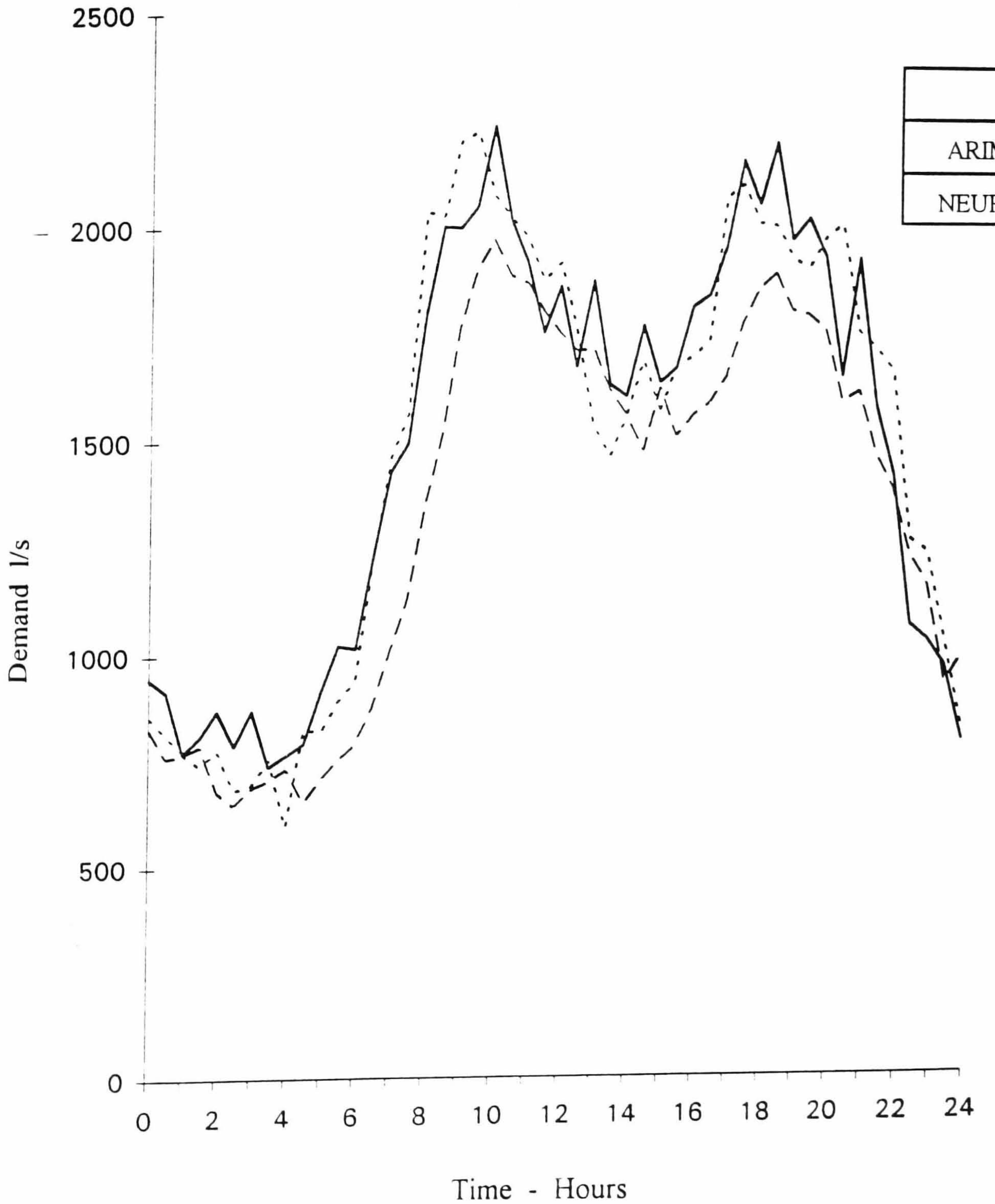


Figure 5.5.7 Actual, Neural and ARIMA Predictions for a Moderately Hot Day



	% RMS Error
ARIMA	11.6
NEURAL	6.5

Figure 5.5.8 Actual, Neural and ARIMA Predictions for a Very Hot Dry Day



	% RMS Error
ARIMA	15.9
NEURAL	7.8

5.6 Discussion

There are several advantages associated with the linear associator network described above. The nature of the problem of accounting for the weather related influences upon water consumption indicates that an attempt to be too specific about the exact effect of a single meteorological variable will potentially lead to a worsening of the predictive capacity of a forecasting system. This is due to the changing nature of the meteorological influences through the seasons and to the subjectiveness of the public perception of the prevailing weather conditions, both of which determine the weather dependent consumption at any particular time.

The day type classification approach provides a suitably 'coarse grained' method of accounting for weather dependent variations in demand. It is capable of absorbing both minor errors in the prediction of weather variables and variations in the relationships between such weather variables and their resultant levels of water consumption. It is a simple task to decide upon a likely day type for the prediction day based on a general weather forecast for the geographical area in question. If the day type chosen is identified at some point during the prediction day as being in error, then a re-run of the network prediction using the new day type is easily completed with no retraining required. The training time itself for the linear associator is of the order of 15 seconds and this time does not vary significantly between different prediction days.

There are potential problems that could arise with the neural network predictor, these are chiefly related to the non availability of sufficient examples of particular day types. For example, there may not be any recent examples of type 3 or type 4 Sundays available in the data set submitted to the program, as a concequence data from type 2 Sundays would have to be used and a potentially significant prediction error would result. There are several possible ways in which this type of problem could be overcome. One approach is to indicate to the rule base described in

Chapter 4 that no examples of one or more day types had been found in the current data set and thereby trigger rules which adjust the predicted demand profile to reduce the adverse effect of the missing data. Alternatively, a more long term approach would be to establish a database of past consumption data spanning at least a year. This would contain examples of all day types for all days of the week. If no examples of a day type exist in the data immediately preceding the prediction day, then a search could be made of the data base to extract the most appropriate daily profiles (in terms of similar time of the year and similar weather conditions) for use in the network training. A scaling factor would be required to account for the long term variations in the overall level of water consumption which may be due to population fluctuation or social change.

CHAPTER 6

INVESTIGATION OF ALTERNATIVE NEURAL NETWORK ARCHITECTURES FOR DEMAND FORECASTING

6.1 Introduction.

As a result of the improvements in demand prediction accuracy that were shown to be achievable with the linear associator network, further investigation was conducted into the performance of neural network prediction systems based on different network architectures and learning algorithms. The network architectures investigated were more complex than that of the linear associator in that they possessed increased number of layers and neurons and the testing of such architectures would indicate whether they possessed a greater potential for successfully mapping the relationship between water consumption and meteorological variations than the linear associator network architecture. Additionally, the more complex architectures provided the opportunity to test the effects of dispensing with the heuristic day type classification routine upon the prediction capacity of a network. The inputs would not be restricted to one of four possible day types, instead, appropriate network architectures were devised that could provide a direct mapping between the values of selected meteorological variables and the resultant shape of the demand profile. The result of these investigations was the development of two separate multi-layer demand forecasting network architectures, each using different learning algorithms. These networks and the results of their application to the same water consumption data as that used for the linear associator network are described in the following sections. Before introducing the detailed descriptions of the two proposed architectures, a brief summary is provided in the following section of research work that has recently been undertaken into the application of neural networks for generating forecasts.

6.2 Recent Research into the Use of Neural Networks for Prediction.

There has been a significant degree of interest within the last three or four years in the possible application of neural networks to the field of prediction. Most of this research relates to load forecasting for electricity supply networks [42,72,76,92,93,111,115] but there have been some papers published that are based on demand forecasting in water networks. One such paper by Cubero [36] introduces the idea of applying up to 15 time lagged example values of past consumption data to a multi-layer feed forward network in order to generate a single value for the predicted load as the network output. The selection of the time lagged values for application to the net is carried out by statistical investigation of a time series of past data in order to determine the correlation between past data points. The system is designed to account for the occurrence of public holidays by the addition of extra input neurons, the output values of these additional neurons being set to a value of 1 when the event which they represent occurs. The system uses an input and output layer separated by two hidden layers of neurons and a simple minimisation function as its training method, the results generated are quoted as being similar to those obtained by using an ARIMA based prediction system.

Further work aimed at providing a comparison between conventional mathematical prediction techniques and a neural network based approach is given in a paper by Atlas Connor and Dambourg [8]. The results from a neural network composed of 6 input neurons, 10 hidden layer neurons and a single output neuron were compared to results produced by the mathematical prediction system used by an American power generation company. Using as input data the hour of the day, the two previous hourly load values, the two previous hourly temperature readings and the current temperature, the neural network generated predictions of the load for the coming hour that were of comparable accuracy to the existing system. Zaiyong Tang, de Almeida and Fishwick [164] found that their feed forward, back propagating networks could out perform Box-Jenkins based mathematical load predictors for time series with short memory. In contrast to this Foster, Collopy and Ungar [52] compare the predictive accuracy of a neural network used as a function approximator for an individual time series with a neural network used to optimally combine traditional mathematical forecasting methods such as linear regression and exponential smoothing. The specific aim of the research was to examine how each approach coped with highly noisy data. A back propagating neural network was trained to simulate the mapping between past values and future values of a time series of chemical process data. The accuracy of the results generated by this network were then compared to the accuracy achieved by the mathematical method selected by a second neural network. This second network uses inputs that are the predictions generated by the mathematical predictors combined with a series of features that characterise the time series under investigation. The output of the network is a score value between 1 and 0 that allows the selection of the best mathematical forecasting method for that time series. Both networks were tested on a large number of time series of varying characteristics and the 'predictor selecting' method was found to produce the most consistent results in terms of prediction accuracy.

There have been a number of papers published that describe the use of neural networks for electrical load prediction that use a three layer perceptron based network structure and back propagation as a learning algorithm [42,111,163]. The perceptron

[128] being a neural network that consists of one or more processing elements that have a structure as shown in Figure 6.2.1.

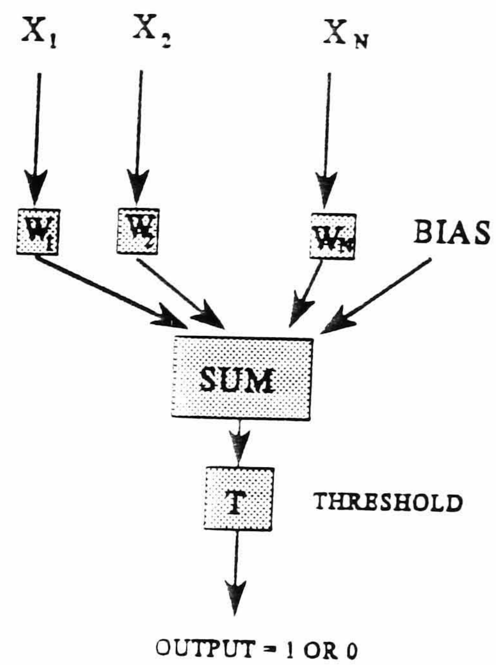


Figure 6.2.1 Diagram of a Perceptron.

A perceptron element has an input consisting of an n+1 dimensional vector $\mathbf{x} = (x_0, x_1, x_2, \ldots, x_n)$ where x_0 is permanently set to 1 (this is termed a bias input). The output of the perceptron is 1 if the weighted sum of the inputs is greater than or equal to zero and the output is 0 if the weighted sum of the inputs is less than zero. Rosenblatt [128] showed that given linearly separable classes, a perceptron network will, in a finite number of training passes, develop a weight matrix that will separate classes of input vectors. The network is therefore carrying out a pattern classification task and in the cases of the load prediction applications the networks learn to classify particular characteristics of applied load profiles and generate the appropriate future load value. Pattern classification is also the aim of a network described by Hsu and Yang [77] who use a two layer Kohonen network similar in structure to the Kohonen layer that comprises part of the network outlined in section 6.4 of this Chapter. The aim of the network training process is ensure that the application of a profile of a particular type of day (Sunday, Saturday or Holiday) will always fire the same output neuron. The trained network is then used to identify the type of day for the prediction day and based on this classification ten profiles for days of the same type are used to generate an average load profile.

Other neural network load forecasting applications also use meteorological information as inputs to the network. Bacha and Meyer [9] proposed a system that uses a number of interconnected three layer networks that take account of the current weather trends in generating predicted gas load values. The aim of the interconnecting networks is to extract during the training process, the relevant patterns from the input weather parameters and associate them with different output levels. Bacha and Meyer suggest that a complex network architecture is required in order to capture all the information on the relationships between weather variables and load. The architecture proposed is a linked series of 24 networks one for each hour of the day, each with 12 inputs of weather data. The interconnected nature of each of the three layer sub-networks allows the single output neuron of each sub-network to be influenced by the two sub-networks located either side of it. In this way the trends in the weather conditions from hour to hour over the 24 hour period are taken into account. The results from this work based on 22 days of data were encouraging and the authors intend to expand the scope of the research in the future.

The applications described below explore some alternative architectures investigated as part of the work conducted for this thesis that aim to extract the maximum amount of information from the available weather input variables and use that information to generate forecasts that can successfully account for meteorological influences on water demand.

6.3 Backpropagation

6.3.1 Introduction to the Backpropagation Network

The backpropagation neural network is a mapping network that has been applied to numerous problems in a range of different fields of research, for example, pattern recognition, noise cancellation, process control etc. [5,113,114,130]. The early development of the backpropagation algorithm was the result of research undertaken independently by several groups and individuals in the early 1970's. However, it was the work of Rumelhart [130] which first comprehensively described the backpropagation algorithm and provided a theoretically sound basis for the training of muli-layer neural networks. This in turn led to a more widespread awareness of the capabilities of neural networks and their possible range of application.

Although there have been examples of very complex backpropagation network architectures, the basic network and its algorithm are relatively simple. As with the linear associator, the back propagation network consists of fully interconnected layers of individual processing elements or neurons. In its simplest form, the network consists of an input layer, an output layer and a single intervening 'hidden' layer. However, in more complex architectures the number of 'hidden' layers can be greater than one. The network learns to carry out a mapping function between bounded subsets by means of the application of training examples $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots, (\mathbf{x}_k, \mathbf{y}_k), \ldots$ of the desired mapping, where $\mathbf{y} = f(\mathbf{x})$. The backpropagation algorithm provides the means by which the network can be trained to correctly reproduce the relationship between the input and output examples.

6.3.2 Structure of the Backpropagating Network

Figure 6.3.1 shows a diagrammatic representation of the structure of a simple three layer backpropagating network. An input vector is applied to the units of the input layer **A** and these signals are distributed via weighted connections to the processing units of the second layer **B**, this second layer is termed 'hidden' as it has no direct connections outside the network. The neurons of the second layer **B** apply a transfer function to the incoming signals, the result of which is then propagated, via weighted connections, to the processing units of the output layer **C**. The units of layer **C** in turn apply their transfer function to the arriving signals from layer **B** to generate the output signal for each unit. Each of the output signals of the layer **C** units constitutes an element of the network output vector. In addition to the feed forward connections described above, each processing element of the hidden layer **B** receives an error feedback signal from the processing elements of the layer above it (layer **C**), these signals carry the information that is used to adjust the weights of the incoming connections to the hidden layer neurons.

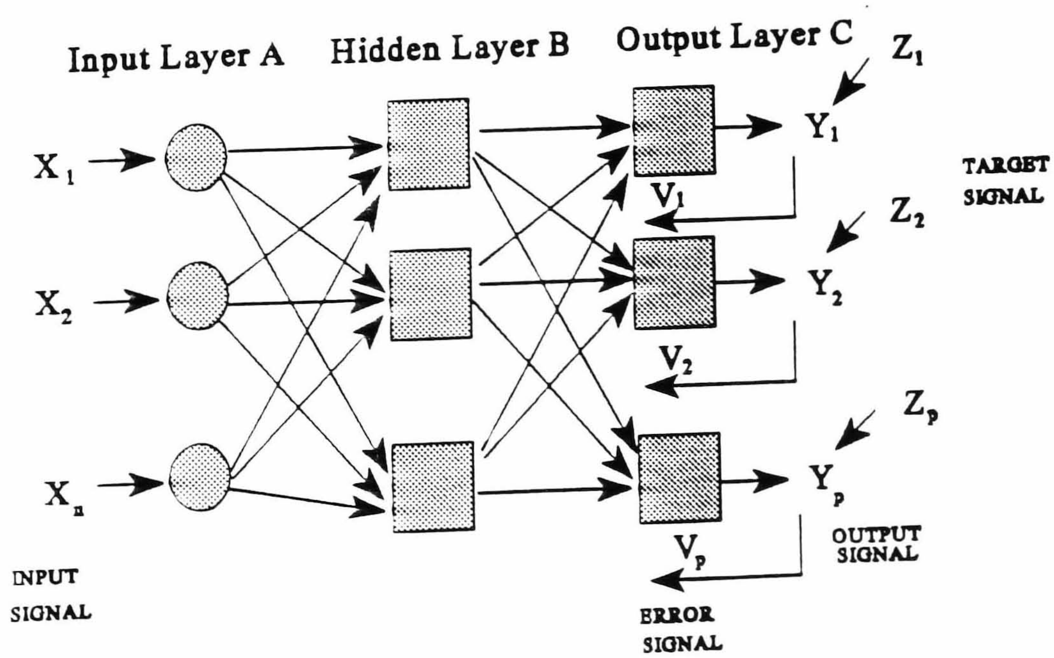


Figure 6.3.1 Structure of a Three Layer Backpropagating Network.

As with the linear associator network described in the previous chapter, the network learns by the process of submitting example pairs of input/output vectors to the network and attempting to minimise the errors between the desired and actual output by the adjustment of the network weights. However, the linear associator had no mechanism whereby the errors observed at the output layer could be used to adjust any weights other than those that are associated directly with the processing units that compose the output layer itself. Hence, the network is restricted to possessing a single set of adjustable network weights. The success of the application of the backpropagation network in numerous different fields, is based on its ability to solve non-linear problems by propagation of the errors observed at the output layer back to the preceding hidden layer(s) of the network and hence allow the adjustment of the weights associated with the processing elements of these preceding layers. The details of this method are described in section 6.3.3.

The structure of one of the individual neurons that compose both the hidden and output layers of the backpropagating network is shown in Figure 6.3.2. Note that the neurons of the input layer differ from the other neurons in Figure 6.3.2, in that they apply no transfer function and serve simply to distribute the incoming signals to the neurons in the first hidden layer.

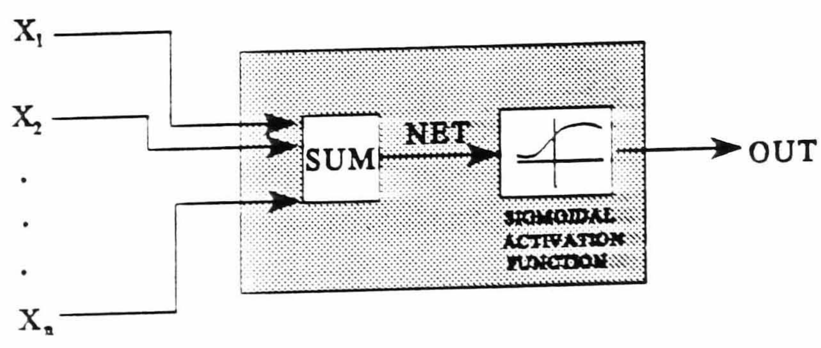


Figure 6.3.2 A Neuron From the Hidden or Output Layer.

The signal vector $x_1, x_2, \ldots, x_n$ arriving at a neuron m in either the hidden or the output layer are multiplied by the associated connection weights $w_{m1}, w_{m2}, \ldots, w_{mn}$ where m indicates the identifier of the neuron within its particular network layer. The summation of these products gives the value of NET for the neuron i.e.

$$NET_m = \sum_{i=1}^{n} w_{mi} x_i \tag{6.3.2.1}$$

The output signal of the neuron m is then generated by applying a squashing function to the value of NET. Many possible functions have been described which can be used to perform this squashing function, however, in this work the sigmoidal function shown below has been chosen.

$$OUT_p = 1/\left(1 + e^{-NET_p}\right) \tag{6.3.2.2}$$

The above sigmoid transfer function has little effect upon strongly positive or negative signals, but amplifies weaker signals that are close to zero.

As mentioned previously the neurons in the hidden layer(s) also have connections $v_1, v_2, \ldots, v_p$ that propagate error signals back from each of the neurons to which they provide an output signal. Hence, in figure 6.3.1 the p neurons in layer **C** each provide an error feedback signal to the neurons of layer **B**. During network training these connections are used to propagate the error information back through the network that allows weight updating to take place.

6.3.3 Training the Network

The training of the network is achieved by the submission to the network of multiple example input/output vectors $(\mathbf{x}_1,\mathbf{y}_1),(\mathbf{x}_2,\mathbf{y}_2),\ldots,(\mathbf{x}_L,\mathbf{y}_L)$ where L is the number of examples in the training set. An input vector $\mathbf{x}_k$ is applied to the network and a corresponding output vector $\mathbf{y'}_k$ is generated by the network, this is compared to the desired output $\mathbf{y}_k$ and the weights of the network are updated in such a way that the difference between the desired and actual output is minimised. The weight updating process for a network with n input layer neurons, m hidden layer neurons and p output layer neurons is as follows:

For each neuron i in the output layer the error between the actual (y'_i) and desired (y_i) output is calculated and fed through a simplification of the transfer function shown in equation 6.3.2.2 to give δ_i

$$\delta_i = y'_i(1-y'_i)(y_i - y'_i) \qquad \text{For } i = 1,2,\ldots,p \qquad (6.3.3.1)$$

The change in value for a weight w_{ji} associated with the connection between neuron j in hidden layer **B** and neuron i in the output layer **C** is given by:

$$\Delta w_{ji} = \alpha\delta_i OUT_j \qquad \text{For } j = 1,2,\ldots,m \text{ and } i = 1,2,\ldots,p \qquad (6.3.3.2)$$

Where α is the training rate (small positive constant) and OUT_j is the output signal of neuron j in the hidden layer **B**.

In order to adjust the weights associated with the signals arriving at hidden layer **B**, the δ values as calculated above for each of the p neurons in the output layer are propagated back through the weights updated by equation 6.3.3.2 and used to calculate the δ values for each of the m neurons in layer **B**.

$$\delta_j = OUT_j(1 - OUT_j)(\sum_{i=1}^{p} \delta_i w_{ji}) \qquad \text{For } j = 1,2,\ldots,m \qquad (6.3.3.3)$$

Where OUT_j is the output signal of neuron j of hidden layer **B** and $(\sum_{i=1}^{p} \delta_i w_{ji})$ is the sum of the error signals propagated back to neuron j from the output layer. The value of δ_j can then be used to update the weights associated with the inputs to neuron j using

$$\Delta w_{hj} = \alpha \delta_j OUT_h \qquad \text{For } h = 1,2,\ldots,n \text{ and } j = 1,2,\ldots,m \quad (6.3.3.4)$$

Where Δw_{hj} is the change in the value of the weight associated with the connection between neuron h of the input layer **A** and neuron j of the hidden layer **B** and OUT_h is the output signal of neuron h.

Training is considered complete when the errors between the network generated output vectors $\mathbf{y}'_k$ and desired output vectors $\mathbf{y}_k$ are minimised for the whole training set $k = 1,2,\ldots,L$.

6.4 A Backpropagation Network Application in Demand Forecasting

6.4.1 Structure of the Network

A backpropagation network design was developed to generate 24 hour demand forecasts based on the same consumption data as that used for the linear associator network i.e. half hourly consumption totals from the Slough and High Wycombe areas of Thames Water's Chilterns Division. The architecture of the network is shown in figure 6.4.1. As can be seen the network consists of 5 input units, a hidden layer containing 20 processing units and an output layer containing 48 processing units.

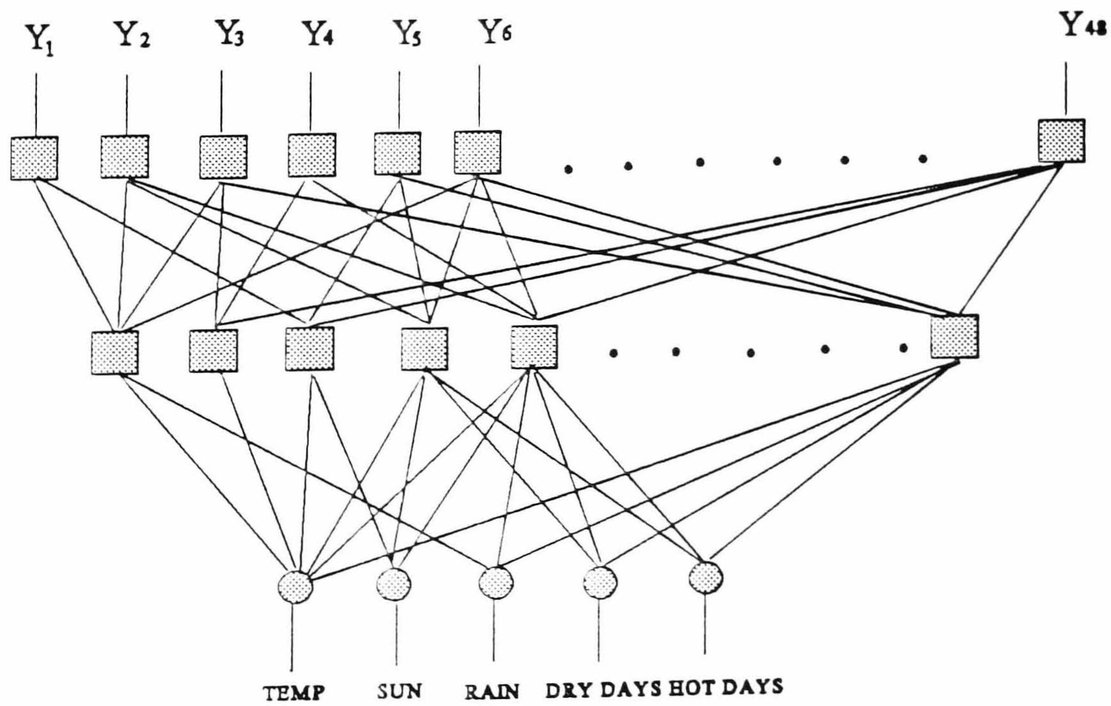


Figure 6.4.1. Structure of the demand forecasting backpropagation network.

The primary purpose behind the investigation of a backpropagating network for the generation of demand forecasts that could account for meteorological influences, was that the network provides the possibility of mapping directly between the individual values of appropriate meteorological variables (temperature, rainfall etc.) and the resultant level of demand. In the linear associator network the representation of the relationship between the meteorological variables and the resultant demand was made relatively coarse by the invoking of heuristics which grouped meteorologically similar days into one of four possible day types. This had the effect of making the mapping task between input and output vectors simpler, but in turn the day type classification process may have been masking important information on the relationship between particular weather variables and demand. The back propagation network provides a method of achieving the mapping between weather variables and demand without the simplification inherent in the classification process used by the linear associator.

The input to the network is a vector whose elements are: maximum temperature for the day in degrees Celsius, total number of hours of sunshine, total rainfall in mm, an antecedent dry day indicator and an antecedent hot day indicator. The antecedent dry day indicator is an integer that can be set to one of three values dependent upon the number of dry days that precede the day in question, it is set to 2 if the number of antecedent dry days is greater than 6, it is set to 1 if the number is greater than 4 and 0 if less than 4. A similar method is used for the hot day indicator, it being set to 1 if 4 or more immediately preceding days have maximum temperatures that exceed 19 degrees Celsius, otherwise the hot day indicator is set to 0. These dry day and hot day indicators were included in the inputs to the network to represent the accumulative effects dry and hot weather conditions have upon the consumption of water in a domestic situation. They reflect primarily the increased usage due to garden watering but also include the increase in other domestic water consuming activities such as washing and cleaning. Figure 6.4.2 shows how the total daily water consumption varies with mean daily temperature in the Slough and High Wycombe areas and figure 6.4.3 shows the variation in consumption in relation to the number of preceding dry days.

Figure 6.4.2 Variation in Total Daily Demand With Max. Temperature

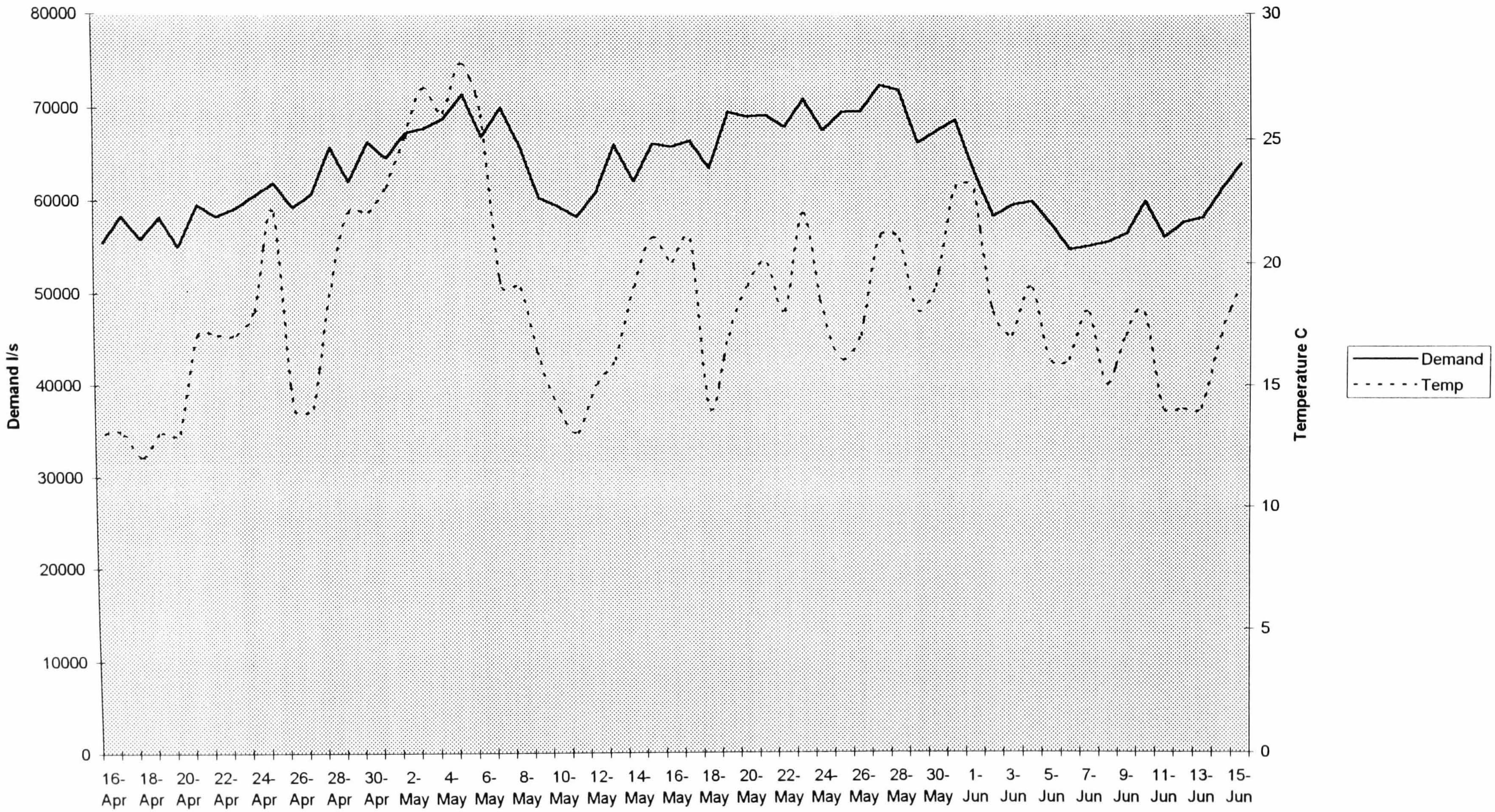
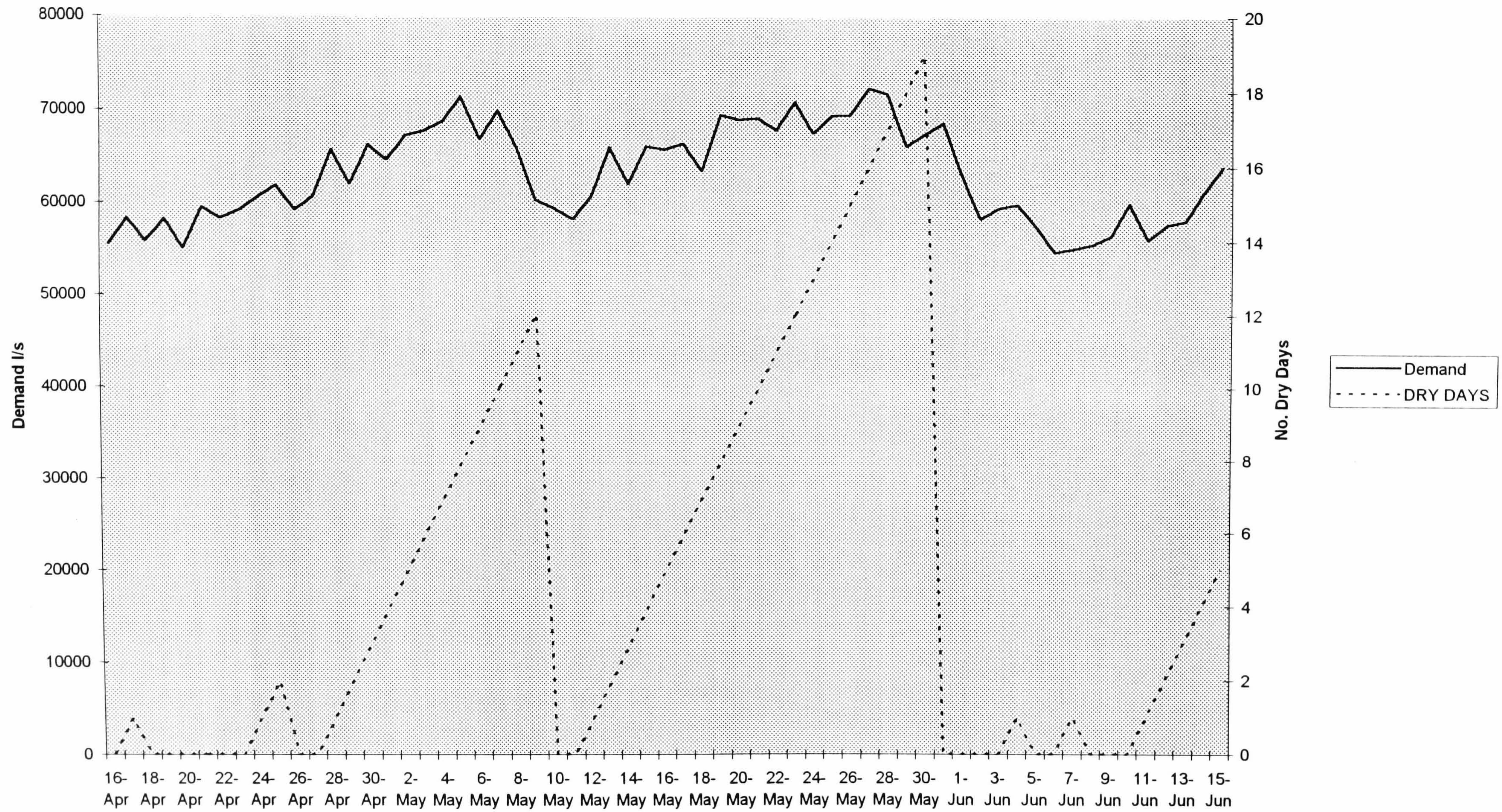


Figure 6.4.3 Variation of Total Demand With No. Dry Days



The output layer of the backpropagating network consists of 48 neurons, these correspond with the 48 half hourly data points of a 24 hour demand prediction. For the hidden layer, there is no formula that will provide the ideal number of hidden layer neurons for a given application or network architecture. Prototype networks were therefore tried with varying numbers of hidden layer neurons (10, 20, 35, 50) and the most successful in terms of network stability and convergence was found to be a 20 neuron hidden layer.

6.4.2 Training the Network

For each day to be predicted, in order to create a training set for the demand forecasting backpropagation network, the available past consumption and meteorological data is processed by a FORTRAN subroutine. The subroutine stores 28 days of the most recent consumption data and the corresponding 28 days of meteorological data, which includes calculating the correct antecedent dry day and hot day indicators. The resulting vector pairs of weather data and consumption data are then used to train the network, the weather data vector being applied to the network and the consumption data being compared to the network output vector.

Training is continued until the error between the network output and the target consumption for each training vector pair within the training set falls below a threshold value.

6.4.3 Results

As with the linear associator network the backpropagation network was tested on Thames Water consumption data from the Slough and High Wycombe areas that covered a period of 4 months in the spring/summer of 1990.

Unlike the data for the linear associator network, the training consumption data for the backpropagation network was not subdivided into weekday and weekend day types. The reason for this was that the aim of this particular element of the research was to determine if there was a relationship between the demand level and the prevailing meteorological conditions that could be learnt by the network. It was assumed that this relationship would be applicable on both weekdays and weekends.

It was found that the training times for each prediction day were predominantly of the order of 20 seconds, however, there were examples of prediction days for which the training times were in excess of three minutes. No examples were encountered where the network failed to converge during training. Predictions were generated using the backpropagation network for each day over a total of 61 days from 16 April to 15 June.

Figures 6.4.4 to 6.4.7 show the prediction profile generated by the backpropagation network in comparison with the actual consumption profile. As can be seen, the predictions in figures 6.4.4 and 6.4.5 achieve a good match with the actual profile, however, the examples in figures 6.4.6 and 6.4.7 show that the accuracy of the backpropagation network is not consistent. Table 6.4.8 shows the accuracy achieved by the backpropagating network over 65 prediction days and compares this accuracy with that achieved by both the linear associator network and the ARIMA algorithm.

Table 6.4.8 Results For 65 Days Spring 1990			
	ARIMA	Linear Associator	**Backpropagation Network**
Average Daily Error	10.5	7.0	**11.8**
No. of Days Error >15%	8	0	**10**
No. of Days Error >10%	17	2	**19**
No. of Days Error >8%	30	12	**27**

Table 6.4.8 The relative accuracy of the backpropagation network.

Figure 6.4.4 Actual and Backpropagation Net Prediction 1

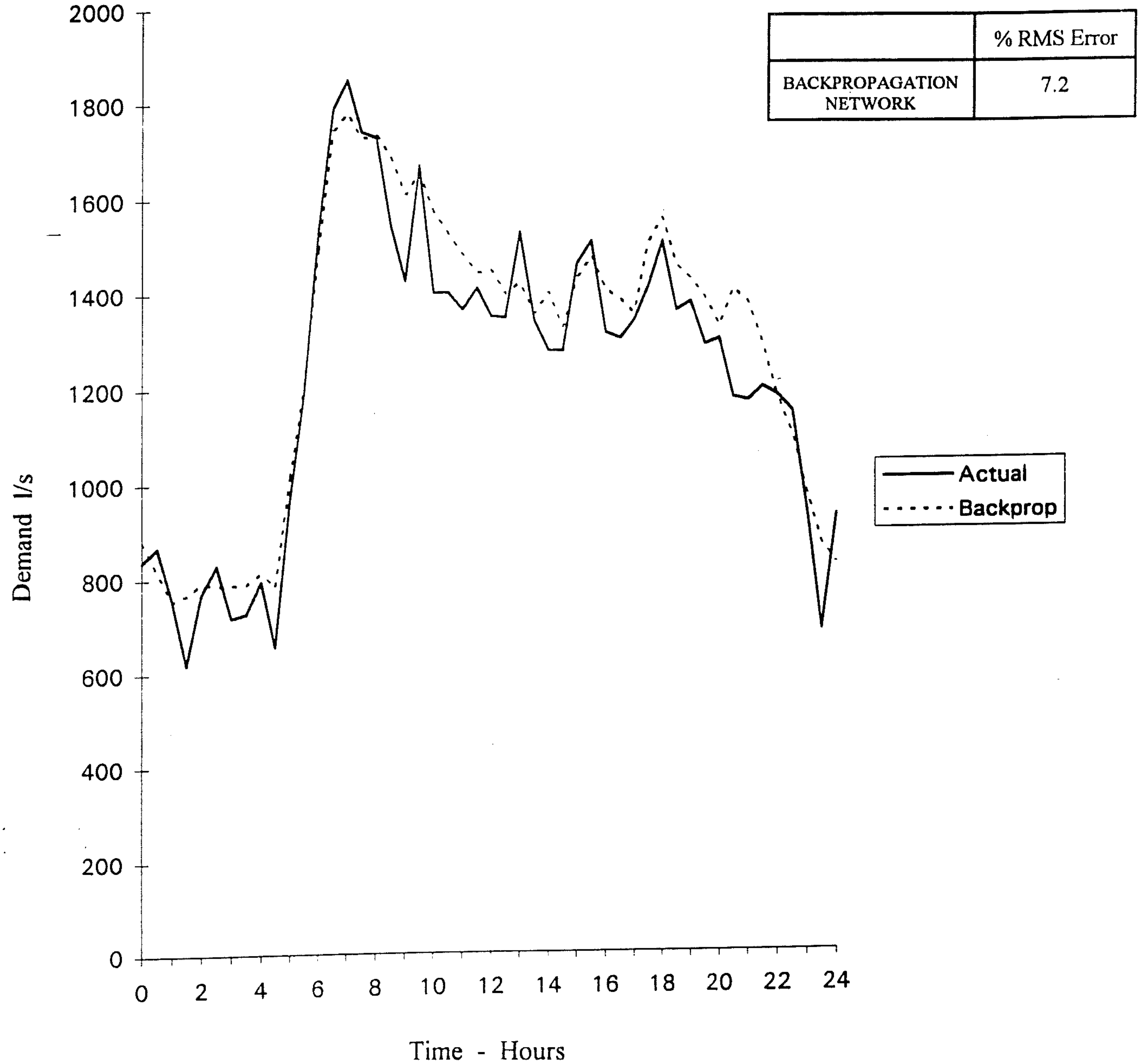
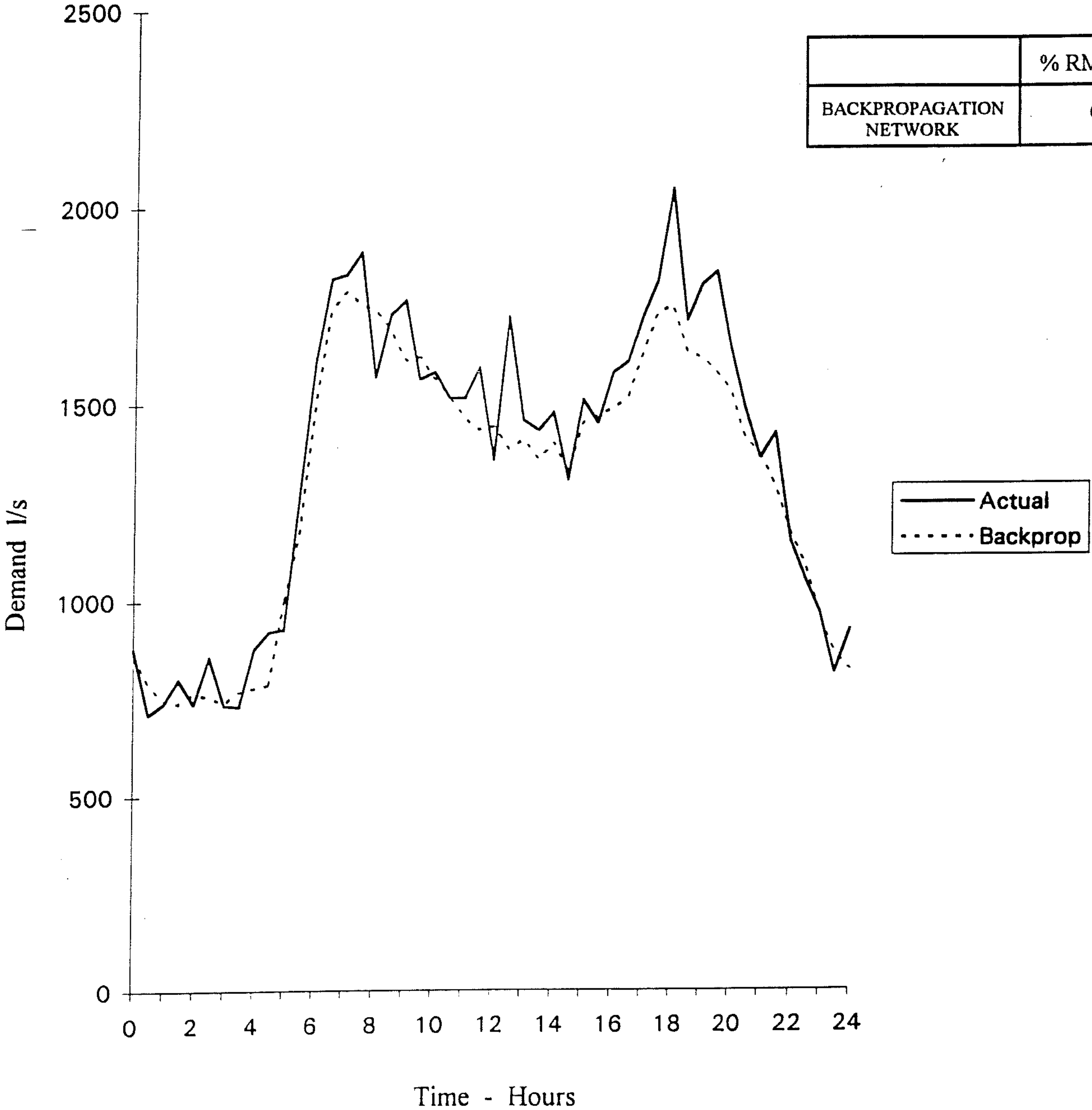


Figure 6.4.5 Actual and Backpropagation Net Prediction 2



	% RMS Error
BACKPROPAGATION NETWORK	6.8

Figure 6.4.6 Actual and Backpropagation Net Prediction 3

	% RMS Error
BACKPROPAGATION NETWORK	10.8

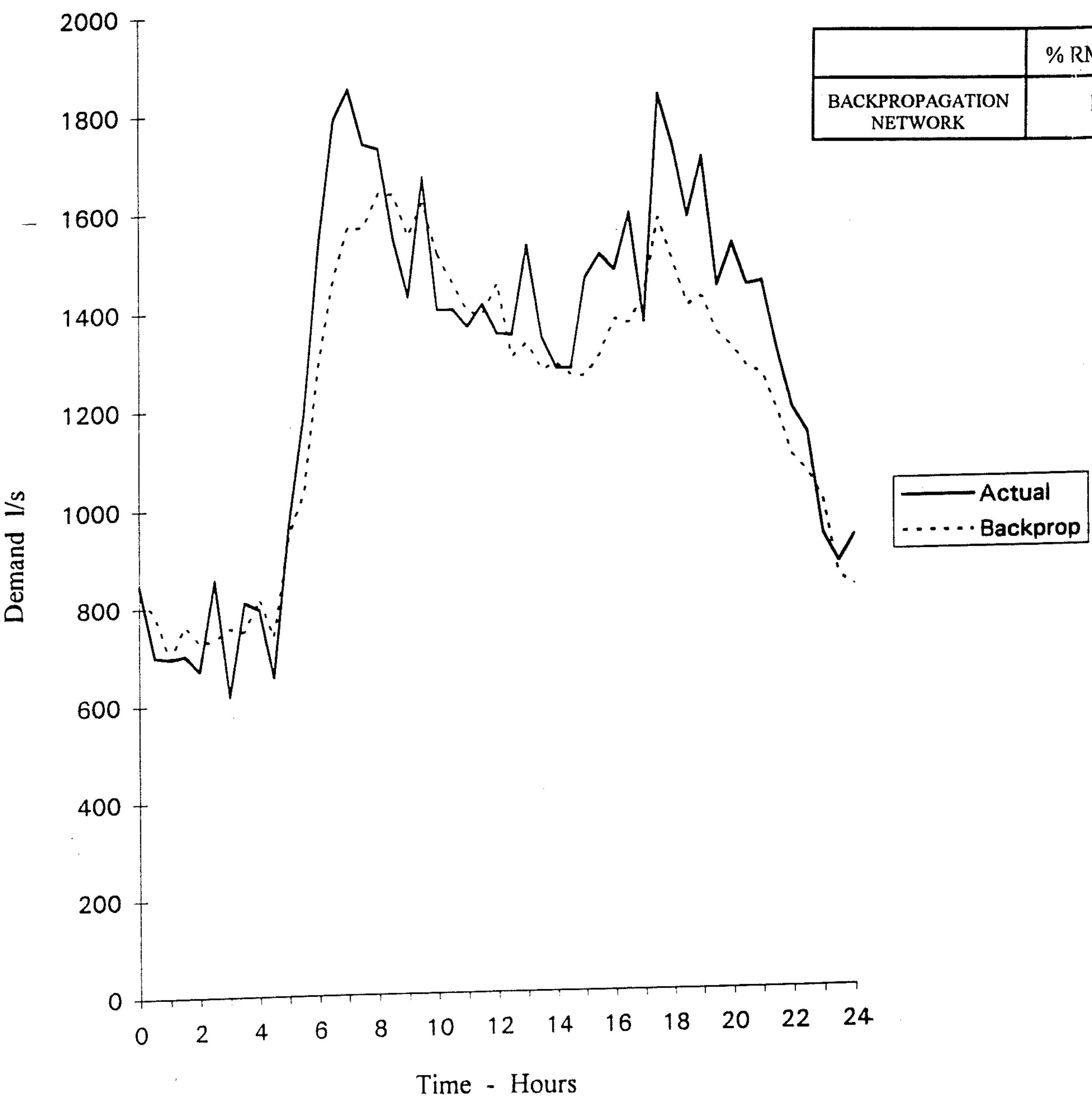
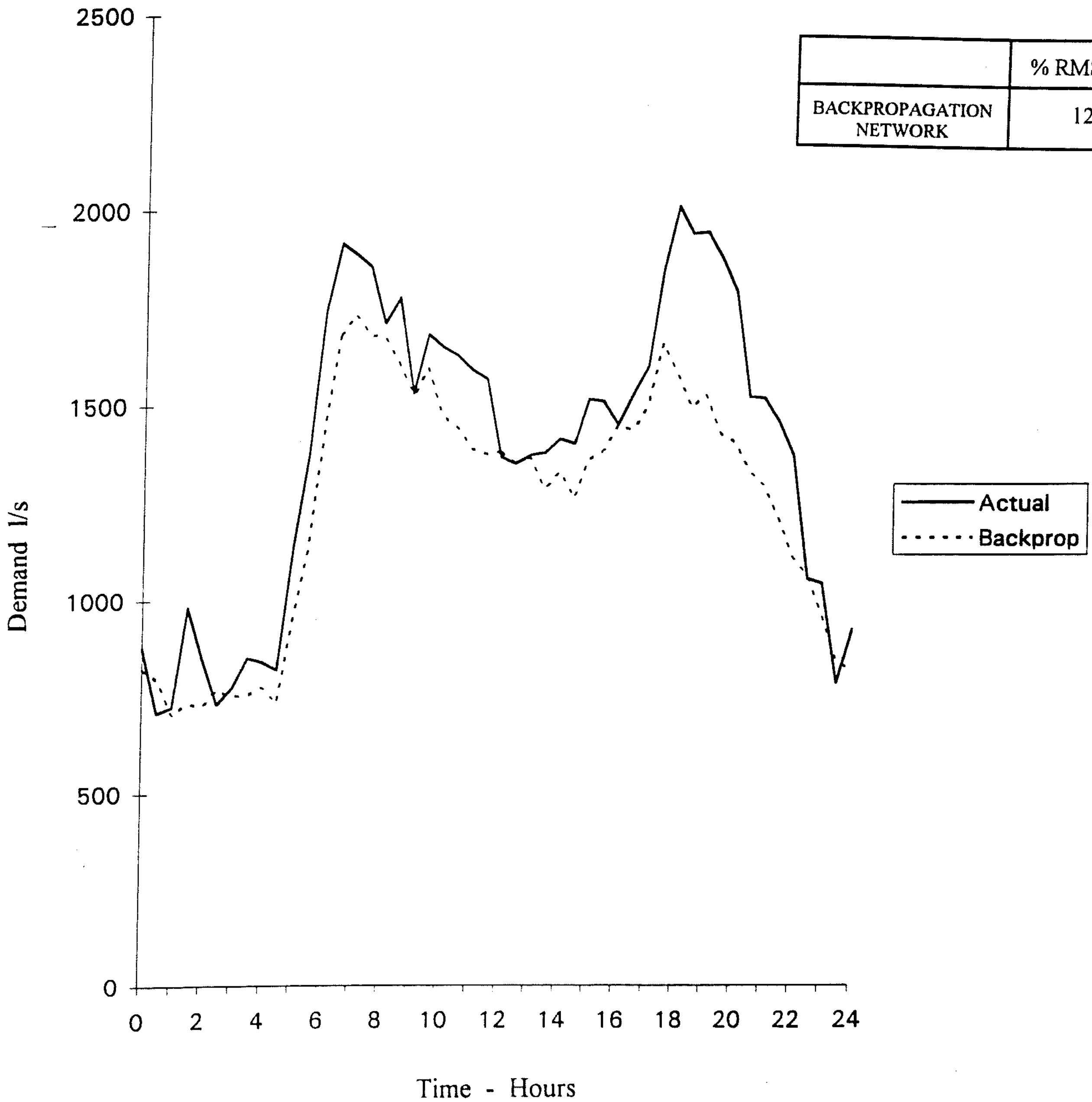


Figure 6.4.7 Actual and Backpropagation Net Prediction 4



	% RMS Error
BACKPROPAGATION NETWORK	12.7

6.5 Counterpropagation Network

6.5.1 Introduction to the Counterpropagation Network

The counterpropagation network was developed by Hecht-Nielsen[68,69] in 1987 and comprises a combination of two previously separate network learning algorithms, the self organising map proposed by Kohonen [89] and the Grossberg outstar[61]. The mapping capabilities of the counterpropagation network are, like the backpropagating network, greater than those achievable by single layer networks. The network operates in such a way as to perform a classification upon the input vector, the result of the classification induces the firing of a particular neuron, which in turn generates a particular output signal.

In terms of the task of incorporating weather information into the generation of water demand forecasts, the operation of the counterpropagation network is highly analogous to the combined operation of the heuristic day type classifier and the linear associator network. It was the prospect of allowing a counterpropagation network to combine the classification and prediction tasks in one network that stimulated the investigation of this particular network type.

6.5.2 Counterpropagation Network Structure

The counterpropagation network in its simplest form consists of three layers, these are termed the input layer, the Kohonen layer and the Grossberg layer. The names of the Kohonen and Grossberg layers are based on the type of learning algorithm associated with these layers. The structure of the network is shown in Figure 6.5.1. The network is fully connected i.e. each neuron in the input layer is connected to each neuron in the Kohonen layer and each neuron in the Kohonen layer is connected to each neuron in the Grossberg layer.

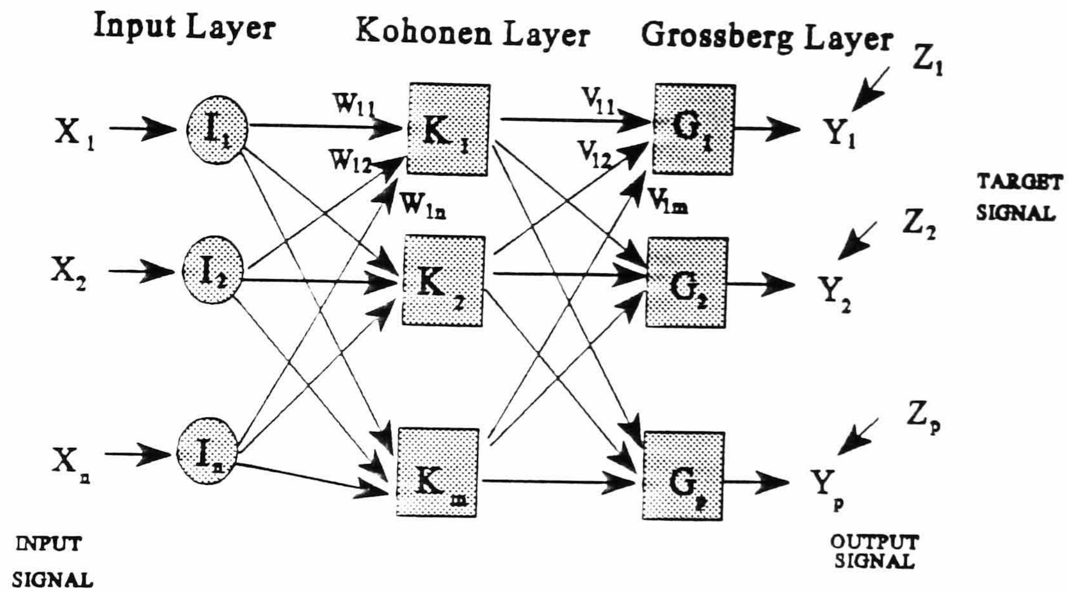


Figure 6.5.1. Structure of a Three Layer Counterpropagation Network.

The above figure shows a counterpropagation network with n input layer neurons, m Kohonen layer neurons and p Grossberg layer neurons. In normal operation (the network has undergone training), a normalised input vector $\mathbf{x}$ of length n is applied to the input layer, the signals are fanned out by the input layer via the $\mathbf{W}$ weighted connections to the Kohonen layer and a transfer function is applied to determine which of the Kohonen neurons is activated. The activated Kohonen neuron then propagates its signal via the $\mathbf{V}$ weighted connections to the Grossberg layer where it is processed by a transfer function to generate the network output $\mathbf{y}'$. The details of the signal propagation and transfer functions of the Kohonen and Grossberg layers are described in the section on network training below.

6.5.3 Training the Network

As with the other neural networks described in this thesis, the counterpropagation network provides a mapping between an applied input vector $\mathbf{x}$ and a desired output vector $\mathbf{y}$, where $\mathbf{y} = f(\mathbf{x})$. During training the network is exposed to examples of the mapping f in the form of training vector pairs $(\mathbf{x}_k, \mathbf{y}_k)$ and the network weights are adjusted so that, in its trained form, the network will generate the correct output vector $\mathbf{y}'_k$ in response to the application of input vector $\mathbf{x}_k$. For a network with n input layer neurons, m Kohonen layer neurons and p Grossberg layer neurons the training process is described below.

Prior to the commencement of training, all the input vectors $\mathbf{x}_k$ in the training set $k = 1 \ldots L$ are normalised to unit length:

$$x_i^N = \frac{x_i}{(x_1^2 + x_2^2 + \ldots + x_n^2)^{\frac{1}{2}}} \qquad \text{For} \quad i = 1 \ldots n \qquad (6.5.3.1)$$

The weights $\mathbf{W}$ associated with the connections between the input layer and the Kohonen layer are given randomised values in the range $(0,1)$. For each neuron in the Kohonen layer the weights associated with the connections arriving at that neuron form a weight vector $\mathbf{w}_j = (w_{1j}, w_{2j}, \ldots, w_{nj})$ for $j = 1 \ldots m$, each of these weight vectors are normalised to unit length.

For each example vector pair $(\mathbf{x}_k, \mathbf{y}_k)$ in the training set the following steps are carried out:

a) Find the Kohonen weight vector $\mathbf{w}_j$ that is closest to the applied input vector $\mathbf{x}_k$

$$\|\mathbf{x}_k - \mathbf{w}_c\| = \min_{j=1}^{m} \|\mathbf{x}_k - \mathbf{w}_j\| \qquad (6.5.3.2)$$

Where c is the Kohonen layer neuron with the closest weight vector to the input vector $\mathbf{x}_k$, and

$$\left\|\mathbf{x}_k - \mathbf{w}_j\right\| = \left|\sum_{i=1}^{n}\left[x_i - w_{ij}\right]^2\right|^{1/2} \tag{6.5.3.3}$$

b) Set the output signal of the winning Kohonen neuron c to the value 1 and set the output signals of all other Kohonen neurons to 0.

c) Update the weight vector $\mathbf{w}_c$ associated with the winning Kohonen neuron c so that it moves closer to the example input vector $\mathbf{x}_k$

$$\mathbf{w}_c^{new} = (1 - \alpha(t))\mathbf{w}_c^{old} + \alpha(t)\mathbf{x}_k \tag{6.5.3.4}$$

Where $\alpha(t)$ is the training rate at time step t. This value typically starts out at a high value i.e.0.8 and gradually decreases to zero as training progresses.

d) Re-normalise the updated weight vector $\mathbf{w}_c$ to unit length.

e) The signals from the Kohonen layer neurons (one of which is 1 and the others 0) are propagated to the Grossberg layer via the weighted connections $\mathbf{V}$. For each neuron in the Grossberg layer the incoming signals form a weight vector $\mathbf{v}_g = (v_{1g}, v_{2g}, \ldots, v_{mg})$ for $g = 1 \ldots p$. The output signal for each Grossberg layer neuron (which are the network outputs) is calculated by:

$$y'_g = \sum_{j=1}^{m} v_{jg}^{old} z_j \qquad \text{For } g = 1 \ldots p \tag{6.5.3.5}$$

Where y'_g is the output of the g^{th} Grossberg neuron, v_{jg} is the weight connecting the j^{th} neuron in the Kohonen layer to the g^{th} Grossberg neuron and z_j is the output signal of the j^{th} neuron in the Kohonen layer.

f) The network generated output vector $\mathbf{y'}_k$ is then compared to the desired output vector $\mathbf{y}_k$ and the weights connecting the Grossberg and Kohonen layers are updated according to the Grossberg learning algorithm, which is implemented for each weight via the equation:

$$v_{jg}^{new} = v_{jg}^{old} + \beta(-v_{jg}^{old} + y_g)z_j \quad \text{For } j = 1 \ldots m \text{ and } g = 1 \ldots p \qquad (6.5.3.6)$$

Where β is the learning rate for the Grossberg layer (a small positive constant) and y_g is the g^{th} element of the current desired output vector.

The above steps are repeatedly carried out for each vector pair in the training set of example input/output vectors. This continues until the errors between the network generated outputs and the desired outputs are minimised. The training process can be summarised as follows:- the input vectors to the network are drawn from a bounded subset of all possible input vectors, the input neurons distribute the applied input vector values to the Kohonen layer neurons and a competition is held between these neurons to determine which of them possesses a weight vector that most closely matches the input vector. The winning neuron of this competition has its output set to 1 and its weight vector updated so as to move it closer to the applied input vector. All other Kohonen layer neurons have their output signals set to 0. The winning Kohonen neuron output signal is propagated to the Grossberg layer, where the Grossberg transfer function serves to select the weights associated with the input to each Grossberg neuron from the winning Kohonen neuron. The values of these selected weights are then output by the Grossberg neurons to form the network output vector. The Grossberg learning law modifies only the weights associated with the connections to the winning Kohonen neuron. As training progresses, so the training rate coefficient for the Kohonen layer decreases towards zero and as a result the Kohonen layer weights stabilise. Hence, as the training set of example input/output vectors are repeatedly applied, the same Kohonen neurons are always triggered by the same input vectors. Given the stabilisation of the Kohonen weights, the training of the Grossberg layer leads to its weights learning the average of the ideal output values associated with the input vector that always causes a particular Kohonen neuron to fire.

Once trained, the counterpropagation network acts as an adaptive lookup table as represented in Figure 6.5.2. An input vector $\mathbf{x}$ is applied to the network and is compared with all of the Kohonen layer neuron weight vectors $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m$ to find the vector $\mathbf{w}_c$ that is the closest match to $\mathbf{x}$. The table then emits the associated Grossberg weight vector $\mathbf{v}_c$.

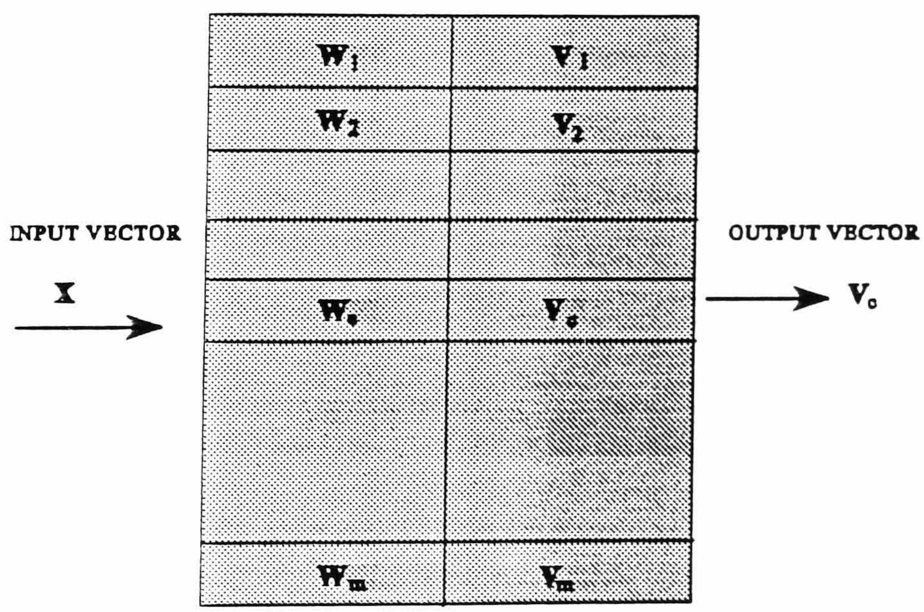


Figure 6.5.2. The Counterpropagation Network Acts as an Adaptive Lookup Table.

6.6 A Counterpropagation Application to Demand Forecasting

6.6.1 Structure of the Network

A counterpropagation network was designed for the generation of demand forecasts based on the input of meteorological variables. As stated in the introduction to the counterpropagation network, the principle design aim was to investigate the possibility of using the network to perform both the weather based day type classification and the consequent generation of the demand prediction. In order to achieve this, a three layer network was constructed, the network consisted of 5 input layer neurons, 4 Kohonen layer neurons and 48 Grossberg layer neurons. Initial tests had been conducted using varying numbers of Kohonen layer neurons (4, 6, 10) and the best results were achieved with the 4 neuron network. Significant errors occurred with the 6 and 10 neuron configurations and examination of the network weights indicated that these were caused by the non triggering of some of the Kohonen neurons during the training process. If a given Kohonen neuron is not triggered by any of the examples in the training set i.e. its weight vector is not selected as being closest to any of the example input vectors, then the Grossberg layer weights that connect to the Kohonen neuron in question are not altered from their initialised values. Hence, if during the normal operation of the trained network, an input vector is applied that does trigger the previously inactivated Kohonen neuron, then the resultant network output will simply be the initialised values of the Grossberg layer weights. This problem was least frequently encountered during the testing of the 4 Kohonen neuron network, it is this configuration which is described below.

The input vector to the network was composed of five elements as for the inputs to the backpropagation network, the maximum daily temperature in degrees Celsius, the total daily sunshine hours, the total daily rainfall in mm, an antecedent dry day indicator and an antecedent hot day indicator. The network output vector corresponds to the 48 half hourly data points of a 24 hour demand prediction.

6.6.2 Training the Network

As for the backpropagation network, the training data for the counterpropagation network is compiled from the raw consumption and meteorological data using a FORTRAN subroutine. Twenty eight days of the most recent consumption data and corresponding meteorological data are each compiled into a separate array. The rows of data in the meteorological storage array form the example input vectors for the network training and the rows of data in the consumption array form the desired output vectors.

The pairs of example vectors are extracted from the storage arrays and applied to the network repeatedly until the errors between the network generated output vectors and the desired output vectors fall below a specified threshold. Training times were of the order of 2 minutes, however, the time taken to successfully train the network was highly dependent on the initial values chosen for the Kohonen layer weights. Normally, initialisation of network weights is achieved by assigning small random values to the weights. In the case of Kohonen layer weights, the initial weight values should be normalised, this is because input vectors are normalised and the trained Kohonen weight values need to match these input vectors. However, purely randomising the Kohonen layer weights, can result in serious problems in the training of the network. This is because the example input vectors submitted to the network are not evenly distributed throughout the hypersphere of all possible input vector values and instead they are 'clumped' in a relatively small portion of such a hypersphere, therefore most randomised weight vectors will be so far from the example vector values that they will never be the best match . Hence, the Kohonen neurons with which these weight vectors are associated will never be triggered and therefore they do not contribute to the function of the network. Conversely, those few randomised Kohonen weight vectors that are in the same hypersphere region as the example vectors, may be too few in number to provide the desired separation of the input vector categories. In order to achieve the desired network performance, it can be seen that a high density of initial weight vectors is required in the vicinity of the applied input vectors.

It is possible to partially overcome the problem outlined above by randomly adding noise to the input vectors such that they span a large selection of the possible input vector values. The outlying Kohonen weight vectors are then 'captured' by these noisy input vectors and as the amount of noise is decreased during training, so the weights are brought to the region containing of the majority of the true input vector values. Alternatively, training can start with randomised Kohonen weights but <u>all</u> of the weight vectors are altered following the application of an input vector, instead of just the weights of the triggered Kohonen neuron. This results in the weights moving to the region of highest input vector density. As training progresses, weight updating is limited to only those Kohonen units whose weight vectors are nearest to the winning neuron and this radius of weight adjustment is gradually reduced until eventually only the winning neuron weights are altered.

The significant disadvantages of the above methods and others that have been proposed to solve this problem, are that they do not guarantee the desired separation of the input vectors and also that they result in greatly increased training times. For the purposes of minimising the training times and achieving successful separation of input vectors for the counterpropagation network described here, the Kohonen weights were initialised to normalised values known to be in the same region as the example input vectors.

6.6.3 Results

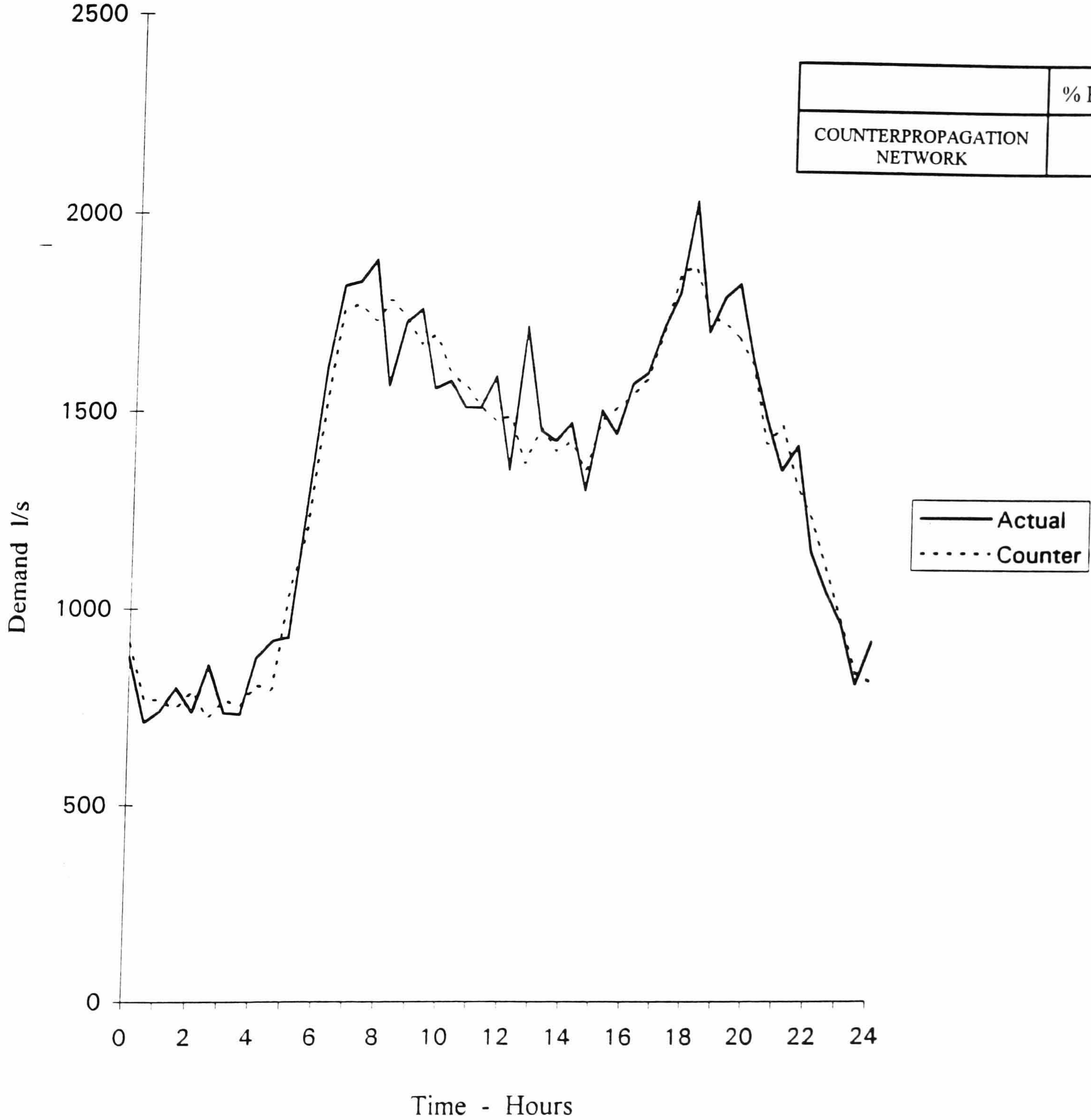
The counterpropagation network was tested on the same consumption and meteorological data as the backpropagating and linear associator networks. However, as the results shown below indicate, the network was significantly less successful in the generation of accurate demand predictions than either of the two other networks. The results below are from a two week period between May 15 and May 28. Generally, the daily percentage errors are greater than the results from either of the two other networks and on Saturday 24 and Monday 23 the network generates nonsensical predictions. These highly erroneous predictions are the result of the input vector causing the triggering of a Kohonen neuron that was not triggered during training and hence the Grossberg connection weights leading from that neuron were not altered from their initialised values. The network output is therefore composed of the unadjusted Grossberg weight values and gives rise to the observed errors.

Figures 6.6.1 to 6.6.4 show the prediction profiles generated by the counterpropagation network in comparison to the actual consumption profile. In figures 6.6.1 and 6.6.2 the degree of prediction accuracy achieved is acceptable (although poorer than that achieved by the linear associator for the same days), however, as figures 6.6.3 and 6.6.4 demonstrate, serious large scale errors between actual and predicted values can occur. Table 6.6.5 shows the average daily percentage errors for the counterpropagation network predictions over 14 days and compares them to the results for the same period for the linear associator and ARIMA applications.

Table 6.6.5 Results For 14 Days Spring 1990			
	ARIMA	Linear Associator	Counterpropagation Network
Average Daily Error	8.8	7.6	12.6
No. of Days Error >15%	1	0	6
No. of Days Error >10%	2	1	8
No. of Days Error >8%	5	2	10

Table 6.6.5 Relative accuracy of the counterpropagation network.

Figure 6.6.1 Actual and Counterpropagation Net Prediction Profile 1



	% RMS Error
COUNTERPROPAGATION NETWORK	5.1

Figure 6.6.2 Actual and Counterpropagation Net Prediction Profile 2

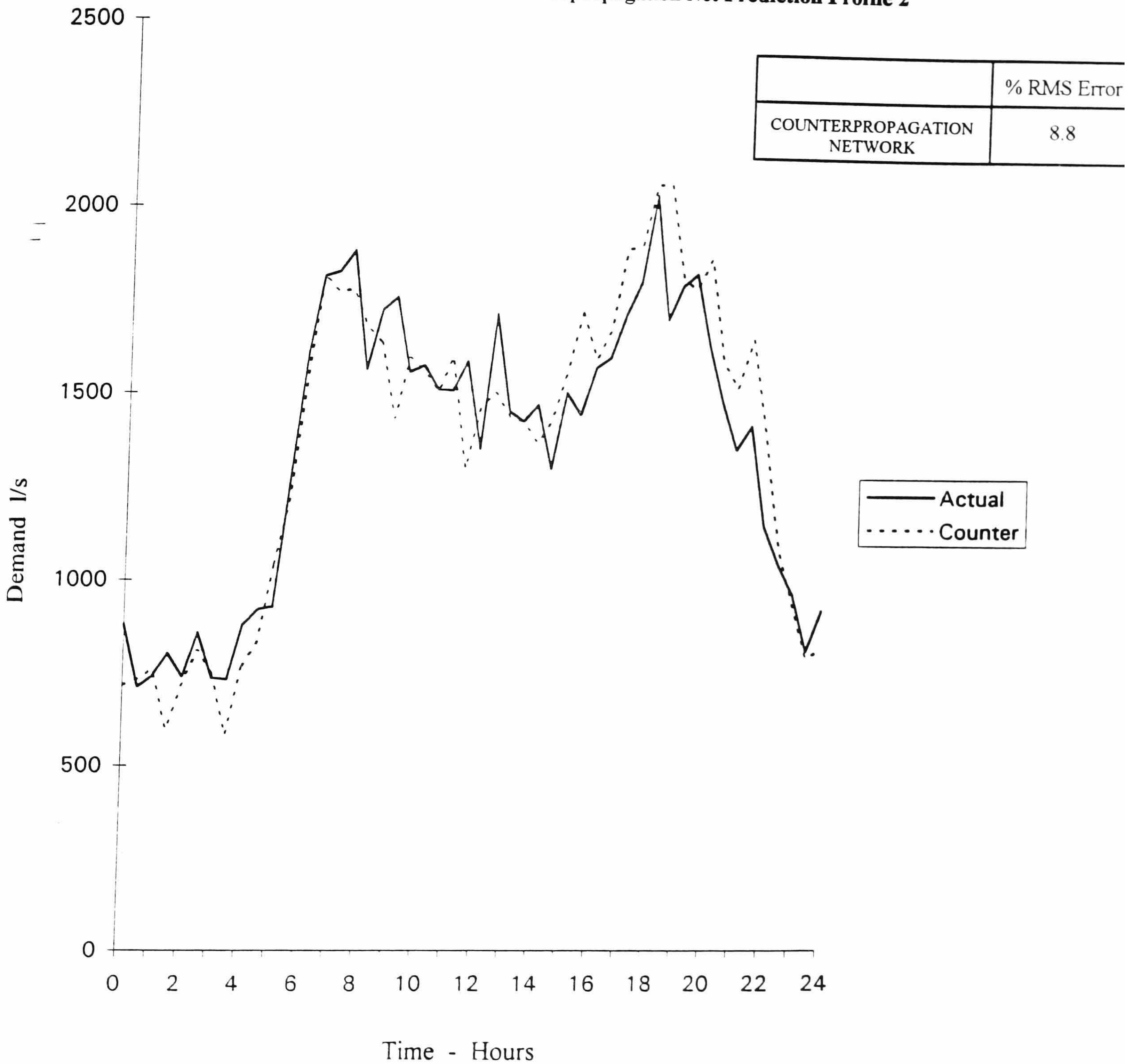
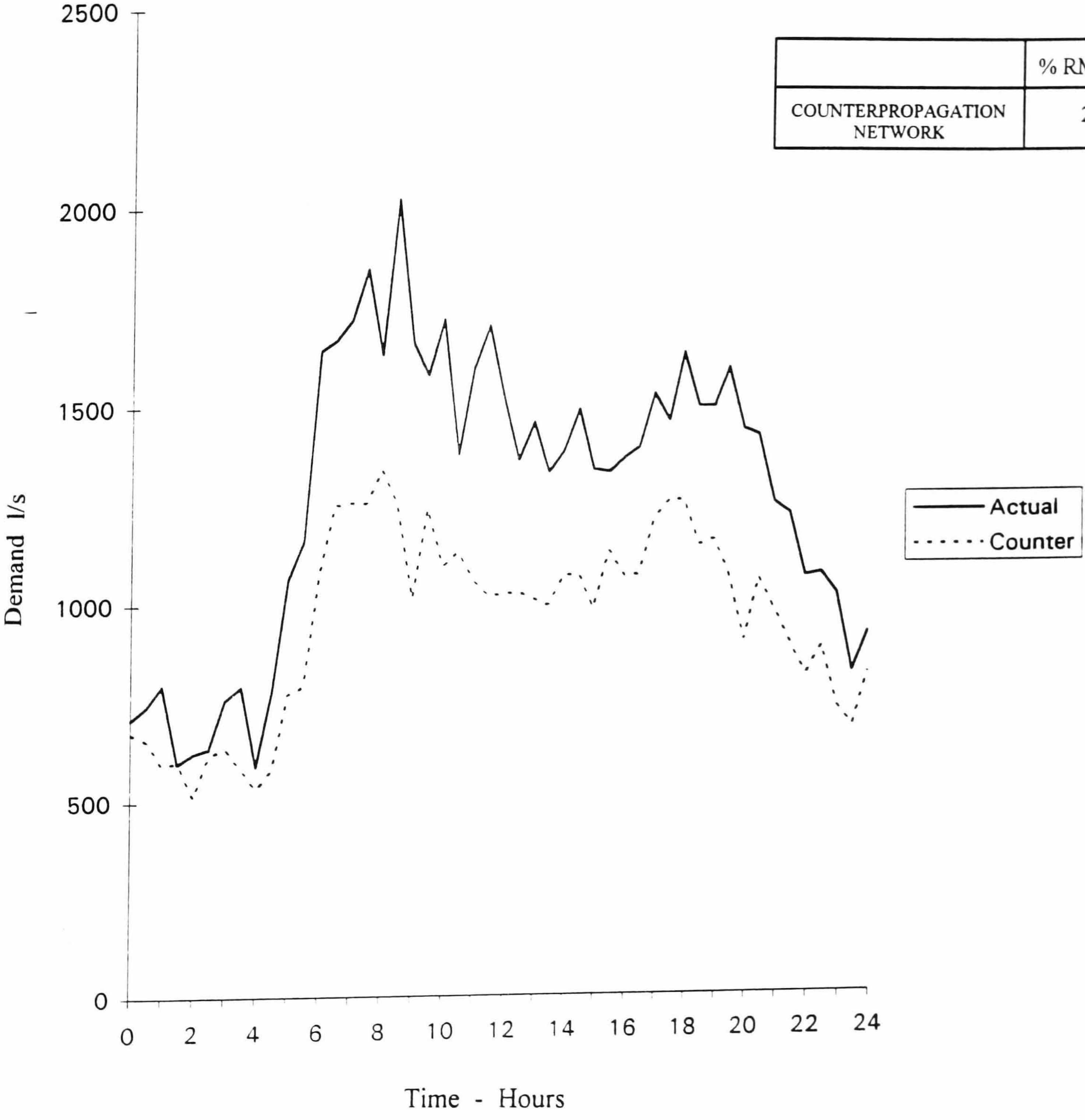
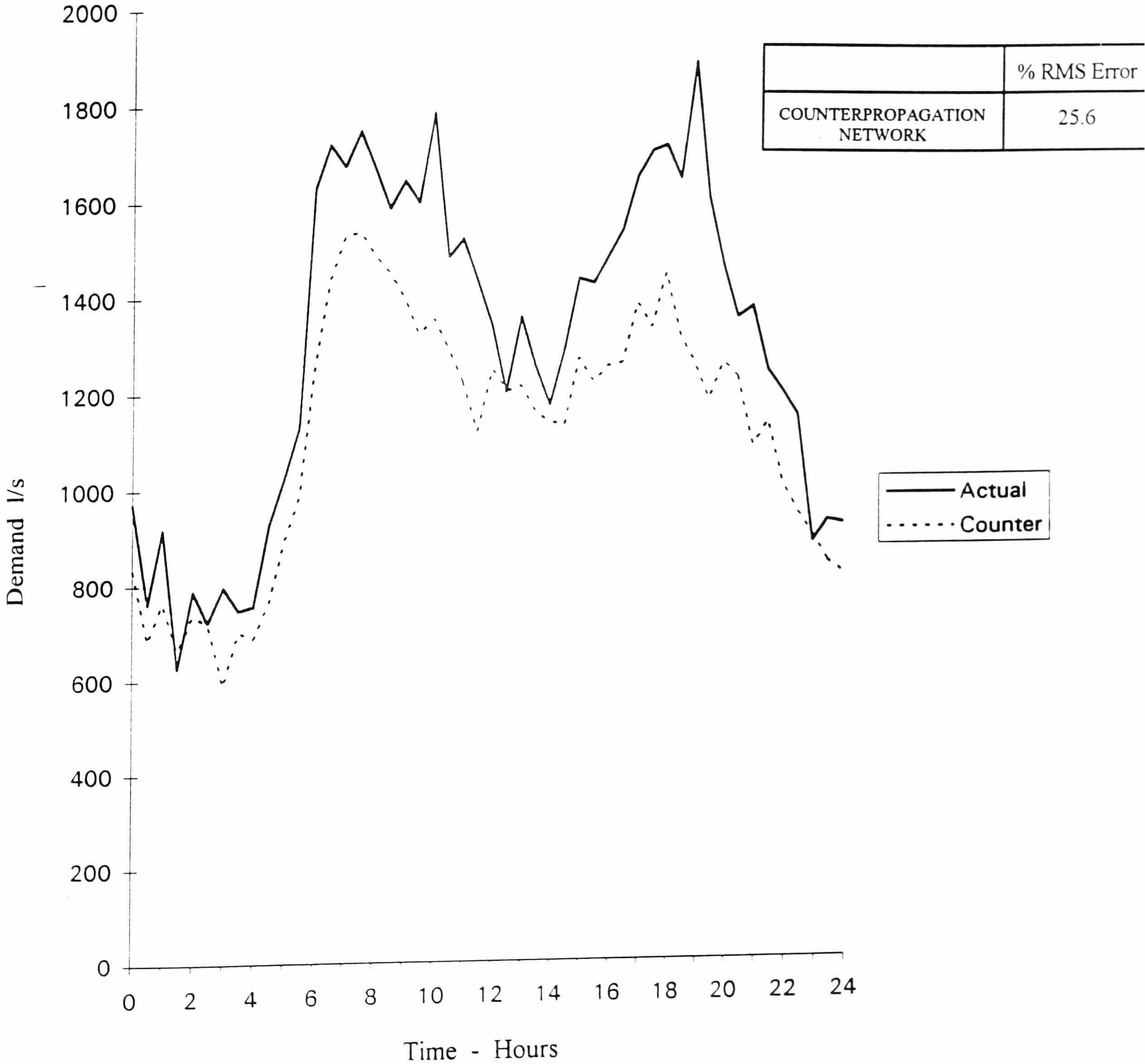


Figure 6.6.3 Actual and Counterpropagation Net Prediction Profile 3



	% RMS Error
COUNTERPROPAGATION NETWORK	27.3

Figure 6.6.4 Actual and Counterpropagation Net Prediction Profile 4



6.7 Discussion

The motivation behind the development of the backpropagation and counterpropagation networks, was to determine the relative advantages and disadvantages of using neural networks to directly map the relationship between particular meteorological variables and resultant levels of water demand. The results could then be assessed in the light of the degree of success achieved by the combination of the heuristic day type classification system and the linear associator network.

The backpropagation network daily percentage error results were generally comparable to those generated by the ARIMA time series prediction algorithm but were inferior to those attained by the day type classifier/linear associator. The backpropagation network predictions were closest to the actual demand values when the meteorological conditions were average for the time of year. However, when more extreme weather conditions were encountered, then the prediction accuracy from the network decreased significantly, this is shown in figures 6.7.1 and 6.7.2.

The network does not appear to be successfully interpolating between the specific training examples it was exposed to during the training process. Although examples of extreme weather conditions were present in the training set, unless the input vector generated from the prediction day meteorological variables is an exact match to one of the training example input vectors, the network does not generate the correct extreme weather demand profile. It may be that this problem is the result of the particular network architecture or of the composition of the network input vector and further refinement would be able to expand on the encouraging results generated for the meteorologically average prediction days.

In the application of the counterpropagation network to the demand forecasting task, it is the failure of the classification function that the network is designed to perform that is the cause of the significant errors. The self organisation inherent in the Kohonen layer weight updating algorithm does not appear to provide the necessary adaptability to match the performance of the heuristic classifier used in conjunction with the linear associator network. Because it is not feasible to determine exactly how the network is making the classifications, there is an uncertainty of outcome that is not present in the rule based approach to day type classification. It would appear from the results that it is desirable for the system designer/operator to have a cognitive control over the criteria used to classify meteorological day types, and this is best achieved in via an interactive rule based application.

Figure 6.7.1 Actual and Backpropagation Net Prediction Showing Errors Introduced by Extreme Weather

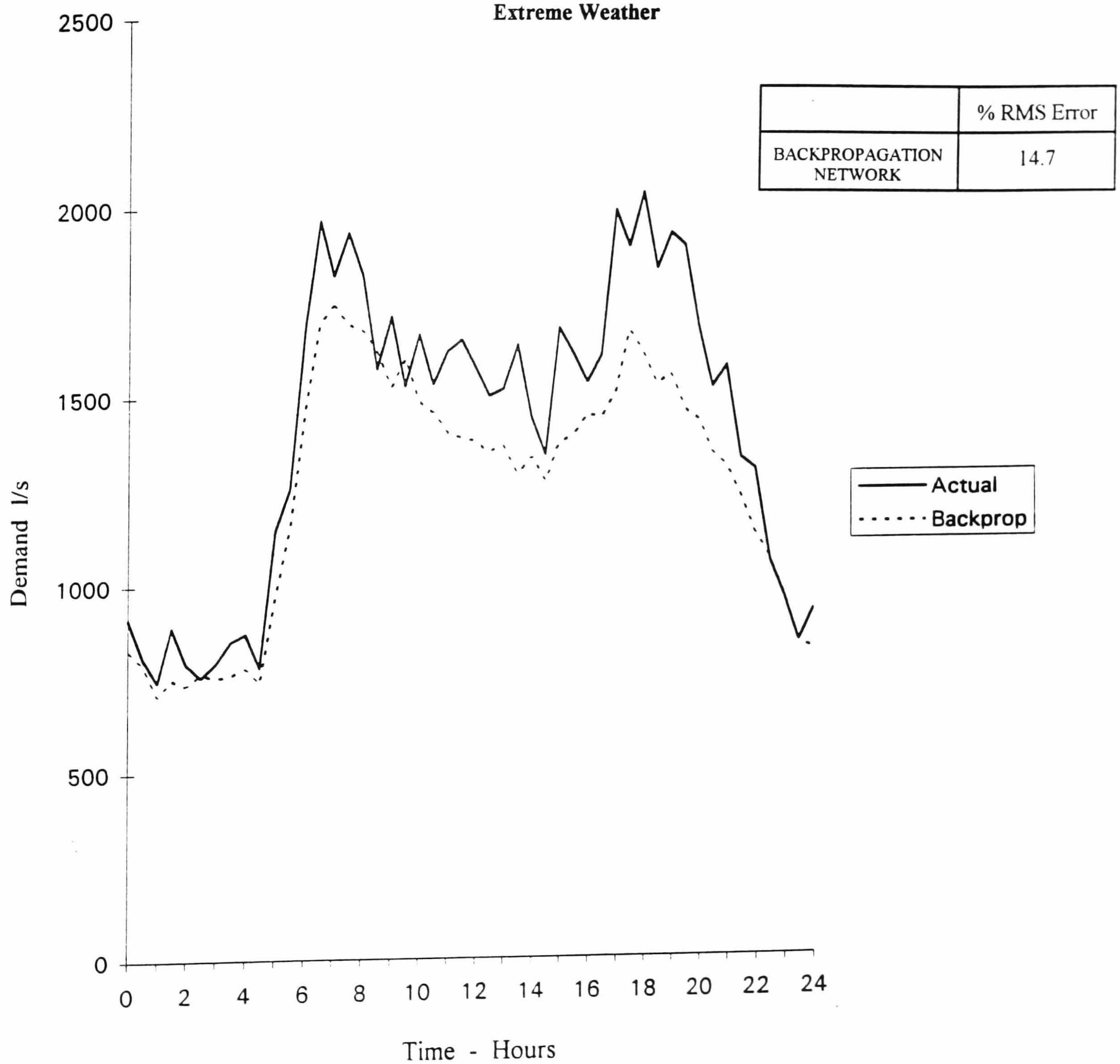
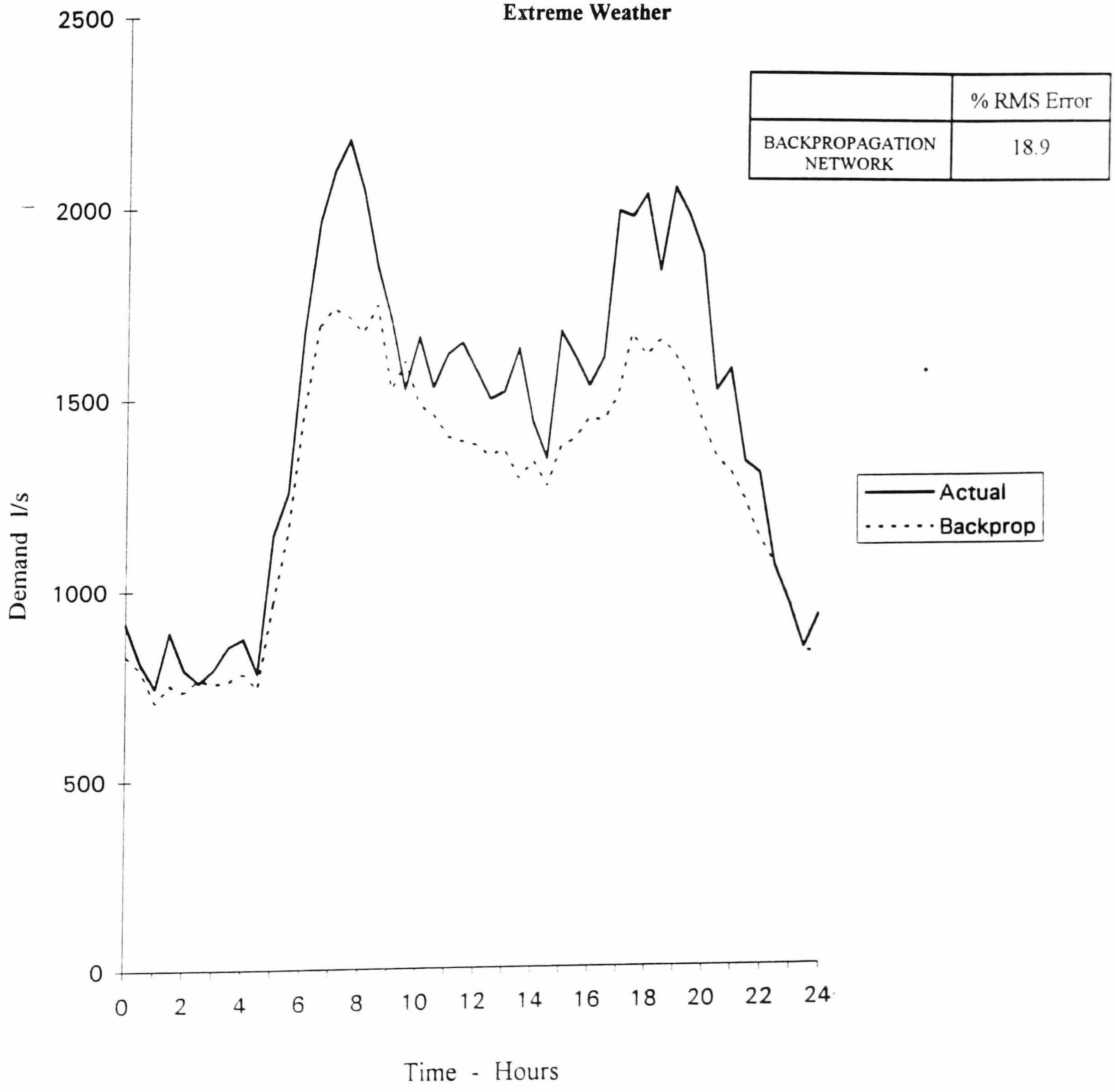


Figure 6.7.2 Actual and Backpropagation Net Prediction Showing Errors Introduced by Extreme Weather



CHAPTER 7

CONCLUSIONS

7.1 Introduction

This chapter sets out the conclusions and insights that have been forthcoming as a result of the work detailed in this thesis. This includes an understanding of the importance of accurate demand forecasts, the problems that were encountered during the research, the reasoning behind the design and implementation of the combined demand forecaster, the advantages the system offers and its relevance to the current and likely future state of the water supply industry.

7.2 Problems, Solutions and Conclusions

The water industry in England and Wales has undergone and is still undergoing a period of significant change. This change has been driven by two main influences, one being the privatisation and subsequent increased emphasis on efficiency in all areas of water operation and the second being the impact of technological advances and investment in network monitoring hardware and software. These two factors combined with a steady increase in the overall demand for water, has meant that there is an increased need within the water companies to control their networks more efficiently. Furthermore, the technology is now available for that increased level of control to be achieved. The situation described above highlights the importance of accurate short term demand forecasting in that it can provide the means by which water supply companies can optimise their pumping operations in order to meet the required level of consumption.

7.2.1 Analysis of the Problem

An essential initial task when approaching the problem of generating accurate demand forecasts is the analysis of the significant components that contribute to the total level of water consumption. In this work, based on a combination of experience within the water industry, examination of example demand profiles and a review of past literature, divisions were made into domestic consumption, industrial/commercial usage, agricultural irrigation and leakage. Proportions of each category vary from area to area and each component is affected in different ways by the determining factors that influence the shape of the demand profile. It was an important consideration in the design of the combined demand forecaster that the system should not be area specific i.e. that it should be flexible enough to allow it to be used in areas with varying demographic composition.

Following the identification of the components that contribute to the total consumption, it was necessary to analyse the factors that caused the observed variations in the shape of the diurnal demand profile. The most significant single factor that determines the pattern of water usage over a given period of time is the cyclic nature of domestic and industrial demand. This results in the pronounced diurnal and weekly repetition of consumption patterns. The weekly cycle highlights the significant differences in the patterns of social behaviour of the majority of the population at weekends as compared to during the week. It was apparent from studies of consumption patterns from a number of different geographic areas, that the presence of stable and regular cycles was highly dependent on the size of the areas concerned. The lower the population within an area, the less pronounced the diurnal and weekly patterns. It is therefore apparent that in order to mask the effects of individual consumptions that do not conform to the 'norm', there is a minimum number of consumers required to be present in the area from which demand data is sampled. Unfortunately, even with a suitably large sample area, examination of a typical record of past demand data reveals there are numerous factors in operation that significantly distort the cyclic pattern of consumption.

Valuable information and ideas that influenced the development of the combined demand forecaster were gained from the examination of previous work done in the fields of both short term electrical load forecasting and water demand forecasting. The basic realisation was that very few of the methods described had been shown to provide consistently accurate forecasts over a wide range of conditions, and that this was largely due to their concentration on a single or limited number of aspects of the many faceted problem of demand forecasting. Those methods that concentrated on extracting the maximum amount of information relating to the cyclic processes present in a time series of demand data were compromised by the presence of non cyclic events or influences. Those methods that concentrated on the causal relationships between meteorological variables and the level of demand generally were not able to track the rapid variations in these relationships that are evident at particular times of year. It was the realisation that the relationship between weather conditions and their influence on water demand was subject to such significant change that led to the investigation into the use of the day type classifier and linear associator neural network.

A classification of the non cyclic factors that cause distortions to the level of demand has been proposed in this thesis and is based on the examination of the available data, the experience gained working in a supply network control room and the consideration of the methods developed in this work to account for each type of effect. The classification comprises three categories, calendar related effects, network related effects and weather related effects. The calendar related effects being those related to a particular date or time of year, the network related effects are those that are linked to changes or events within the network itself and the weather effects are linked to the prevailing meteorological conditions. The aim of the combined demand forecaster was to provide a system that could model the regular cyclic variations and also have the ability to account for those effects that distorted the demand data.

7.2.2 The Use of Rules

The results generated by a prototype forecasting system that incorporated the ARIMA algorithm and a small rule base constructed in FORTRAN, showed that the effects of calendar related events could be successfully accounted for by a rule oriented approach. However, as the number of rules became larger, the FORTRAN rule base became less efficient and harder to structure correctly. By transferring the rule base to the POPLOG AI environment, which incorporates the PROLOG and POP11 programming languages with their advanced inference engine and pattern matching capabilities, a much more effective means of rule base construction and manipulation was achieved.

In order that the system was easy for a non computing expert operator to use, a menu driven interface was developed which allowed rule entry, selection, editing and deletion. The quality of the user interface would be highly significant in determining the acceptability of a demand forecasting system in the control room environment, it was therefore very important that the menu driven system described in Chapter 4 and the GKS graphical interface described in Appendix A were developed in parallel with the forecast generating rule base and neural networks.

The results generated by combining the POP11 rule base and the ARIMA algorithm showed that it was possible to remove a significant proportion of the large prediction errors that occurred when using the ARIMA algorithm on its own. The use of rules appears to be well suited to those effects such as calendar and network effects which are relatively stable i.e. their occurrence can be foreseen with a degree of certainty and their influence upon the demand is more or less constant. The flexibility of the rule based system is highly important in that it allows events that become apparent only during the ongoing operation of the forecasting system to be incorporated as and when they appear. In a similar way, the rule base flexibility prevents the forecasting system from being restricted in application to the geographic area for which it was originally constructed i.e. area specific effects can be incorporated by the construction of area specific rules.

As was shown in the results of Chapter 3, the ARIMA algorithm fails to provide accurate predictions when the weather conditions are varying significantly. Problems were also encountered in attempting to account for these weather related influences by the construction of rules. Rules do not provide a suitable solution in situations where the knowledge they are trying to represent is highly volatile. It was found that it was not possible to derive rules that would remain valid over any period of time without constant updating of the rule weights.

7.2.3 Neural Networks

The aim of the investigations into the use of neural networks was to develop a system that could track the variations in the relationship between the values of selected significant meteorological variables and the resultant level of demand. The classification of days into one of a number of predefined types based on the prevailing meteorological conditions was found to have a number of advantages, it simplified the network structure, it made the prediction of the meteorological conditions of the forecast day easier and also allowed for a degree of variation that avoided the problem of the network trying to learn the inconsistent relationship between specific meteorological values and resultant demand levels. Over the period of the forecast the day type classification is capable of absorbing both minor errors in the prediction of weather variables and the variations in the relationship between such weather variables and the resultant demand levels. It was found that the use of the day type classifier in pre-ordering the training data used by the linear associator network was of equal importance to the structure of the network itself in achieving the level of accuracy observed.

The results in Chapter 5 proved that the day type classifier and the linear associator network could provide accurate predictions over the majority of meteorological situations. This included situations where the level of demand varied dramatically from one day to the next such as the first wet day after a dry spell, the accuracy of the neural network forecast remained good as long as there were examples of wet days present in the training set. This contrasted with the results from the ARIMA algorithm, the same situation of a wet day following a dry spell consistently led to a large over estimate in the predicted demand for the wet day. Another factor that was found to have a significant impact upon the resulting accuracy of the neural network predictions was the routine that was developed to ensure that the data from the most recent example days was submitted more frequently to the network during the training phase than the data from further back in the data set. By adjusting this frequency of re-submission of the most recent data, it was possible to fine tune the bias so that the system was reactive to changes taking place in the four or five days prior to the prediction day, but not too sensitive as to be severely distorted by an unusual profile that may occur in this recent data.

There were situations when the accuracy of the day type classifier and linear associator were compromised, these generally involved either extreme weather conditions or situations where there was a lack of available past data. In these cases, such as the first hot weekend of the year where no example data exists in the training set, the neural network could not be expected to provide accurate predictions. As shown in Chapter 4 Section 4.5.3 the solution was found by adding appropriate rules to the POP11 rule base to provide the required knowledge to adjust the prediction.

The linear associator network is, as its name suggests, only capable of modelling linear relationships, the incorporation of the day type classification module into the forecasting system had the effect of simplifying the relationship between weather and demand thereby allowing the linear associator to successfully provide the required mapping. It was possible that this arrangement was masking information that could otherwise be extracted from the weather and demand data by networks capable of modelling non linear relationships. In order to test this hypothesis, the more advanced networks described in Chapter 6 were constructed. The three layer backpropagating network was tested using data covering the same period as the linear associator network but without being subdivided into weekday and weekend data. The results were of a comparable accuracy to the ARIMA results but less accurate than those produced by the day type classifier and linear associator net. The network was found to be behaving in a similar manner to the ARIMA algorithm in that it could track the variations in demand due to weather conditions provided these conditions were not changing rapidly. When rapid meteorological changes did occur, the variations in the relationship between weather conditions and demand levels were such that the network had problems stabilising. It was also noted that the prediction accuracy of the backpropagation network was decreased when predicting for a day with extreme weather conditions i.e. very hot and sunny. The cause of this problem is related to the lack of examples of such extreme conditions in the training set, some method would be required of ensuring the training set contained a complete range of example meteorological conditions and also of ensuring that such a range of examples remained valid over varying prediction dates.

The counterpropagation network was developed in order to investigate the possibility that the network would be able to carry out the same day type classification task that is conducted in the linear associator application by a set of rules. The network is composed of three layers, an input layer, a Kohonen layer and a Grossberg layer with each layer undergoing training in separate phases. A number of different network configurations with varying numbers of Kohonen neurons were tested, this amounted to testing how successful the network was at classifying the input vectors into mutually distinguishable sub-groups. The best results in terms of consistent triggering of Kohonen layer neurons, was found to be with 4 neurons in the Kohonen layer. It is interesting to compare this with the four day type categories as used with the linear associator network, this may be a function of the particular data set used but it implies that it may be unproductive to attempt to identify and classify days into a large number of types based on their weather conditions.

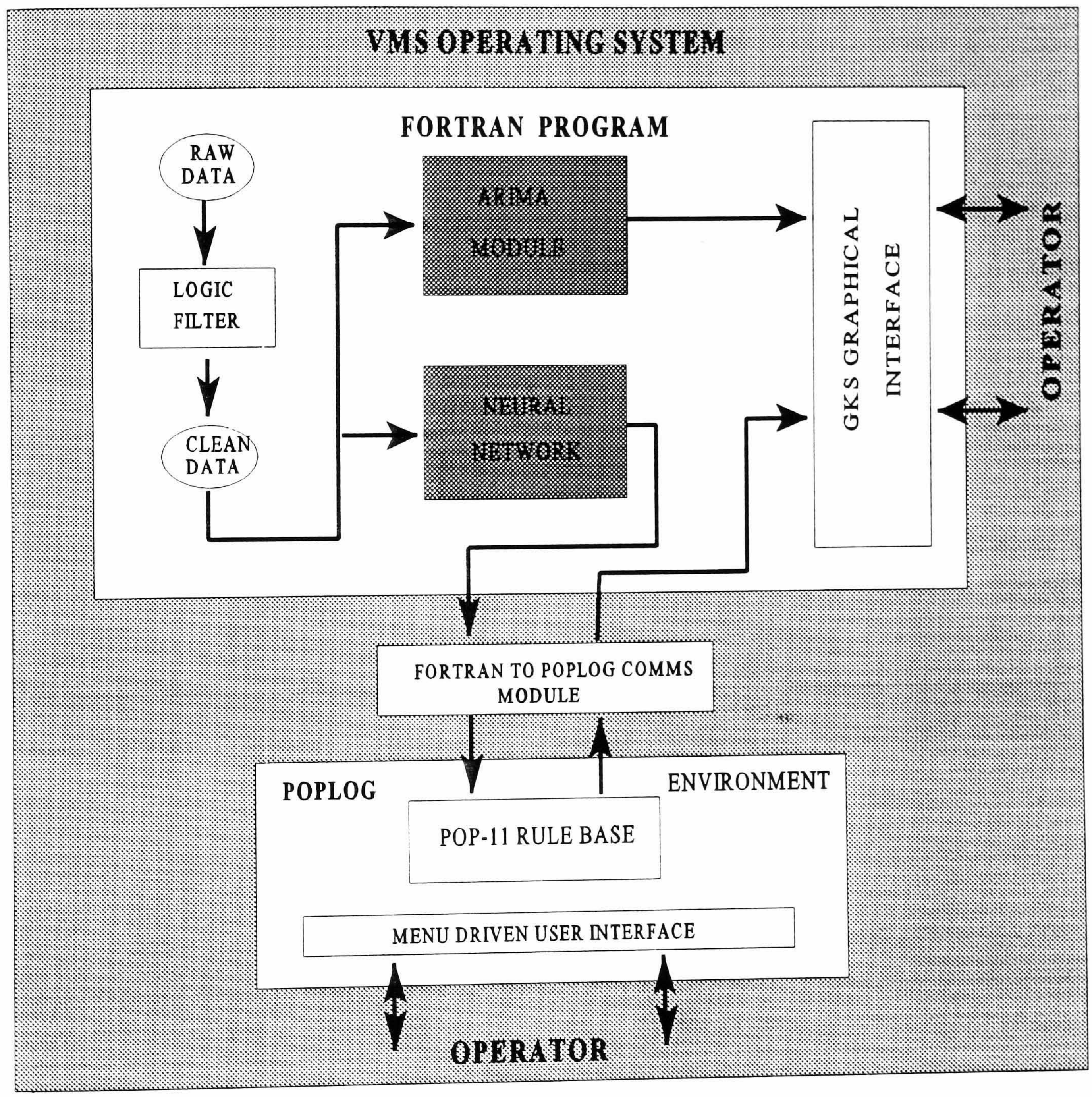
The results generated by the counterpropagation network were acceptable for only 4 days out of the training set of 14 days, gross prediction errors occurred on the remaining days. Investigation as to the cause of these gross errors revealed that they originated from the triggering of neurons of the Kohonen layer that had not been triggered during the training process, which indicates that the input vectors are not sufficiently separable for a consistent classification to be carried out by a purely automatic process. The separation into the day type categories carried out for use with the linear associator network was achieved by a set of rules and it appears that the knowledge held in those rules, such as 'if it is sunny and hot then more water is likely to be consumed than if it is equally hot but cloudy', is required in order to achieve a consistent classification. The counterpropagation network has no access to such heuristic knowledge and purely on the basis of the values of the meteorological variables, cannot achieve the desired classification with the required consistency.

7.2.4 The Combined Forecaster System

This thesis showed that the combined forecasting system comprising the linear associator, the day type classifier, the POP11 rule base and the GKS user interface was the most successful means of providing consistent forecasts over a range of conditions. The neural predictor combined with the day type classifier provide a means of accounting for the fluctuations in demand caused by the effect of meteorological conditions. The POP-11 rule base provides a mechanism by which the heuristic knowledge about events that are known to influence the shape of the consumption profile can be captured. This knowledge can be added to and updated by the operator via the menu driven interface. The GKS graphical user interface (described in Appendix A) provides an accessible means of viewing the results generated by the system and of visually assessing the system performance.

Although the component approach leads to a certain degree of program complexity (a schematic of the demand forecaster structure is shown in Figure 7.2.1), the result is a system that is straightforward for an operator to use and which can provide forecasts of consistent accuracy. This in turn means that the likelihood is increased of such a system being incorporated into a control room environment and utilised to provide input to pump scheduling or network analysis packages.

Figure 7.2.1 Schematic of the Operation of the Demand Forecaster.



7.3 Future Developments Relating to Demand Forecasting

The forecasting system described in this thesis represents a significant step forward in the use of AI techniques such as neural networks and knowledge based systems for the provision of accurate demand predictions. It successfully allows the incorporation into a 24 hour prediction effects that are very difficult or impossible to model by traditional mathematical means. The system is also flexible enough to cope with the different demand influencing factors that may be encountered in different geographic areas. However, there is much more research to be done to explore the possibilities for extending the use of heuristic knowledge in order to improve forecasting not only in relation to water networks but also in the related fields of electrical load prediction and gas consumption. It should be noted that any such research is critically dependant on the availability of sufficient, good quality historic data. It is the lack of such data within the water industry in the UK that is currently is the main factor limiting the ability of forecasting systems to achieve improved accuracy. It is only by building up a database of coherent past data that the required number of examples can be made available of the sort of events that are suitable for representation by heuristic means. This avoids the undesirable situation of basing a prediction on a single past occurrence of a similar day. With the increase in the use of telemetry systems to monitor water supply networks and the increasing statutory requirements upon the water utilities to gather and process data on consumptions, leakage levels etc. it is hoped that the data required to extend the investigations into predicting demands, both long and short term, will become more easily and widely available.

REFERENCES AND BIBLIOGRAPHY

1. ABOU-HUSSIEN, M.S., KANDIL, M.S., TANTAWY, M.A., "An Accurate Model for Short Term Load Forecasting", IEEE Trans. on PAS-100, No.9, Vol 100 Sept 1981.

2. AGTHE, D.E., BILLINGS, R.B., " Dynamic Models of Residential Water Demand", Water Resources Research, No. 16, 1980.

3. ALEKSANDER, I., MORTON, H., " An Introduction to Neural Computing ", Chapman and Hall, 1990.

4. ANDERSON, J.A., " A Simple Neural Network Generating an Interactive Memory", Math. Biosci. Vol. 14, 1972.

5. ANDERSON, J.A., ROSENFELD, E., " Neurocomputing: Foundations of Research", MIT Press, 1988.

6. ASH, T., "Dynamic Node Creation in backpropagation Networks", ICE Report, Cognitive Science Dept., University of California, Feb 1989.

7. AITCHINESS, J., " Prototypical Knowledge for Expert Systems", Artificial Intelligence, No. 20, 1983.

8. ATLAS, L., CONNOR, J., DAMBORG, M., " Comparisons of Conventional Techniques and Neural Networks In Power System Load Forecasting", Proc. of the American Power Conf. Vol. 2, 1991.

9. BACHA, H., MEYER, W., " Neural Network Architecture for Load Forecasting", IJCNN Int. joint Conf. on Neural Networks, Baltimore, 1992.

10. BARRAT, R., RAMSEY, A., SLOMAN, A., "POP-11 A Practical Language for Artificial Intelligence", Ellis Horwood Ltd. 1985.

11. BARTO, A., SUTTON, R., ANDERSON, C., " Neuron-like Adaptive Elements that can Solve Difficult Learning Control Problems", IEEE Trans. Systems Man and Cyber., SMC-13, 1986.

12. BINFORD, T.O., "Survey of Model Based Image Analysis Systems", International Journal of Robotics Research, No. 1, Vol. 1 1982.

13. BLAND, A., " Peak Demand Forecasting", Water Demand Forecasting Workshop, Leicester University, July 1985.

14. BOLAND, J.J., DZIEGIELEWSKI, B., " Forecasting Urban Water Use - The IWR MAIN Model", Water Resources Bulletin, Vol. 25, No.1, 1989.

15. BOLZERN, P., FRONZA, G., "Role of Weather Inputs in Short Term Load Forecasting of Electric Load", Electrical Power and Energy Systems, No.1 Vol. 8 Jan 1986.

16. BOOKER, L.B., GOLDBERG, D.E., HOLLAND, J.H., " Classifier Systems and Genetic Algorithms", Artificial Intelligence, Vol. 40 , 1989.

17. BOX, G.E.P., JENKINS, G.M., "Time Series Analysis - Forecasting and Control", Holden and Day, 1970.

18. BUCHANAN, B.G., SHORTLIFFE, E.H., "Rule Based Expert Programs: The MYCIN Experiments of the Stanford Heuristic Programming Project", Addison Wesley, MA, 1984.

19. BUCHANAN, B.G., DUDA, R.O., " Principles of Rule Based Expert Systems", Advances In Computers, Vol. 27, 1985.

20. BUCHANAN, B.G., FEIGENBAUM, E.A., " Dendral and Metadendral: Their Applications Dimension", Artificial Intelligence, Vol. 11, 1978.

21. BUNN, D.W., FARMER, E.D., "Comparative Models for Electrical Load Forecasting", John Wiley and Sons Ltd. 1985.

22. BUNN, D.W., "Short Term Forecasting: A Review of Procedures in the Electricity Supply Industry", Journal of the Operational Research Society, No.6 Vol. 33.

23. BUNN, D.W., SEGAL, J.P., "Forecasting the Effects of Television Programs upon Electrical Loads", Journal of the Operational Research Society, No.1 Vol.34, 1983.

24. CAMPO, R., RUIZ, P., "Adaptive Weather Sensitive Sort Term Load Forecast", IEEE Trans on Power Systems, No.3 Vol. PWRS-2, Aug 1987.

24. CHAN, D.Y.C., PRAGER, D., " Analysis of Time Series by Neural Networks", IEEE Int. Joint Conf. on Neural Networks, Vol. 1, 1991.

25. CHEN, J.R., MARS, P., " Artificial Neural Networks and Nonlinear Systems Identification", School of Engineering and Applied Science, University of Durham, 1990.

26. CHEN, R.H., WANG, S.Y., XIANG, N.D., "Building Expert System for Static Security Assessment of Power Systems", Preprints of the 1989 Symp. on Power Systems and Power Plant Control, Korea 1989 pp 702-707.

27. CHEN, S.T., YU, D.C., MOGHADDAMJO, A.R., " Weather Sensitive Short Term Load Forecasting Using Non-Fully Connected Artificial Neural Network", IEEE Trans. Power Systems, Vol. 7 No. 3, 1992.

28. CHISTIANSE, W.R., "Short Term Load Forecasting Using General Exponential Smoothing", IEEE Tarns an PAS-90, No.2 Vol. 90 March/April 1971.

29. CHURCHLAND, P.S., " Neurophilosophy", MIT Press 1986.

30. CLAIRE, P., HATABIAN, G., MULLER, C., " Progress in Forecasting by Neural Networks", IJCNN Int. Joint Conf. on Neural Networks, Vol. 2, IEEE 1992.

31. CLANCEY, W.J., " The Epistemology of a Rule Based Expert System: A Framework for Explanation", Artificial Intelligence, Vol. 20, 1983.

32. COELHO, S.T., ALEGRE, H., " Demand Analysis in Water Supply and Distribution Systems", Nat. Laboratory of Civil Eng. LNEC, Lisbon, Portugal.

33. COHEN, B.L., " A Powerful and Efficient Structural Pattern Recognition System", Artificial Intelligence, Vol. 9, No. 3, 1977.

34. COULBECK, B., TENNENT, S.T., ORR, C.H., "Introduction to the GINAS Network Analysis Program", Leicester Polytechnic, Research Report No.55 1984.

35. COULBECK, B., TENNENT, S.T., ORR, C.H., "Development of a Demand Prediction Program for use in the Optimal Control of Water Supply", Leicester Polytechnic, Research Report No.36 1985.

36. CUBERO, R.G., " Neural Network for Water Demand Time Series Forecasting", Proc. Int. workshop on Artificial Neural Networks, Spring. Verlag, 1991.

37. DAMBOURG, M.J., CHEN, M., "An Example of Integrating an Expert System into a Control Centre Using a Dispatcher Training Simulator", Symposium on Expert Systems Applications to Power Systems, Stockholm-Helsinki, Aug 1988.

38. DAVIES, M., "The relationship Between Weather and Electricity Demand", Journal of Institute of Electrical Engineers, Vol. 106C 1958.

39. DE MOYER, R., HOROWITZ, L.B., " A System Approach to Water Distribution Modelling and Control", Lexington Books, 1973.

40. DEHDASHTI,A.S., TUDOR,J.R., SMITH, W.C., "Forecasting Hourly Load by Pattern Recognition. A Deterministic Approach", IEEE Trans on PAS-101, No. 9 Vol. 101, Sept 1982.

41. DIETTERICH, T.G., MICHALSKI, R.S., " Discovering Patterns In Sequences of Events", Artificial Intelligence, Vol. 25, 1985.

42. DILLON, T.S., SESTITO, S., LEUNG, S., " Short Term Load Forecasting Using an Adaptive Neural Network", Electrical Power and Energy Systems, Vol. 13 No. 4, 1991.

43. DILLON, T.S., "Expert Systems: Potential and Limitations in the Application to Power Systems", Symposium on Expert Systems Applications to Power Systems, Stockholm-Helsinki, Aug 1988.

44. DRYAR, H.A., "The Effect of Weather on the System Load", AIEE Transactions, Vol. 63 1944.

45. DUDA, R.O., HART, P.E., NILSSON, N.J., SUTHERLAND, G.L., " Semantic Network Representations in Rule Based Inference Systems", Pattern Directed Representations in a Rule Based Inference System", Pattern Directed Inference Systems, Ed. by Waterman and Hayes-Roth, Academic Press, NY, 1978.

46. ERNOULT, M., MATTATIA, R., MESLIER, F., RABUT, P., "Estimation of the Sensitivity of the Electrical Energy Demand to the Variations in Meteorological Conditions", Electrical Power and Energy Systems, Vol. 5, No. 3 July 1983.

47. ERNOULT, M., MATTATIA, R., "Short Term Load Forecasting: New Developments at The E.D.F.", 8th PSCC Helsinki, pp 369-375.

48. FARMER, E.D., " A Method of Predicting for Non-Stationary Processes and its Application to the Problem of Load Estimation", IFAC 1963 pp. 47-54.

49. FARMER, E.D., " On Line Load Prediction", Central Electricity Research Laboratories, REF. RD/L/ 1969.

50. FARMER, E.D., POTTON, .M.J., " Prediction of Load on a Power System", Proc. of the 3rd IFAC Congress, London, 1966.

51. FLETCHER, R., POWELL, M.J.D., " A Rapidly Convergent Decent Method for Minimisation", Computing Journal, Vol. 6, 1963.

52. FOSTER, W.R., COLLOPY, F., UNGAR, L.H., " Neural Network Forecasting of Short Noisy Time Series", Computers and Chemical Engineering, Vol. 16, No. 4, 1992.

53. FUKUSHIMA, K., "Neocongnitron: A Hierarchical Neural Network Capable of Visual Pattern Recognition", Neural Networks, Vol. 1, 1988.

54. FUKUSHIMA, K., MIYAKE, S., " Neocognitron: A New Algorithm for Pattern Recognition Tolerant of Deformations and Shifts of Position", Pattern Recognition, Vol. 15, No. 6, 1982.

55. GANN, J.O., " On Line Consumer Demand Forecasting Using Adaptive Time Series Analysis", School of Engineering and Applied Science, University of Durham.

56. GINSBERG, A., " Automatic Refinement of Expert System Knowledge Bases", Research Notes in AI, Pitman, Morgan and Kaufman.

57. GOH, T.N., ONG, H.L., LEE, Y.O., "A New Approach to Statistical forecasting of Daily Peak Power Demand", Electric Power Systems Research, No.2 Vol. 10 March 1986.

58. GOLDBERG, D.E., " Genetic Algorithms in Search and Optimisation of Machine Learning", Addison Wesley, MA, 1989.

59. GROSS, G., GALIANA, F.D., " Short Term Load Forecasting", Proceedings of the IEEE, No. 12, Vol. 75, Dec 1987.

60. GROSSBERG, S., " Neural Networks and Natural Intelligence", MIT Press, 1988.

61. GROSSBERG, S., " Embedding Fields: Underlying Philosophy, Mathematics and Applications to Psychology, Physiology and Anatomy", Journal of Cybernetics, Vol. 1 1971.

62. GUPTA, P.C., YAMADA, K., "Adaptive Short Term Forecasting of Hourly Load Using Weather Information", IEEE Trans. on PAS-91, Vol. 91 Sept/Oct 1972.

63. HAGAN, M.T., BEHR, S.M., " The Time Series Approach to Short Term Load Forecasting", IEEE Trans. on Power Systems, No. 3 Vol PWRS-2, Aug 1987.

64. HAYES-ROTH, B., " A Blackboard Architecture for Control", Artificial Intelligence, Vol. 26, 1985.

65. HAYES-ROTH, F., WATERMAN, D.A., LENAT, D.B., "Building Expert Systems", Addison Wesley, 1983.

66. HEBB, D., " The Organisation of Behaviour", Wiley, New York 1949.

67. HECHT-NIELSEN, R., "Neurocomputing", Addison Wesley, 1987.

68. HECHT-NIELSEN, R., "Applications of Counterpropagation Networks", Neural Networks, Vol. 1, 1988.

69. HECHT-NIELSEN, R., " Counterpropagation Networks", Proc. of the Int. Conf. on Neural Networks, Vol. II, IEEE Press, 1987.

70. HIPEL, K.W., MCLOED, A.I., LENNOX, W.C., "Advances in Box-Jenkins Modelling, 1. Model Construction", Water Resources Research, No. 3 Vol. 13 June 1977.

71. HO, K.L., HSU, Y.Y., YANG, C.C., " Short Term Load Forecasting Using a Multilayer Neural Network with an Adaptive Learning Algorithm", IEEE Trans. Power Systems, Vol. 7, No. 1 , 1992.

72. HO, K.L., ET AL " Short Term Load Forecasting of Taiwan Power System Using a Knowledge Based Expert System", IEEE Trans. Power Systems, Vol. 5 , No. 4, 1990.

73. HOGG, R.A., " Expert system Based Load Prediction", School of Engineering and Applied Science, University of Durham, 1989.

74. HOLLAND, J.H., " Adaption in Natural and Artificial systems", The University of Michigan Press, 1975.

75. HOPFIELD, J.J., " Neural Networks and Physical Systems with Emergent Collective Computational Abilities", Proceeding of the National Academy of Sciences, Vol. 79, 1982.

76. HSU, Y.Y., YANG, C.C., " Design of Artificial Neural Networks for Short term Load Forecasting Part 1 - Self Organising Feature Maps", IEE Proc. C Vol. 138, Sept. 1991.

77. HSU, Y.Y., YANG, C.C., " Design of Artificial Neural Networks for Short term Load Forecasting Part 1 - Multilayer Feedforward Networks", IEE Proc. C Vol. 138, Sept. 1991.

78. JABBOUR, K., RIVEROS, J.F.K., LANDSBERGEN, D., MEYER, W., " ALFA: Automated Load Forecasting Assistant", IEEE Trans. Power Systems, No. 3 Vol. 3 Aug. 1988.

79. JOWITT, P.W., ET AL " Real Time Forecasting and Control for Water Distribution ", Computer Application in Water Supply, Vol.2 " Ed. by Coulbeck and Orr , Research Studies Press Ltd, 1987.

80. JOWITT, P.W., XU, C., " Demand Forecasting for Water Distribution Systems", Int. Journal of Environ. and Anal. Chemistry , Vol. 26, 1992.

81. KALMAN, R.E., " Proceeding of the 1st Symposium on Engineering Applications of Random Function Theory and Probability" John Wiley and Sons, 1963.

82. KAPLAN, J., " Co-operative Responses form a Portable Natural language Query System", Computational Models of Discourse, Ed. by Brady and Batwick, MIT Press, 1983.

83. KARHUNEN, K., Ann. Acad. Science Fennicae A.I. No. 37 1947.

84. KATZ, B., WINSTON, P.H., " A Two Way Natural Language Interface", Integrated Interactive Computing Systems, Ed, by Degano and Sandewall, North Holland, Amsterdam, 1982.

85. KERONEN, J.J., " An Expert System Prototype for Event Diagnosis and Real Time Operational Planning in Power Systems Control", IEEE Trans. Power Systems, No. 2 , Vol. 4 May 1989.

86. KEYHANI, A., DANIELS, H., " Real Time Load Modelling Technique: An Application to Short Term Load Prediction", IEEE 1974 pp 132-136.

87. KIRSCHEN, D.S., WOLLENBERG, B.F., IRISARRI, G.D., BANN, J.J., MILLER, B.N., "Controlling Power Systems During Emergencies: The Role of Expert Systems", IEEE Computer Applications in Power , April 1989.

88. KNIGHT, B., " Sensor Validation Techniques Using Expert Systems Techniques", Expert Systems and Process Control Conference, Brunel University, 1991.

89. KOHONEN, T., " Self Organising Formation of Topologically Correct Feature Maps", Biol. Cyber. , Vol. 43, 1982.

90. LAING, W.D., SMITH,D.G.C., " A Comparison of Time Series Methods of Prediction CEGB Demand", 9th PSCC Portugal, pp 369-375.

91. LAUGHTON, M.A., TERJESON, N.J., " Medium and Long Term Energy Demand Forecasting: The Case for a Knowledge Based Approach", Symposium on Expert Systems Applications to Power Systems, Stockholm-Helsinki, Aug 1988.

92. LEE, K.C., YANG, J.S., PARK, S.J., " Neural Network Based Time Series Modelling: ARMA Model Identification via ESACF Approach", IEEE Int. Joint Conf. on Neural Networks, Vol. 1, 1991.

93. LEE, K.Y., CHA, Y.T., KU, C.C., "A Study on Neural Networks for Short Term Load Forecasting", Proc. of 1st Int. Forum on Applications of Neural Networks to Power Systems, Ed. by El Sarkawi and Marks, IEEE, 1991.

94. LEE, K.Y., CHA, Y.R., PARK, J.H., "Short Term Load Forecasting Using an Artificial Neural Network", IEEE Trans. Power Systems, Vol. 7, No. 1, 1992.

95. LI, C.M., " Application of Rule Based Knowledge to Load Forecasting", Ph.D Thesis, School of Engineering and Applied Science, University of Durham, 1990.

96. LI, F., " An Adaptive Approach for Short Term Demand Prediction Incorporating Weather Information", Memorandum, NGD Research in Confidence, NGD/L/TSP/0011/M89.

97. LIJESEN, D.P., ROSING, J., " Adaptive Forecasting of Hourly Loads Based on Load Measurements and Weather Information", IEEE Trans. on PAS-90, No. 4 Vol. 90 July/Aug 1971.

98. LINDSAY, R., BUCHANAN, B.G., FEIGENBAUM, E.A., LEDERBERG, J., " Applications of Artificial Intelligence for Chemical Inference: The DENDRAL Project", McGraw Hill , NY, 1980.

99. LYMAN, A.R., " Peak and Off Peak Residential Water Demand", Water Resources Research, Vol. 28, Sept. 1992.

100. MATTHEWMAN, P.D. NICHOLSON, H., " Techniques for Load Prediction in the Electricity Supply Industry", IEE Procedings Part C, IEE No. 10 Vol. 115 1968.

101. MATTHEWMAN, P.D. "Further Investigations into the Method of Prediction of Load Demand Using the Spectral Expansion Technique", Departmental Control Group Report, Sept 1967.

102. MCDERMOTT, J., " R1: A Rule Based Configurer for Computer Systems", Artificial Intelligence, Vol. 19 , No. 1 1982.

103. METROPOLIS, N., ROSENBLUTH, A., ROSENBLUTH, M., TELLER, A., TELLER, E., " Equation of State Calculations by Fast Computing Machines", Journal of Chemical Physics, Vol. 21, 1953.

104. MINSKY, M., " A Framework for Representing Knowledge", Psychology of Computer Vision, Ed. by Winston, MIT Press 1975.

105. MOGHRAM, I., RAHMAN, S., " Analysis and Evaluation of Five Short Term Load Forecasting Techniques", IEEE Trans. on Power Systems, No. 4 Vol. 4 Oct. 1989.

106. MOSS, S.W., "Online Optimal Control of a Water Supply Network", Ph.D Thesis, Cambridge University Engineering Dept. 1975.

107. MULLER, H., LAYR, P., " Very Short term Forecasting - Description of an Efficient Method", 7th PSCC Lausanne, pp. 495-499.

108. MULLER, H., " Classification of Daily Load Curves By Cluster Analysis", 8th PSCC Helsinki pp 381-388.

109. MULLER, H., " An Approach to Very Short Term Load Forecasting by Exponential Smoothing with Trend Correction Based on Previous Day Comparison and Error Difference Smoothing", 6th PSCC Darmstadt.

110. O'SHEA, T., SELF, J., THOMAS, G., "Intelligent Knowledge Based Systems - An Introduction", Harper and Row, 1987.

111. PARK, D.C., EL-SHARKAWI, M.A., MARKS, R.J., ATLAS,L.E., DAMBORG, M.J.," Electric Load Forecasting Using an Artificial Neural Network", IEEE Trans. Power Systems, Vol. 6, No. 2, 1991.

112. PARK, Y.I., PARK, J.K., " An Expert System for Short Term Load Forecasting by Fuzzy Decision",Preprints of the 1989 Symp. on Power Systems and Power Plant Control, Korea 1989 pp 831-836.

113. PARKER, D.B., "Optimal Algorithms for Adaptive Networks: Second Order Backpropagation, Second Order Direct Propagation, and Second Order Hebbian Learning", Proc. of the Int. Conf. on Neural Networks, Vol. 2, IEEE Press, 1987.

114. PARKER, D.B., " A Comparison of Algorithms for Neuron Like Cells", Proc. 2nd Ann. Conf. on Neural Networks for Computing, Ed by Denker, J., Am. Inst. of Physics, 1986.

115. PENG, T.M., HUBELE, N.F., KARADY, G.G., " Advancement in the Application of Neural Networks for Short Term Load Forecasting", IEEE Trans. Power Systems, Vol. 7, No. 1, 1992.

116. PERRY, P.F., "Demand Forecasting In Water Supply Networks", Proc. American Soc. Civil Engineers, ASCE, Vol. 107, No. HY9 1981.

117. PINEDA, F.J., " Recurrent Backpropagation and the Dynamical Approach to Adaptive Neural Computation", Neural Computation, Vol. 1 , 1989.

118. POYSTI, J.L., " Box-Jenkins Method in Short Term Forecasting of Grid Load In Finland", 8th PSCC Helsinki, pp. 357-368.

119. PRATT, A., GINEBRA, J., CATOT, J.M., LORES, J., " Expert System for Forecasting", AI Expert Systems and Languages in Modelling Simulation, IMACS Symposium, Barcelona 1987.

120. QUEVEDO, J., CEMBRANO, G., VALLS, A., SERRA, J., " Time Series Modelling of Water Demand, A Study on Short Term and Long Term Predictions", Technical Document , Sociedad General de Aguas de Barcelona, 1987.

121. RAHMAN, S., BABA, M., " An Integrated Load Forecasting/Load Management Simulator: its Design and Performance", IEEE Trans. on Power Systems No. 1 Vol. 4 Feb. 1989.

122. RAHMAN, S., SHRESTHA, G., " Priority Vector Based Technique for Load Forecasting", IEEE Trans. Power Systems, Vol. 6 , No. 4 , 1991.

123. RAHMAN, S., BABA, M., " Software Design and Evaluation of a Microcomputer based Automated Load Forecasting System", IEEE Trans. on Power Systems No. 2 Vol. 4 May 1989.

124. RAHMAN, S., BHATNAGAR, R., " An Expert System Based Algorithm for Short Term Load Forecast", IEEE Trans. on Power Systems No. 2 Vol. 3 May 1988.

125. REFENES, A.N., "CLS: An Adaptive Learning Procedure and its Application to Time Series Forecasting", IEEE Int. Joint Conf. on Neural Networks, Vol. 1, 1991.

126. REITER, R., MACKWORTH, A.K., " A Logical Framework for Depiction and Image Interpretation", Artificial Intelligence, Vol. 41, 1990.

127. REMIOR, M., AYUSO, J.L., " E.L.F.O.S. Expert System for Short Term Load Forecasting",Symposiumm on Expert Systems Applications to Power Systems, Stockholm-Helsinki, Aug 1988.

128. ROSENBLATT, F., "The Perceptron: A Probabalistic Model for Information Storage and Organisation in the Brain", Psychol. Review, Vol. 65, 1958.

129. ROSENFELD, A., " Multiresolution Image Processing and Analysis", Springer Verlag, NY, 1983.

130. RUMELHART, D.E., McLELLAND, J.L., " Parallel Distributed Processing: Explorations in the Microstructure of Cognition, I and II", MIT Press, 1986.

131. SCHANK, R.C., RIEGER, C.J., " Inference and the Computer Understanding of Natural Language", Artificial Intelligence, Vol. 5 , 1974.

132. SEKINE, Y., " Application of AI Techniques to Power Systems", Symposium on Expert Systems Applications to Power Systems, Stockholm-Helsinki, Aug 1988.

133. SHAMIR, U., SHVARTSER, L., FELDMAN, M., " Forecasting Hourly Water Demands by a Pattern Recognition Approach", Journal of Water Resources Planning and Management, ASCE, Vol. 119, No. 6, 1993.

134. SHARDA, R., PATIL, R.B., " Connectionist Approach to Time Series Prediction, an Empirical Test", Journal of Intelligent Manufact., Vol. 3, No. 5, 1992.

135. SHARMA, K.L., MAHALANABIS, A.K., "Recursive Short Term Load Forecasting Algorithm", IEE Proceedings Part C, IEE No. 1 Vol. 121 Jan 1974.

136. SINGH, A., EMMANUAL,P., KALRA, P.K., "Point of View for Development of Knowledge Based System for Load Forecasting", Symposium on Expert Systems Applications to Power Systems, Stockhlom - Helsinki, 1988.

137. SIMPSON, P.K., " Artificial Neural Systems", Pergammon Press, 1990.

138. SMITH, D.G.C. " Demand Forecasting over Special Periods Using a Modified Box-Jenkins Method", CEGB Research in Confidence, TPRD/L/ECS162/M87 1987.

139. SMITH, R.G., MITCHELL, T.M., WINSTON, H.A., BUCHANAN, B.G., "Representation and Use of Explicit Justifications for Knowledge Base Refinement", Proc. 9th Int. Joint Conf. on AI. 1985.

140. SMITH, D.G.C. " Short Term Demand Prediction Using a Univariate Box-Jenkins Method", Memorandum, CEGB Research in Confidence, TPRD/L/CI383/M86 1986.

141. SRINIVASAN, D., LIEW, A.C., CHEM, J.S.P., " Short Term Forecasting Using Neural Network Approach", Proc. of 1st Int. Forum on Applications of Neural Networks to Power Systems, Ed. by El Sarkawi and Marks, IEEE, 1991.

142. STEINER, R.C., " Short Term Forecasting of Municipal Water Use" Ph.D Dissertation, The Johns Hopkins University, Baltimore, 1984.

143. STEINER, R.C., " Short Term Forecasting of Municipal Water Use" A Critical Assessment of Forecasting in Western Water Resources Management, American Water Resources Association, 1984.

144. STERLING, M.J.H., BARGIELA, A., " Adaptive Forecasting of Daily Water Demand", Comparative Models in Electrical Load Forecasting, Ed. by Bunn and Farmer, John Wiley and Sons, 1985.

145. STERLING, M.J.H., " Power Systems Control", IEE, Peter Peregrinus Ltd. London 1978.

146. STERLING, M.J.H., ANTCLIFFE, D.J., " A Technique for the Prediction of Water Demand from Past Consumption Data Only", Inst. of Water Engineers, No. 8 , Vol. 28, 1974.

147. STERLING, M.J.H., BARGIELA, A., " The Enhancement of Time Series Modelling Techniques for Adaptive Forecasting", Department of Engineering , University of Durham.

148. TAYLOR, T., LUBKEMAN, D., " Applications of Knowledge Based Programming to Power Engineering Problems", IEEE Trans. on Power Systems, No. 1 , Vol. 4 Feb. 1989.

149. THOMPSON, R.P., " Weather Sensitive Electric Demand and Energy Analysis on a Large Geographically Diverse Power System - Application to Short Term demand Forecasting", IEEE Trans. on PAS-95, No.1 Vol. 95 1976.

150. TOYODA, J., CHEN, M., INOUE, Y., " An Application of State Estimation to Short Term Load Forecasting, Part 1 - Forecasting Modelling", IEEE Trans. on PAS-89, No. 7 Vol 89, 1970.

151. TSOI, A.C., KOBE, M.U., " Load Forecasting in a Power System from a Supply Authority Point of View", Electric Power Systems Research, No. 1 Vol. 6 Jan. 1983.

152. VEMURI, S., HUANG, W.L., NELSON, D.J., " On Line Algorithms for Forecasting Hourly Loads of an Electrical Utility", IEEE Trans. on PAS-100, No. 8 Vol. 100 Aug. 1981.

153. WALKER, R.S., "Water Industry Expert Systems Club Final Report", Pub. By Water Research Centre, May 1988.

154. WASSERMAN, P.D., "Neural Computing Theory and Practice", Van Nostrand Reinhold, 1989.

155. WATER RESEARCH CENTRE, " Leakage Control Policy and Practice", Standing Technical Committee, Report No. 26.

156. WATERMAN, D.A., " A Guide to Expert Systems", Addison-Wesley 1985.

157. WEIZENBAUM, J., "ELIZA- A Computer Program for the Study of Natural Language", Communication Between Man and Machine, 1966.

158. WIDROW, B., HOFF, M.E., " Adaptive Switching Circuits", IRE WESCON Convention Record, New York , 1960.

159. WILLIS, H.L., PARKS, T.W., " Fast Algorithm for Small Area Electric Load Forecasting", IEEE Trans. on PAS-102, No. 10, Vol. 102, Oct. 1983.

160. WILSON, L., LUKE, R., " Discussion on Forecasting Urban Water Use ", Water Resources Bulletin, Vol. 26, No. 3 1990.

161. WINSTON, P.H., BINFORD, T.O., KATZ, B., LOWRY, M.R., " Learning Physical Descriptions from Functional Definitions, Examples and Presidents", Proc. of the National Conference on Artificial Intelligence, Washington D.C., 1983.

162. WU, H.T., LU, C.N., " Using Artificial Neural Network for Providing Hourly Load Update and Next Day Load Profile", Int. Conf. on Advances in Power Systems Control, Operation and Management, Vol. 2, IEE, 1991.

163. YDSTIE, B.E., " Forecasting and Control Using Adaptive Connectionist Networks", Computers in Chemical Engineering , Vol. 14, No. 4/5, 1990.

164. ZAIYONG TANG, DE ALMEIDA, C., FISHWICH, P.A., " Time Series Forecasting using Neural Networks vs. Box Jenkins Methodology", Simulation, Vol. 57, No. 5, 1991.

165. ZHANG, Z.Z., HOPE, G.S., MALIK, O.P., "Expert Systems in Electric Power Systems - A Bibliographical Survey", IEEE Trans. on Power Systems, No. 4 , Vol. 4 1989.

APPENDIX A

GRAPHICAL USER INTERFACE

This appendix provides a description of the elements of the graphical user interface that has been implemented as part of the combined demand forecasting implementation outlined in this thesis.

One of the highly important aims of the demand forecasting application as described in Chapters 4, 5 and 6 was to involve the operator as much as possible in the process of arriving at the final demand prediction profile upon which the water network control decisions for the coming 24 hours would be based. The reasons for wishing to achieve a high level of operator involvement are twofold, the first is that an experienced operator holds a significant amount of heuristic system knowledge both in terms of foreseeing events that are likely to occur that would influence consumption within the system and in the ability to diagnose abnormal effects that are observed in historical data. The second reason is that it is highly important that the operator has confidence in the performance of the forecasting system and this confidence can only be achieved by involving the operator in the prediction process and providing him/her with information on how a particular prediction result was arrived at.

The rule base described in Chapter 4 provides a method for incorporating heuristic knowledge into the demand forecast, however, this is not a mechanism designed to remove the operator from the prediction process, its aim is instead to maximise the contribution the operator can make towards improving the forecast accuracy. There will always be situations where an event occurs that was not foreseen at the time of the rule base construction, or when system changes make adjustment of the rule base necessary, or when a rule needs to be manually triggered. In each of these cases the successful operation of the demand forecaster is dependent on the input of the operator. The menu driven system associated with the rule base is designed to allow updating and adjustment of the knowledge held within the rules by non computer expert personnel. This is necessary because it is envisaged that a demand forecasting system such as the one described here would be operated for the majority of the time by system control staff i.e. not highly computer literate engineers.

In addition to the manipulation of heuristic information held within the rule base, it is equally important to provide the operator with a user friendly means of reviewing the performance of the system and to aid in identifying trends in the current data or unexpected deviations from the expected consumption profile. A highly effective method of achieving this via a graphical user interface and such a systems has been designed and written for the demand forecaster using the GKS Graphical Kernel System. The GKS interface operates from within the FORTRAN controlling program and provides a mouse operated system that is simple and easy to use. The GKS procedures are passed the raw, modified and historical prediction data as well as actual consumption data and meteorological data for the week leading up to the prediction day. The required data can then be selected for display via mouse activated click boxes.

The main display screen is shown in Figure A.1 and is composed of a consumption against time graph covering the 24 hour prediction period. The click boxes in the upper right portion of the screen allow the user to select the profiles he requires for display on the graph. The profiles available for display are, the unaltered ARIMA prediction, the modified neural network prediction, the profile of the actual consumption for the previous day and the profile of the actual consumption for the same day as the prediction day the previous week. These profiles can be superimposed upon each other or removed from the display as required by the user, thereby allowing the most effective comparison of the available data. The purpose of including the previous days and previous weeks profiles is to provide a reference for the operator against which to evaluate the current prediction i.e. if the current prediction is radically different to the previous weeks profile then there must be a reason for this, either there has been some change to the system or the weather conditions are significantly different, the operator could then inspect the neural predictor day type or the rule base to determine the likely reason for the difference.

The click boxes in the bottom right of the main screen allow other display options to be selected that show the performance of both the ARIMA and neural network prediction systems in terms of prediction accuracy. The actual and predicted profiles for the previous day can be displayed in a similar format to the main display screen (Figure A.2), the profiles being superimposed and an RMS percentage error figure being displayed to indicate the relative accuracy achieved by each prediction methodology. The operator can also select a screen that indicates the performance of the prediction systems over the past week. The actual consumption profile for the previous seven days can be shown superimposed with either or both of the seven days of ARIMA and neural prediction results. Below this on the same screen the operator can choose to display a graphical plot of the past seven days of weather variables (maximum temperature, sunshine hours and rainfall totals). This allows a visual assessment to be made of the current consumption trends and their relation to the prevailing meteorological conditions, which in turn is useful in selecting neural prediction days types and weather related rules for the next days prediction. Figure A.3 shows a representation of the weeks data screen.

A statistics screen can also be selected that displays the current prediction profiles within an error band based on the average RMS prediction errors over the preceding seven days, Figure A.4 illustrates an example of this display.

Figure A.1 The Main Display Screen.

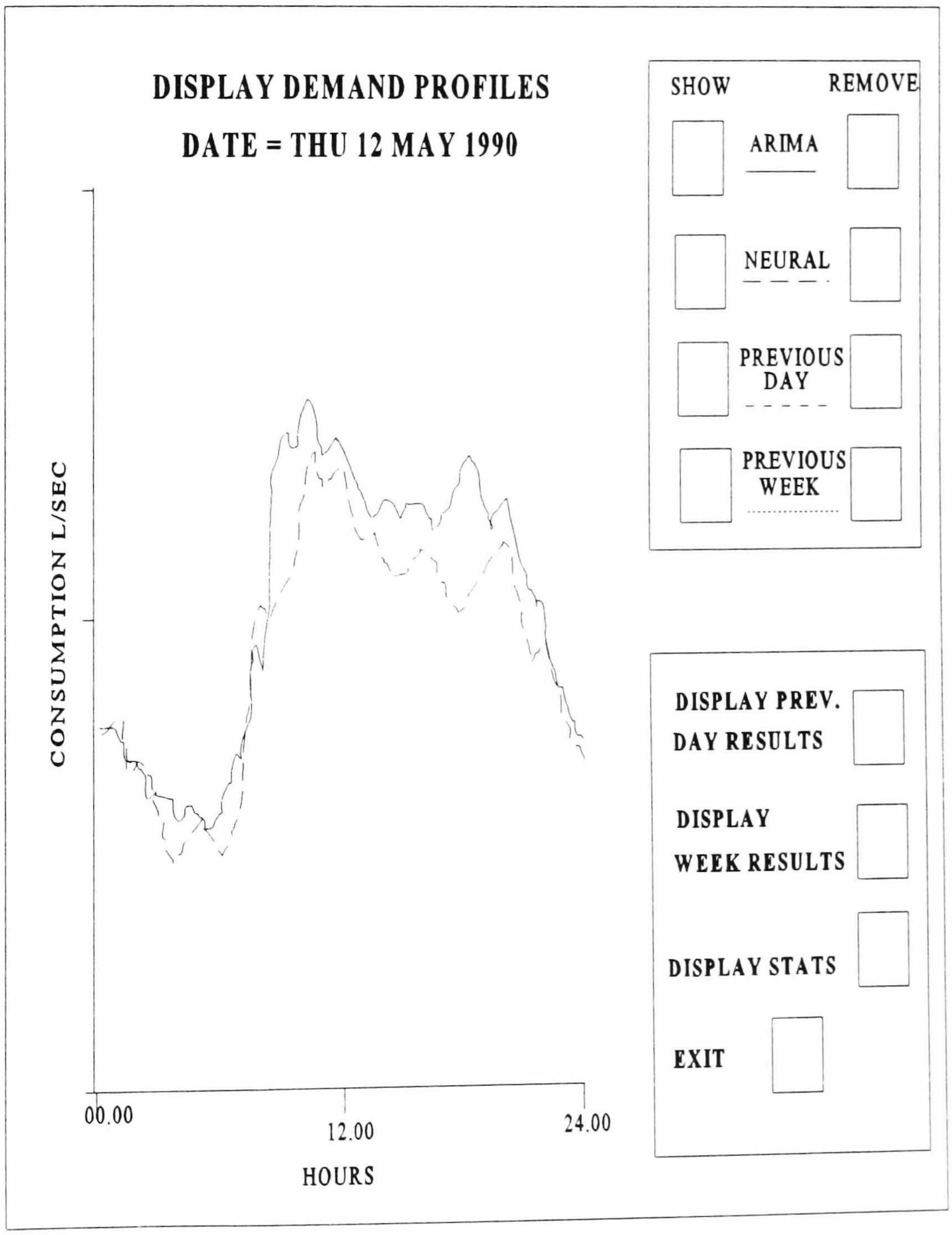


Figure A.2 The Previous Day Prediction Results Screen.

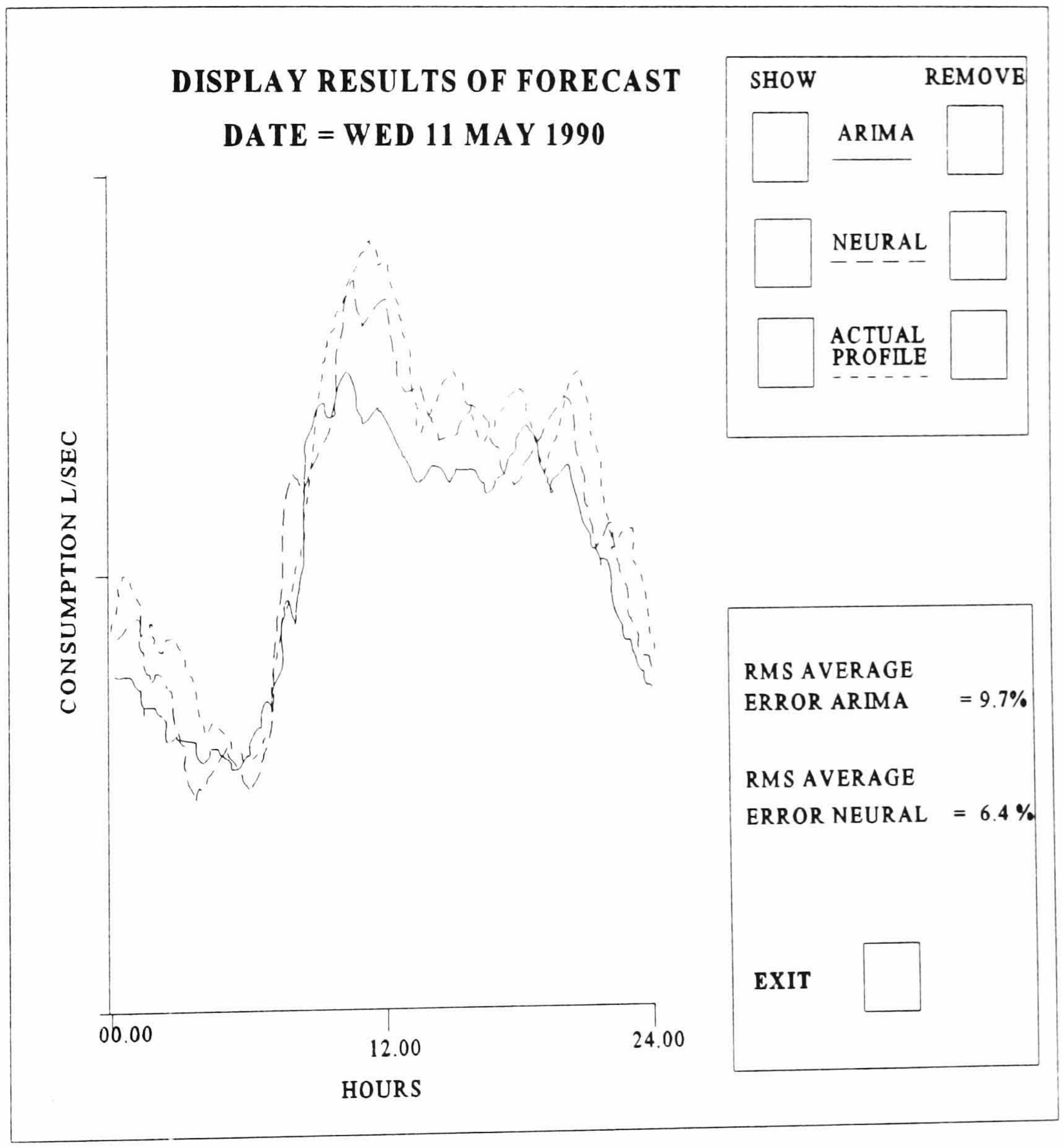


Figure A.3 Screen Showing Prediction Performance and Meteorological Variables.

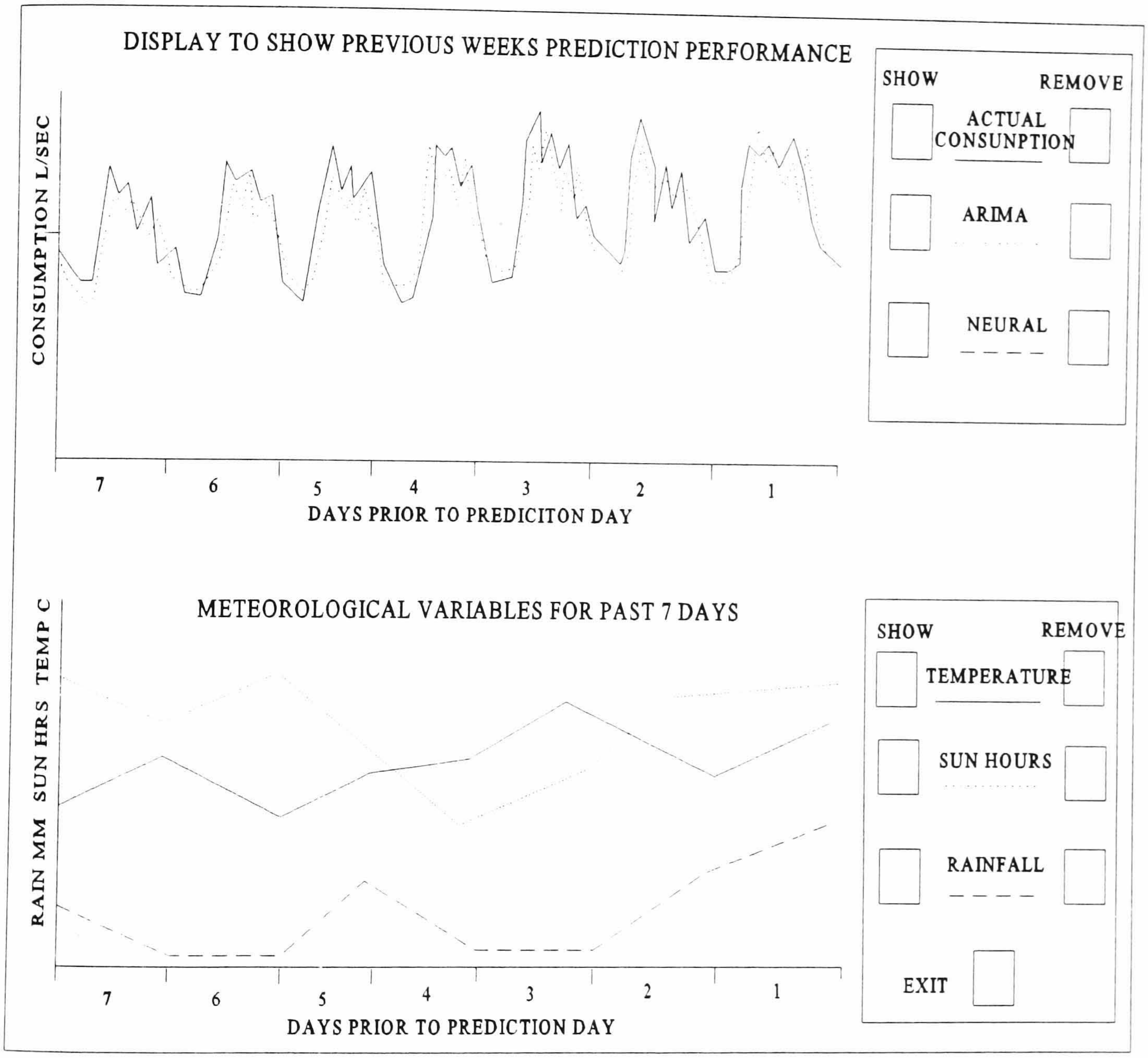


Figure A.4 The Statistical Error Band Screen.