ACKNOWLEDGEMENTS

The author would like to thank Professor V. Panuska for proposing the project and for his expert guidance throughout the preparation of this report.

The author would also like to express his gratitude to Professor J. C. Giguere and Dean French for granting him the extension necessary to complete the report.

Finally, the author would like to thank Mrs. Julie Strick for her skillful typing of the manuscript and his wife for her general help and encouragement.

Montreal, Quebec
March 1978

Jitender Hitkari

TABLE OF CONTENTS

LIST OF SYMBOLS

SECTION	SYMBOL		DESCRIPTION
1.1	$x(t)$	-	Observed variable
2.2.2	$Z(t)$	-	Variable being predicted
	$\underline{a}_t$	-	Model coefficient vector
	$\underline{f}(t)$	-	Fitting function vector
	$e(t)$	-	Uncorrelated noise term
	$\underline{L}$	-	Transition matrix
	$\hat{Z}_1(T)$	-	One-step ahead prediction of $Z(t)$
	$\underline{h}$	-	Smoothing vector
	l	-	Lead time of prediction
	α	-	Smoothing constant
2.3.3	w_t	-	A series consisting of random drawings from a Gaussian distribution
	μ	-	Mean of the time series Z_t
	Z_t	-	Observed time series
	ϕ_k	-	A model parameter to be estimated ($k = 1, 2, 3 \ldots p$)
	Θ_k	-	A model paramter to be estimated ($k = 1, 2, 3 \ldots q$)
	z_t	-	The deviation of the observed time-series from its mean
	B	-	Backward shift operator
	θ_o	-	A function of ϕ's and μ defined by eqn. (2.15)
	p,q	-	Order of the model

SECTION	SYMBOL		DESCRIPTION
2.3.4	∇_t	-	Differencing operator
	$\gamma(B)$	-	Generalized autoregressive operator
	$\theta(B)$	-	Moving average operator
	$\psi(B)$	-	Weighting factors defined by eqn. (2.25)
2.3.5	ψ'	-	Weighting factors defined by eqn. (2.28)
	E	-	Expected value operator
	$V(l)$	-	Variance of prediction error, l being lead time
	$e_t(1)$	-	One-step ahead prediction error
	$\hat{z}_t(1)$	-	One step ahead prediction of variable z_t
2.4	$\hat{x}$	-	Prediction of variable x
	$\underline{x}$	-	State vector
	$\underline{z}$	-	Observed system output vector
	$\underline{H}$	-	Measurement matrix
	$\underline{L}$	-	State transition matrix
	$\underline{K}$	-	Kalman gain matrix
2.5.2	z	-	Variable being predicted
	x_n	-	Independent variable
	f,h	-	Functions of x
2.5.4	a_x	-	Parameters of x
	u	-	A function of x
	D	-	Sum of squares of deviations $(z(x)-u(x))$

SECTION	SYMBOL		DESCRIPTION
3.3	Z_t	-	Total electrical load
	Y_t	-	Periodic component
	R_t	-	Residual component
	N'	-	Number of harmonics
	t	-	Basic unit of time
	α, β	-	Model parameters
	U_t	-	A temperature dependent function
	E	-	A correlated noise term
	W	-	White noise
	h	-	Delay factor
	A,B,C,D	-	Polynomials in backward shift operator Z^{-1} with coefficients a_i, b_i, c_i and d_i respectively
	P_t, $\gamma_{i,t}$	-	These symbols result by applying Einstein's notation to make the expression compact
	r	-	Lead time of prediction
	[ ]	-	Square brackets denote conditional expectations
	$\hat{Z}_t(r)$	-	Prediction of z_t at lead time r
	$[Z_{t+r}]$	-	Conditional expectation of Z_t at lead time r.
	∇_{24}	-	Seasonal differencing operator

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 GENERAL

The objective of this report is to investigate how prediction algorithms can be implemented on micro and minicomputers. The term 'prediction algorithm', as employed in this report denotes a systematic procedure to predict or forecast the future values of an observed variable. It is assumed that the observations are made at equidistant intervals of time. Suppose a variable $x(t)$ is observed up to an instant of time t. The prediction algorithm provides an estimate of $x(t+1)$, or more generally $x(t+l)$, where l is called the lead time of prediction. There is, of course, no universally applicable algorithm available [6]. Instead, one has to look for the algorithm most appropriate for a given set of conditions. In addition, it must be computationally simple so that it can be implemented on a micro or minicomputer.

1.2 CONTENTS OF THE REPORT

In Chapter 2, a brief discussion of mathematical modelling is presented. This is followed by a survey of prediction techniques which include exponential smoothing and time-series methods.

In Chapter 3, small computer systems are defined and their general characteristics are given. The minicomputer system used by the author is also described. This is followed by a detailed description of the models and prediction algorithms used in the application example. This example predicts the electrical power load over a lead time of 24 hours.

In Chapter 4 the results of the implemented algorithms are discussed. This includes the error summary and memory maps for each of the algorithms.

Finally, in Chapter 5, conclusions are presented.

CHAPTER 2

A SURVEY OF PREDICTION TECHNIQUES

2.1 INTRODUCTION

In this chapter a number of prediction techniques are described. There are two main categories of prediction methods. First there are methods based on developing suitable stochastic (or probabilistic) models [6]. For the last several years considerable attention has been given to stochastic modelling. The underlying principle is to consider the observed variable as the output of a stochastic process [6]. Thus, one has to identify the stochastic process which evolves the observed variable. The concepts of prediction theory are then applied to develop prediction algorithms. It is sometimes useful to depart from a purely stochastic model approach and develop the prediction model in two parts. One part describes the deterministic behaviour of the variable and the other part, its stochastic behaviour [19]. Therefore, two models are required. This is, of course, dependent upon the application under consideration. For example in the prediction of the outcome of a chemical process there is no identifiable deterministic behaviour; for that reason a purely stochastic model would be required. On the other hand, in the prediction of electrical load, for example, one can identify a deterministic trend. In this case a two part model can be developed.

The stochastic models are also capable of incorporating external factors which affect the future value of the variable. This is illustrated in the application example (Chapter 3) in which the future values of the electrical load are affected by the future temperature values, temperature being an external variable.

The second class of prediction methods are the so-called, non-probabilistic methods. These methods consist of representing the observed variable by suitable mathematical functions such as polynomials and exponential functions. Hence, matching a suitable function to the data becomes the gist of the prediction. Such prediction methods are, in essence, mathematical approximations of observed variable and are not capable of incorporating any external factors which affect the future values of the variable. Thus, this class of methods has limited value. The last part of this chapter deals briefly with these methods.

2.1.1 Concluding Remarks

Model building is somewhat of an art in the sense that a certain amount of educated guesswork is inevitable during the development stage[8]. Thus, no model can be said to be the best since every model will necessarily have its limitations. The choice of a model for a particular application is also dependent upon the way the model is to be used and the accuracy required [6].

The prediction techniques to be described in the following sections are selected on the basis of the mathematical technique employed in arriving at a prediction alogrithm.

2.2 Exponential Smoothing Method

2.2.1 General

Christiaanse [7] employs general exponential smoothing technique in the development of a prediction algorithm. His method uses a deterministic component and an error term. The description of the method follows.

2.2.2 Description of the Algorithm

The model proposed consists of two terms:

i) a linear combination of known functions of time

ii) a noise component.

$$\underline{Z}(t) = \underline{a}_t \underline{f}(t) + e(t) \qquad (2.1)$$

where -

$\underline{Z}(t)$ = value of the observed variable Z(t) at time t

$\underline{a}_t(t)$ = model coefficients, assumed constant over lead time of prediction

$\underline{f}(t)$ = a fitting function chosen for a particular set of data

$e(t)$ = uncorrelated, normally distributed random variables with zero mean and constant variance.

To implement the method we require an appropriate set of fitting functions and a method to estimate the coefficients a_t from the observed values of the data under consideration.

The fitting functions must be chosen such that the expected value of the data for the maximum lead time of prediction can be described as a linear combination of these functions. Fourier series representation for the fitting functions is often employed [7], [12], [15].

The initial estimate of the coefficients a_t are computed using the weighted least square criterion, by minimizing the expression —

$$\sum_{i=0}^{T} \alpha^{i} \left[\underline{Z}(T-i) - \underline{a}'(T)\, \underline{f}(T-i) \right]^{2} \quad \ldots \ldots \ldots (2.2)$$

where

T = current time (the last instant for which the variable Z(t) is known

α = smoothing constant such that $0<\alpha<1$.

Equation (2.2) is the exponentially weighted sum of the squared deviations of the model from the observed data. The smoothing constant α controls the rate at which the past errors are discounted.

The choice of α is made on some criterion such as standard error obtained. High values of α give predictions which depend much more on recent observations of Z(t), whereas low values result in predictions which depend on a large number of past observations.

The estimates of the coefficients are updated as actual data becomes available, according to —

$$\underline{a}(T) = \underline{L}'\, \underline{a}(T-1) + \underline{h}\left[Z(t) - \hat{Z}_1(T) \right] \quad \ldots \ldots \ldots (2.3)$$

where $\underline{L}$ = a transition matrix, which is assumed to exist for a chosen set of fitting functions $\underline{f}(t)$ such that

$$\underline{f}(t) = \underline{L}\, f(t-1) \quad \ldots \ldots \ldots (2.4)$$

$\underline{h}$ = a smoothing vector obtained from

$$\underline{h} = \left[\sum_{i=0}^{\infty} \alpha^{i}\, \underline{f}(-i)\underline{f}'(-i) \right]^{-1} \underline{f}(0) \quad \ldots \ldots \ldots (2.5)$$

and $\hat{Z}_1(T)$ = one step ahead prediction of the variable $Z(t)$.

The predictions can be computed by extrapolating equation (2.1) giving —

$$Z_l(T + l) = \underline{a}'(T)\ \underline{f}(t + l) \quad \ldots\ldots\ldots\ldots \quad (2.6)$$

where l = lead time of prediction.

2.2.3 Discussion of the Method

The prediction algorithm is general in nature. Any linear combination of analytical functions such as polynomials, exponentials, etc. can be easily written in the transition matrix form of equation (2.4). It is assumed that the matrix

$$\sum_{i=0}^{\infty} \alpha^i\ f(-i)\ \underline{f}'(-i) \quad \ldots\ldots\ldots\ldots \quad (2.7)$$

exists for chosen α and $\underline{f}(t)$ function. The condition which insures that the matrix exists is $f_k(t) > \alpha^{-t/2}$, $k = 1, 2, 3 \ldots\ldots$ This is not a very restrictive condition so that the matrix in equation (2.7) will exist in most cases. The condition merely means that the fitting function $\underline{f}(t)$ when extrapolated into the past should not increase at a rate faster than the rate at which the exponential weighting function decays [7].

The algorithm indicates that the predictions are based solely on the past values of the observed variable. The fact that the model parameters are readily updated as the next observation becomes available is a good feature of the algorithm. One matrix multiplication and a vector addition are required to update the model. This method would be suitable in an on-line environment where the computer constantly updates the model and outputs the predictions.

2.3 TIME SERIES METHOD

2.3.1 GENERAL

The time series approach to prediction and control was developed by Box and Jenkins [2,3]. Chatfield [6] also described prediction algorithms based on the time series idea. It was proposed to represent the observed variable as a time series with observations available at discrete equispaced time intervals. These time series would be fitted by stochastic models.

2.3.2 STATIONARY TIME-SERIES

The concept of stationarity plays a vital role in time series analysis. A series is said to be stationary if its first moments, namely its mean and variance, are independent of the shift in time origin. This may seem restrictive at first but many nonstationary time series encountered in practice can be converted into a stationary one by a simple process of differencing as discussed in Section 2.3.5.

2.3.3 AUTO-REGRESSIVE & MOVING-AVERAGE MODELS

Let the values of a stationary time series of a variable z_t, at times t, t-1, t-2, . . . be z_t, z_{t-1}, z_{t-2}, . . . respectively. Also let w_t, w_{t-1}, w_{t-2}, . . . be a series consisting of random drawings from a gaussian distribution having zero mean and some variance σ_w^2. The discussion of the time series models which follow were originally developed by Yule [3]. The time series z_t, whose successive observations are assumed to be highly correlated, is transformed into a series of uncorrelated components w_t which can be thought of as evolving the series. There are two ways of generating such series:

1) The deviation of the observed series from its mean at the time origin t can be made linearly dependent on previous such deviations and on w_t. Thus an autoregressive model is obtained as —

$$z_t - \mu = \phi_1 (z_{t-1} - \mu) + w_t \quad \ldots\ldots\ldots\ldots \quad (2.8)$$

where μ is the mean of the original time series assuming stationarity and ϕ_1 is a model parameter to be estimated.

Let

$\bar{z}_t = z_t - \mu$, then,

$$\bar{z}_t = \phi_1 \bar{z}_{t-1} + w_t \quad \ldots\ldots\ldots\ldots \quad (2.9)$$

Equation (2.9) is a first order autoregressive model.

2) The other method to generate the time series is to make z_t linearly dependent on w_t and on its previous values w_{t-1}, w_{t-2}, w_{t-3} . . . . . etc. This results in a finite moving average model, thus we have —

$$\bar{z}_t = w_t + \Theta_1 w_{t-1} \quad \ldots\ldots\ldots\ldots \quad (2.10)$$

Equation (2.10) is a first order finite moving average model.

In order to represent moving average behaviour using the autoregressive model, one would require an infinite number of autoregressive terms and vice versa. In order that the number of parameters are as few as possible, both kinds of terms are included to give a general autoregressive moving average model of order (p,q) as —

$$\bar{z}_t - \phi_1 \bar{z}_{t-1} \ldots\ldots - \phi_p \bar{z}_{t-p} = w_t - \Theta_1 w_{t-1} \ldots - \Theta_q w_{t-q} \quad \ldots\ldots\ldots\ldots \quad (2.11)$$

where p and q are 0, 1 or 2 in most practical cases.

To manipulate the model it is convenient to define a backward shift operator B such that —

$$Bz_t = z_{t-1} \quad \ldots \quad (2.12)$$

Equation (2.11) can be expressed as —

$$\phi_p(B)\bar{z}_t = \theta_q(B)w_t \quad \ldots \quad (2.13)$$

where

$$\phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 \ldots \ldots - \phi_p B^p \quad \ldots \quad (2.14)$$

and

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

$\phi_p(B)$ and $\theta_q(B)$ are polynomials in B of degree p and q which have to be estimated from the observed data. The condition for stationarity of the general model equation (2.13) is that the roots of $\phi(B) = 0$ and $\theta(B) = 0$ must lie outside the unit circle [3]. The general model can be expressed in terms of the observed variable z_t instead of the deviation $\bar{z}_t$ as —

$$\phi_p(B)z_t = \theta_o + \theta_q(B)w_t \quad \ldots \quad (2.15)$$

where

$$\theta_o = (1 - \phi_1 - \phi_2 \ldots \ldots - \phi_p)\mu$$

2.3.4 Homogeneous Nonstationary Time Series

Many physical processes are best represented by nonstationary time series. Fig. 2.1(a) shows a nonstationary series commonly encountered. The characteristic of the series is that it looks much the same if we examine it piecewise, except for the level difference. It can be shown that the first order differencing of such series yields a series which exhibits stationarity, namely —

$$\nabla_t z = z_t - z_{t-1} = (1 - B)z_t \quad \ldots \quad (2.16)$$

Fig. 2.1(b) shows another very frequently met nonstationary series. It can be shown that for such series a second order differencing yields a stationary series [3], that is —

$$\nabla_t^2 z_t = (1-B)^2 z_t \quad \text{(2.17)}$$

Incorporating the above ideas into the general model of equation (2.13) we get —

$$\phi_p(B)(1-B)^d z_t = \theta_0 + \theta_q(B) w_t \quad \text{(2.18)}$$

where it is assumed that the dth order differencing is necessary to obtain stationarity. We can further define —

$$\gamma_{p+d}(B) = \phi_p(B)(1-B)^d \quad \text{(2.19)}$$

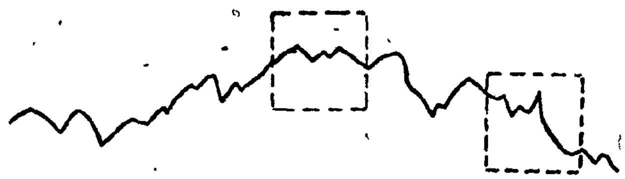
$\gamma_{p+d}(B)$ is called the generalized autoregressive operator. $\theta(B)$ is called the moving average operator. We can also omit θ_0 from the general model without loss of generality since it represents a deterministic linear trend in the time series which is not related to the stochastic behaviour represented by the model. Thus the final form of the model becomes —

$$\gamma_{p+d}(B) z_t = \theta_q(B) w_t \quad \text{(2.20)}$$

This model can be expressed in three explicit forms:

1) Difference equation form
2) The model in terms of current and previous values of w_t
3) In terms of weighted sums of the previous values of z_t and w_t.

Only the first two forms are discussed in this report. These forms are used to derive prediction algorithms.



(a)

(b)

Fig. 2.1(a) - A series showing non-stationarity in level.

(b) - A series showing non-stationarity in level and slope.

The difference equation form is obtained as follows —

$$\gamma_{p+d}(B) = \phi_p(B)(1-B)^d = 1 - \gamma_1 B - \gamma_2 B^2 \; \ldots\ldots \; \gamma_{p+d} B^{p+d} \quad \ldots (2.21)$$

also

$$z_t - \gamma_1 z_{t-1} - \gamma_2 z_{t-2} \ldots\ldots - \gamma_{p+d} z_{t-(p+d)} = w_t - \theta_1 w_{t-1}$$

$$-\theta_2 w_{t-2} \ldots\ldots - \theta_q w_{t-q} \quad \ldots\ldots (2.22)$$

therefore

$$z_t = \gamma_1 z_{t-1} \ldots\ldots \gamma_{p+d} z_{t-(p+d)} - \theta_1 w_{t-1} \ldots - \theta_q w_{t-q} + w_t$$

$$\ldots\ldots (2.23)$$

Equation (2.23) is the difference equation form of the general model. To derive the second form of the general model let —

$$z_t = w_t + \psi_1 w_{t-1} + \psi_2 w_{t-2} + \ldots\ldots \quad (2.24)$$

which is a linear model with output z_t and whose input is the gaussian process w_t defined in section 2.3.3.

Equation (2.24) can be expressed as —

$$z_t = w_t + \sum_{i=1}^{\infty} \psi_i w_{t-i} \quad \ldots\ldots (2.25)$$

$$= \psi(B) w_t$$

Substituting w_t from eqn. (2.25) into the general model of equation (2.20) we get —

$$\gamma_{p+d}(B)\psi(B)w_t = \theta_q(B)w_t$$

so that

$$\gamma_{p+d}\psi(B) = \theta_q(B)w_t \quad \ldots\ldots (2.26)$$

Hence the ψ weights may be determined by comparing coefficients in the expression —

$$(1 - \gamma_1 B - \gamma_2 B^2 \; . \; . \; . \; . \; - \gamma_{p+d} B^{p+d})(1 + \psi_1 B + \psi_2 B^2 + \; . \; . \; . \;)$$

$$= (1 - \theta_1 B - \theta_2 B^2 \; . \; . \; . \; . \; . \; . \; - \theta_q B^q) \quad \ldots\ldots\ldots (2.27)$$

2.3.5 Derivation of the Prediction Algorithm

Let us assume that the prediction function steps ahead is of the form —

$$\hat{z}_t(l) = \psi'_l w_t + \psi'_{l+1} w_{t-1} + \psi'_{l+2} w_{t-2} \quad \ldots\ldots\ldots (2.28)$$

where the weights ψ'_l , ψ'_{l+1} . . . . are to be determined. Then using equation (2.25) the mean square error of prediction may be written as —

$$E[z_{t+l} - \hat{z}_t(l)]^2 = (1 + \psi_1^2 + \psi_2^2 + \; . \; . \; . \; . \; \psi_{l-1}^2)\sigma_W^2 + \sum_{i=0}^{\infty}$$

$$[\psi_{l+i} - \psi'_{l+i}]^2 \sigma_W^2 \quad \ldots\ldots\ldots (2.29)$$

Equation (2.29) is minimized if $\psi'_{l+i} = \psi_{l+i}$ which is a result derived from prediction theory [3]. Thus we have —

$$z_{t+l} = (\psi_l w_t + \psi_{l+1} w_{t-1} + \; . \; . \; . \;) + (w_{t+l} - \psi_1 w_{t+l-1} \; . \; . \; . \; .$$

$$\psi_{l-1} w_{t+1}) = \hat{z}_t(l) + e_t(l) \quad \ldots\ldots\ldots (2.30)$$

where $\hat{z}_t(l)$ is the l step ahead prediction and $e_t(l)$ is the corresponding prediction error. It is seen from equation (2.30) that the minimum mean square error prediction at origin t, for lead time l, is the conditional expectation of z_{t+l}, at time t. That is —

$$E[z_{t+l} \mid z_t, z_{t-1}, \ldots] = \hat{z}_t(l) \quad \ldots\ldots\ldots\ldots \quad (2.31)$$

Also, the prediction error for lead time l is given by —

$$e_t(l) = w_{t+l} + \psi_1 w_{t+l-1} \cdots\cdots \psi_{l-1} w_{t+1} \quad \ldots\ldots \quad (2.32)$$

and since

$$E[e_t(l)] = 0.$$

Hence the prediction is unbiased and the estimate of the prediction error is zero. The variance of the prediction error is given by —

$$V(l) = (1 + \psi_1^2 + \psi_2^2 \cdots \psi_{l-1}^2)\, \sigma_w^2 \quad \ldots\ldots\ldots \quad (2.33)$$

From equations (2.30) and (2.32) the one step ahead prediction error is

$$e_t(1) = z_{t+1} - \hat{z}_t(1) = w_{t+1} \quad \ldots\ldots\ldots\ldots\ldots \quad (2.34)$$

This is a key result since it gives significance to the gaussian process w_t which so far has been an unknown. It is seen that the values of w_t are in fact the one step ahead prediction errors. The prediction algorithm can now be established using equation (2.23) by taking its conditional expectation at time $t+l$. Lets define the conditional expectance operator as —

$$E[z_t \mid z_{t-1}, z_{t-2} \ldots] = [z_t] \quad \ldots\ldots\ldots \quad (2.35)$$

Then applying the conditional expectation operator to both sides of equation (2.23) we get —

$$[z_{t+l}] = \hat{z}_t(l) = \gamma_1[z_{t+l-1}] + \ldots + \gamma_{p+d}[z_{t+l-p-d}]$$

$$- \theta_1[w_{t+l-1}] \cdots - \theta_q[w_{t+l-q}] + [w_{t+l}] \quad \ldots\ldots \quad (2.36)$$

The various conditional expectations can be determined using the following results [3].

$$[z_{t-i}] = z_{t-i}, \qquad i = 0, 1, 2, 3 \ldots\ldots$$
$$[z_{t+i}] = \hat{z}_{t-i}, \qquad i = 1, 2, 3, 4 \ldots\ldots \qquad (2.37)$$
$$[w_{t+i}] = 0 \qquad i = 1, 2, 3, 4 \ldots\ldots$$
$$[w_{t-i}] = z_{t-i}-\hat{z}_{t-i-1}(1), \qquad i = 0, 1, 2, 3. \ldots$$

2.3.6 Concluding Remarks

The method described above is computationally simple since it involves fewer processing steps. The difference equation is strictly an algebraic expression applied recursively to obtain predictions. The only cumbersome quantities to compute are the one step ahead predictions for the noise components w_t since their computations involve backtracking the time origin stepwise and determining these predictions. The method has been applied practically in the field of electrical engineering by researchers such as Panuska [15] and Vemuri et al [20].

2.4 Kalman Prediction Algorithm

Once the system model is arrived at, and an initial estimate of its parameters obtained, the Kalman filtering and prediction algorithms can be applied. The model is put in the state space form. The Kalman filtering is employed first to estimate the current state using past data values. Then the prediction algorithm is used to determine the one-step ahead prediction of the state. The mechanics of the Kalman filter is such that it adjusts the initially estimated state in accordance with the difference between the newly available output and the output calculated from the initial estimate of the state. In other words the filter operates in a 'predict-correct' fashion [13]. The filtered estimate is given by —

$$\underline{\hat{x}}(j+1 \mid j+1) = \underline{L}(j+1, j)\underline{\hat{x}}(j \mid j)-\underline{K}(j+1)[\underline{z}(j+1)-\underline{H}(j+1)\underline{L}(j+1,j)\underline{\hat{x}}(j|j)] \quad \ldots \ldots \ldots (2.38)$$

for $j = 0, 1, 2, 3, \ldots\ldots,$ where $\underline{x}(0|0) = 0$

$\underline{x}$ is an 'n vector', the state, $\underline{\hat{x}}$ being the prediction of x

$\underline{z}$ is an 'm vector', the observed system output

$\underline{H}$ is an m x n matrix called the measurement matrix

$\underline{K}$ is an n x m matrix called the Kalman gain matrix

$\underline{L}$ is an n x n state transition matrix.

Fig. 2.2 illustrates the filtering scheme. Let us assume that the filtered estimate at some time origin t = j is known and that we want to compute the one step ahead estimate of $\underline{x}$ given the observation z(j+1). First we must premultiply the initial state estimate $\underline{\hat{x}}(j|j)$ by the state transition matrix $\underline{L}$. This step results in the dynamic extrapolation of the preceding state estimate. Then the term

$\underline{z}(j+1)-\underline{H}(j+1)\underline{L}(j+1,j)$ in equation (2.38) is computed. This term represents the measurement residual defined by

$$\underline{z}(j+1|j) = \underline{z}(j+1)-\underline{z}(j+1|j) \quad \ldots \ldots \ldots (2.39)$$

The residual is then premultiplied by K(j+1), the weighting matrix and the result is added to the first term computed, to obtain the one step ahead prediction of variable $\underline{x}$. The following relationship expresses the l step ahead prediction of $\underline{x}$ as —

$$\underline{\hat{x}}(j+l \mid j) = \underline{L}(j+l, j)\underline{\hat{x}}(j|j) \quad \ldots \ldots \ldots (2.40)$$

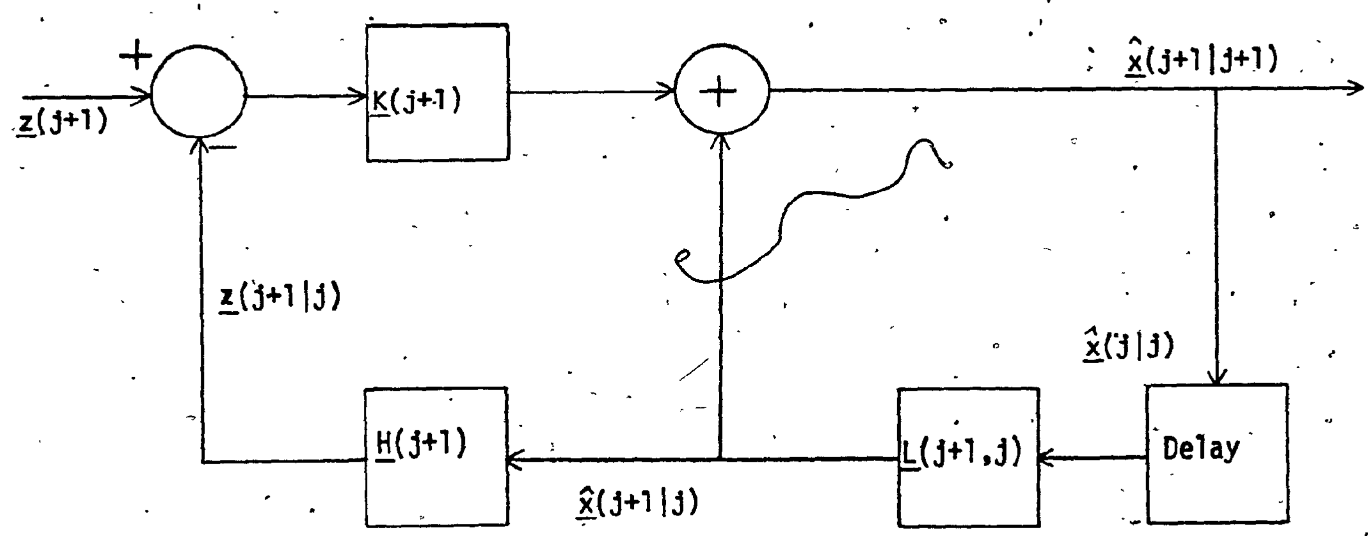


Fig. 2.2 — Kalman Filter

The computation of Kalman gain matrix K which is expressed in terms of two covariance matrices is discussed in Meditch [13] and will not be dealt with in this report.

The Kalman filter method has been used in power engineering application by Galiana [8] and McIntosh [12] to derive predictions. The method is not suitable for small computer implementation as the matrix manipulations required are rather extensive for a small computer to handle efficiently.

2.5 Some Non-Probabilistic Prediction Techniques

2.5.1 General

In the analysis of a physical process one often ends up with tabulated data. It is plausible that a mathematical representation of the data can be found. This is where curve fitting prediction methods are employed. The most elementary technique is graphical curve-fitting. The observed data points are plotted and the curve which best fits the data is drawn. The curve can then be extended into the future resulting in a short term prediction. The method is obviously crude and therefore limited in value. Other methods, namely the method of collocation and the method of averages can be used more effectively to actually represent the observed data mathematically. A further refinement is the method of least squares which is widely used in engineering analysis. These methods are presented briefly in the following sections.

2.5.2 Mathematical Representations

Let a variable z be measured and tabulated. Further, let z depend on variables x_1, x_2, x_3, ... x_n. We can establish the following relationship.

$$z = f(x_1, x_2, x_3 \ldots\ldots x_n) \ldots\ldots\ldots (2.41)$$

It is required to determine a mathematical function of independent variables, say u where

$$u = h(x_1, x_2, x_3 \ldots\ldots x_n) \ldots\ldots\ldots (2.42)$$

where the function h is evaluated for each set of values of the independent variables, such that the deviations $z(x)-u(x)$ are sufficiently small. In other words the errors are within some acceptable range. The problem boils down to determining the function $h(x)$ and its parameters. The form of this function will depend entirely on the application under consideration. In a large number of cases, however, we can give $h(x)$ a polynomial or a Fourier series representation. As for the parameters, several generalized computing methods exist by which they may be estimated [10].

2.5.3 METHOD OF COLLOCATION

Let $h(x)$ be selected to be some polynomial containing k unknown parameters. In order to determine these we must have at least k sets of data values. If the number of data points and parameters are the same then the latter may be determined by substituting the data values, one set at a time, into the polynomial and then solving a set of k simultaneous equations. The estimated parameters may then be applied to the data points to determine their precision. One possible criterion would be to determine if the parameters result in the least difference between the most positive and the most negative error. Another criterion could be to minimize the sum of the errors. This can be achieved by using the method of averages.

In this method, the data points are divided into say, m groups. The parameters are determined as explained above using each of these groups. Then the estimated parameters from each group are plugged into the chosen function and an average equation is determined using each group. As a result we again have m equations that may be solved for the parameters.

2.5.4 Method of Least Squares

This method relies on the minimization of the sum of the squares of the deviation between the observed and the computed value of a variable. Let a variable z be dependent on another variable x and its parameters a_x such that

$$z = f(x_1, x_2, x_3 \ldots\ldots, a_{x_1}, a_{x_2}, a_{x_3}, \ldots) \quad \ldots (2.43)$$

We seek a function u where

$$u = h(x_1, x_2, x_3 \ldots\ldots, a_{x_1}, a_{x_2} \ldots\ldots) \quad \ldots\ldots (2.44)$$

The sum of squares of the deviations can now be written as —

$$D = \sum_{j=1}^{n} (z_j - u_j)^2 = \sum_{j=1}^{n} (z_j - h(x_1, x_2, x_3 \ldots\ldots a_{x_1}, a_{x_2}, a_{x_3} \ldots))^2 \quad \ldots\ldots\ldots (2.45)$$

The condition that minimizes D is —

$$\frac{\partial D}{\partial a_{x_1}} = 0, \quad \frac{\partial D}{\partial a_{x_2}} = 0, \ldots\ldots \text{ etc}$$

That is, the partial derivatives of D, the deviations, with respect to the parameters of x are equal to zero. This gives us a system of simultaneous equations which can be solved for the parameters a_{x_1}, a_{x_2}, a_{x_3} . . . etc.

2.5.5 Concluding Remarks

The techniques described in the foregoing sections can be applied in prediction computations. The models obtained are totally deterministic. The parameters can be adjusted to best fit the data. The method of least squares is a very versatile tool in mathematical modelling, since we can employ well established laws of mathematics to find the minimum sum of squares of errors. Another very useful modelling aid is the Fourier series as they can approximate any single valued function with finite discontinuities. The computations of model parameters can be easily done using small computers which offer a high-level language such as Fortran.

Chapter 3

DESCRIPTION OF SELECTED ALGORITHMS AND THEIR IMPLEMENTATION

3.1 General Remarks

The model used for the application example is derived from Panuska and Koutchouk [16] and Panuska [15]. It is a weather dependent model representing electrical power load. This model resembles that proposed by Galiana [8], the major difference being the fundamental computational approach. Galiana chose the state-space analysis whereas Panuska uses the model equations directly to develop a prediction algorithm. The latter approach lends itself very well to small computer implementation since no matrix manipulations are involved. It should be emphasized that matrix operations especially inversions, require large memory storage and high CPU execution times. This is one of the major considerations when selecting an algorithm which could be efficiently implemented on small computers. Before undertaking a close study of the model, the general characteristics of small computers are presented. This is followed by the description of the minicomputer system employed by the author to implement the prediction algorithm.

3.2 Small Computer Systems

Small computer systems can be configured using micro or minicomputers. Before proceeding any further the distinction between a microprocessor and a microcomputer must be explained. A microprocessor is the processing element of a microcomputer. Thus a basic microcomputer is a computer consisting of a microprocessor, memory, I/O ports and a power supply. The cost of a microcomputer can range from $1,000 to $7,000, depending upon the hardware components required. A large variety of microprocessors are available in the market today. Fig. 3.1 [1] shows the various families of microprocessors along with their application range. Table 3.2 [4] indicates the general characteristics of microprocessors.

A minicomputer, on the other hand, consists of a more powerful and expensive processor, in addition to memory and I/O ports. The price of a minicomputer can range from $8,000 to $30,000, depending on the configuration. Table 3.3 [5] shows the general characteristics of some widely used minicomputers.

It is not possible to make a one-to-one comparison between micro and minicomputers as their application ranges are different. Figure 3.1 shows that on the far side of the application range the distinction between micro and minicomputers is hard to define. Microprocessors like TI-9900 are as powerful as minicomputer processors and hence could be classified in either category. References [1], [4], [5], are recommended to the reader for in-depth discussion of small computer applications. The choice of a particular processor will depend on the complexity of the computations involved and the programming language used, for a given algorithm. Other factors such as cost and processor speed must also be considered.

In the development of a small computer system for implementing prediction algorithms, both hardware and software design would be required. Fig. 3.1 shows the general steps required in the overall design of such a system.

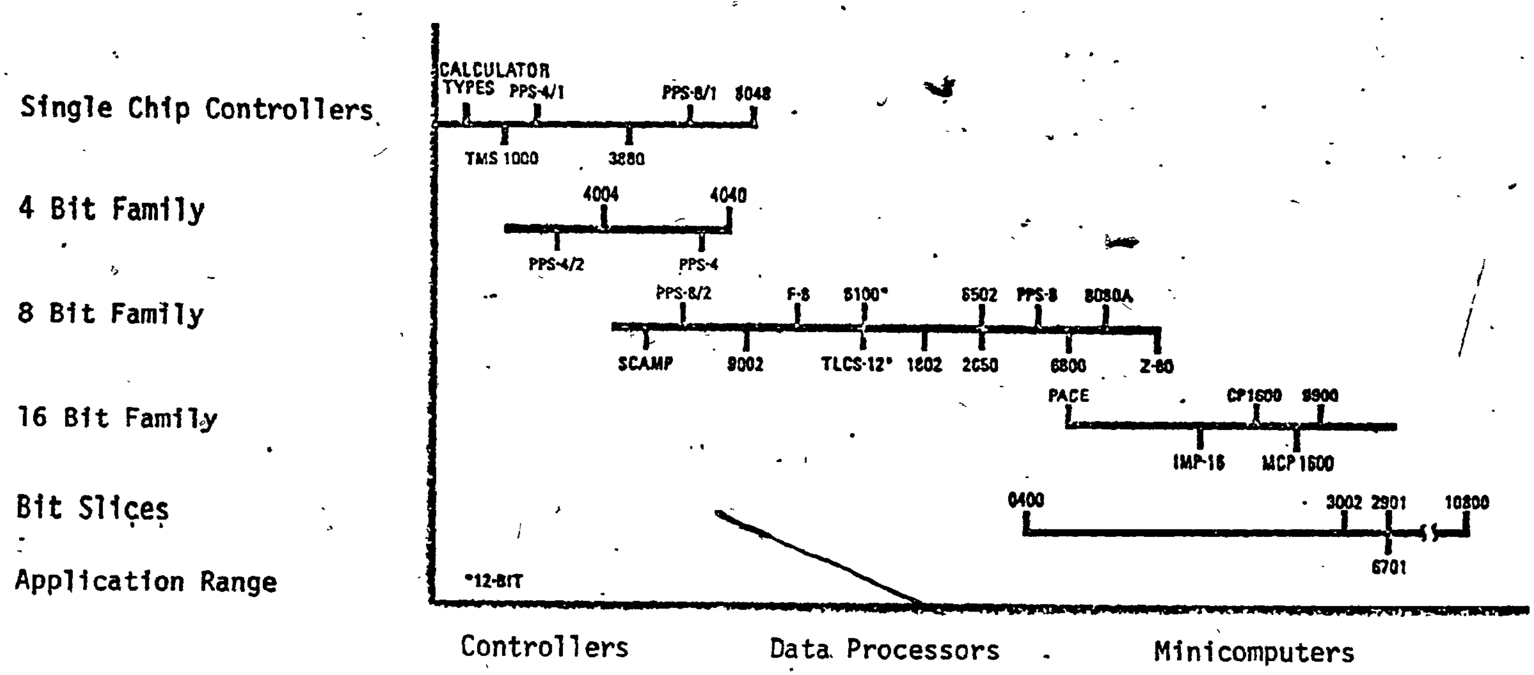


Fig. 3.1 — Microprocessor Families and their Application Range.

Manufacturer	Processor	technology	word size (data/instr.)	direct addressing (words)	# of basic instructions	clock freq MHz/phases	instruction time shortest/longest (µS)	TTL compatible	BCD arithmetic	on-chip interrupts	# of internal registers	# of stack registers	on-chip clock	DMA capability	Memory & I/O avail.	prototyping avail	Assembly language	package size (pins)	supplies (volts)
Motorola	MC14500	CMOS	1/4	0	16	1/1	1/1	Yes	No	Yes	1	0	Yes	No	No	No	No	16	3 to 18
Intel	4004	PMOS	4/8	4k	46	0.74/2	10.8/21.6	No	Yes	Yes	16	3x12	No	No	Yes	No	Yes	16	15
Intel	4040	PMOS	4/8	8k	60	0.74/2	10.8/21.6	No	Yes	Yes	24	7x12	No	No	Yes	Yes	Yes	24	15
NEC Microcomputers	µPD541	PMOS	4/8	4k	69	0.5/2	6.4/38.4	Yes	Yes	Yes	4	8x12	No	Yes	Yes	Yes	Yes	42	5,—5
Fairchild	F8	NMOS	8/8	64k	69	2/1	2/13	Yes	Yes	Yes	64	RAM	Yes	Yes	Yes	Yes	Yes	40	5,12
General Instrument	8000	PMOS	8/8	1k	48	0.8/2	1.25/3.75	No	Yes	Yes	48	0	No	No	Yes	Yes	No	40	5,—12
Intel	8008	PMOS	8/8	16k	48	0.8/2	12.5/37.5	No	Yes	Yes	6	7x14	No	No	Yes	Yes	Yes	18	5,—9
Intel	8080A	PMOS	8/8	64k	78	2.6/2	1.5/3.75	Yes	Yes	Yes	8	RAM	No	Yes	Yes	Yes	Yes	40	5,12,—5
Intel	8085	NMOS	8/8	64k	80	3/1	1.3/5.85	Yes	Yes	Yes	8	RAM	Yes	Yes	Yes	Yes	Yes	40	5
Motorola	M6800	NMOS	8/8	64k	89	2/1	1/2.5	Yes	Yes	Yes	0	RAM	No	Yes	Yes	Yes	Yes	40	5
Motorola	M6809	NMOS	8/8	64k	100+	2/1	2/5	Yes	Yes	Yes	0	RAM	Yes	Yes	Yes	Yes	Yes	40	5
Motorola	M6802	NMOS	8/8	64k	89	2/1	2/5	Yes	Yes	Yes	0	RAM	Yes	Yes	Yes	Yes	Yes	40	5
National Semiconductor	SC/MP	PMOS NMOS	8/8	64k	46	4/1	5/10	NMOS only	Yes	Yes	0	RAM	Yes	Yes	No	Yes	Yes	40	5,—7
NEC Microcomputers	µPD8080A	NMOS	8/8	64k	78	2/2	1.92/8.16	Yes	Yes	Yes	8	RAM	No	Yes	Yes	Yes	Yes	40	5,12,—5
RCA	1802	CMOS	8/8	64k	91	6.4/1	2.5/3.75	Yes	Yes	Yes	16	RAM	Yes	Yes	Yes	Yes	Yes	40	3 to 12
RCA	1803	CMOS	8/8	64k	91	6.4/1	2.5/3.75	Yes	Yes	Yes	16	RAM	Yes	Yes	Yes	Yes	Yes	28	3 to 12
Signetics	2650	NMOS	8/8	32k	75	1.2/1	4.8/9.6	Yes	Yes	Yes	7	8x15	No	Yes	Yes	Yes	Yes	40	5
Zilog	Z80	NMOS	8/8	64k	150+	4/1	1/5.75	Yes	Yes	Yes	14	RAM	No	Yes	Yes	Yes	Yes	40	5
Intersil	6100	CMOS	12/12	4k	81	4/1	2.5/5.5	Yes	No	Yes	0	RAM	Yes	Yes	Yes	Yes	Yes	40	4 to 11
Data General	mN601	NMOS	16/16	32k	42	8.33/2	1.2/29.5	Yes	No	Yes	4	RAM	Yes	Yes	Yes	No	Yes	40	5,10,14,—4.25
Digital	LSI-11/2	NMOS	16/16	64k	66	10/2	2.1/232	Yes	Yes	Yes	8	RAM	No	Yes	Yes	Yes	Yes	40	5,12,—5
Fairchild	9440	I^2L	16/16	64k	42	10/1		Yes	No	Yes	4	RAM	Yes	Yes	No	No	No	40	
General Instrument	CP1600	NMOS	16/16	64k	87	4/2	1.6/4.8	Yes	No	Yes	8	RAM	No	Yes	Yes	Yes	Yes	40	5,12,—3
National Semiconductor	INS8900/PACE	NMOS/ PMOS	16/16	64k	45	2/2	2.5/5	No	Yes	Yes	4	10x16	No	Yes	Yes	Yes	Yes	40	5,8,—12
Texas Instruments	TMS9980	NMOS	16/16	16k	69	4/4	3.2/49.6	Yes	No	Yes	16	RAM	Yes	Yes	Yes	No	Yes	40	5,12,—5
Texas Instruments	TMS/SBP9900	NMOS I^2L	16/16	64k	69	4/4	2/31	Yes	No	Yes	16	RAM	No	Yes	Yes	No	Yes	64	5,12,—5

CANADIAN ELECTRONICS ENGINEERING FEBRUARY 1978

Table 3.1 — Microprocessors.

	Manufacturer/model				
	IBM Series/1 4953	IBM Series/1 4955	DEC PDP 11/34	DG Nova 3D	HP 21MX E
DATA FORMAT					
Word length (bits)	16 + 2 parity		16 + 2 parity	16 + 1 parity	16 + 1 parity
Fixed-point operand length (bits)	16		16	15	16, 32
Instruction length (bits)	16		16, 32, 48	16	16, 32
Floating point	optional		optional	optional	standard
FP operand length (bits)	32/64		32/64	32/64	32/64
MAIN MEMORY					
Storage type	MOSFET		CORE MOS	CORE/MOS	MOS
Cycle time (µs/wd)	0.8	0.66	0.98 0.725	0.7	0.65
Access time (µs/wd)	0.6	0.3	0.51 0.635	0.35	0.325
Minimum size (kilobytes)	16		16	4	4
Maximum size (kilobytes)	128		124	262	2,000
Parity checking	standard		standard	optional	standard
Error correction	no		no	no	optional
Storage protection	none	standard	standard	optional	optional
Storage mapping	optional		standard	standard	optional
CENTRAL PROCESSOR					
Accumulators	–		6	4	2
Index registers	–		6	2	2
Direct addressable kilowords	32		32	256	32
Addressing modes	4		8	8	7
Control storage (kilobits)	read-only store 6	24	n.a.	n.a.	ROM, RAM 8.5
Add time (µs)	8.4	2.42	2.03	0.7	1.94
Hardware multiply/divide	standard		optional	optional	standard
Hardware byte manipulation	standard		standard	optional	standard
Battery backup	optional		optional	optional	n.a.
Real-time clock/timer	optional		standard	optional	optional
INPUT/OUTPUT					
Direct memory access	n.a.		standard	standard	optional
Maximum I/O rate (million words/s)	0.8		2.5	1.1	0.616
External interrupt levels	4		variable	16	60

Source: Lee Walther & Co.

Table 3.2 — Minicomputers.

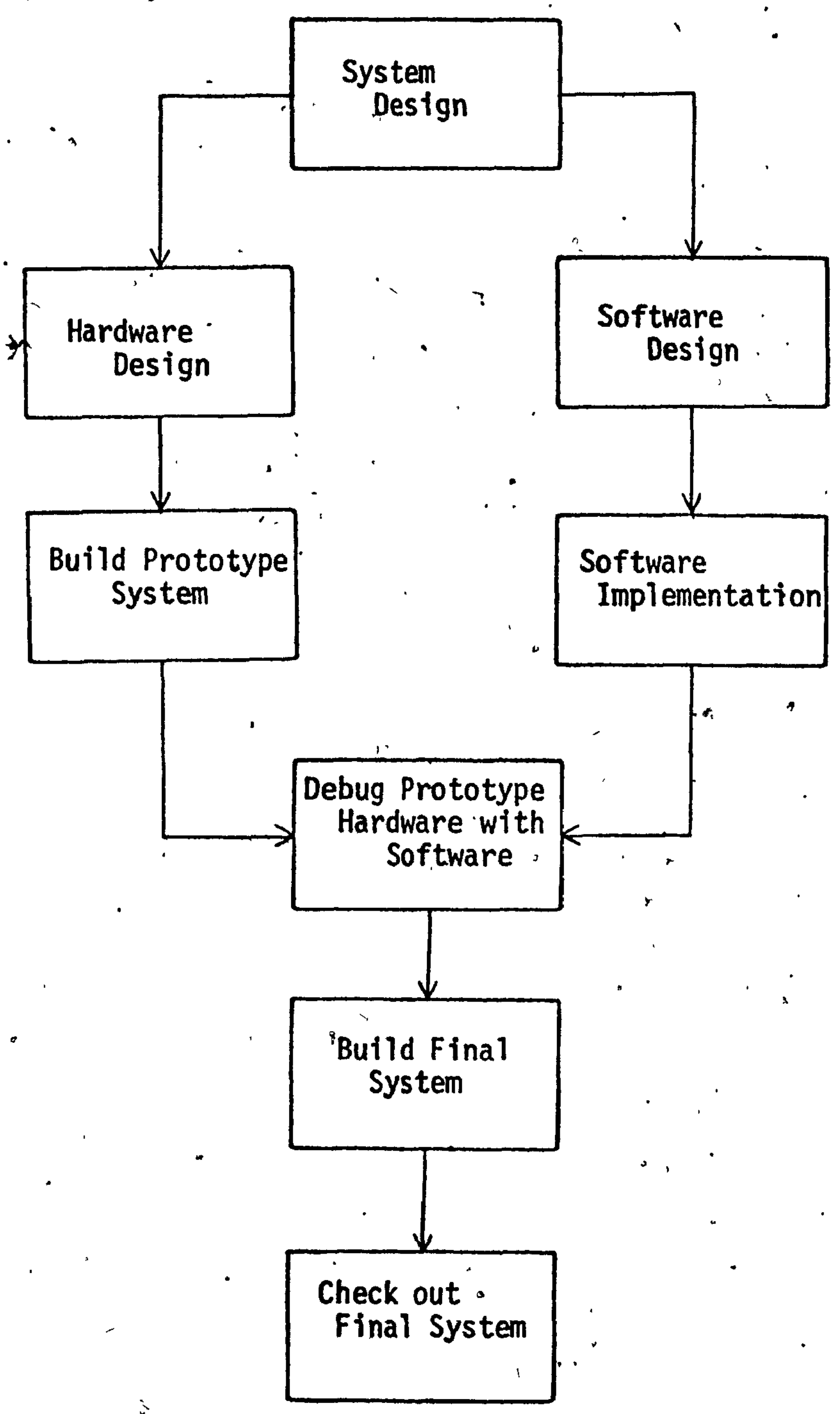


Fig. 3.2 — System Development Block Diagram.

3.3 Description of the System Used

Fig. 3.3 shows the block diagram of the minicomputer system used to implement the prediction algorithms. The CPU is a PDP11/15 minicomputer with 20K words of core memory. The word size is 16 bits. The instruction times vary between 3 and 10 microseconds, depending upon the instruction. A high speed moving-head disk drive is included as a mass storage media for programs and data. The source program in punched card form, is read into the memory using the card reader, and is subsequently stored on the disk. The program is then compiled to obtain an executable load module. The model parameters are entered through the terminal keyboard and the predictions are printed on the line printer. The programming language used is Fortran.

Fig. 3.4 illustrates a proposed microprocessor implementation. The hardware components required are indicated in the diagram. The scratch pad random access memory (RAM) is used to store data and parameters. The program read only memory (PROM) contains the programmed algorithm. The model parameters can be entered through the teletypewriter and the predictions printed out. Alternatively, the data and the parameters can be obtained through a communication interface, from a large computer. The choice of the programming language used for the microprocessor implementation is dependent on the complexity of the prediction algorithm. Traditionally, Assembly language has been used to program microprocessor based systems. However, with the price of solid state memory falling steadily during the past few years, it is not inconceivable that a high-level language such as Fortran could be employed.

In the following sections of this chapter, two algorithms are discussed. One algorithm uses a periodic and a residual component to model electrical power load, and the other algorithm is based solely on the residual model. The results obtained from these algorithms are discussed in the next chapter.

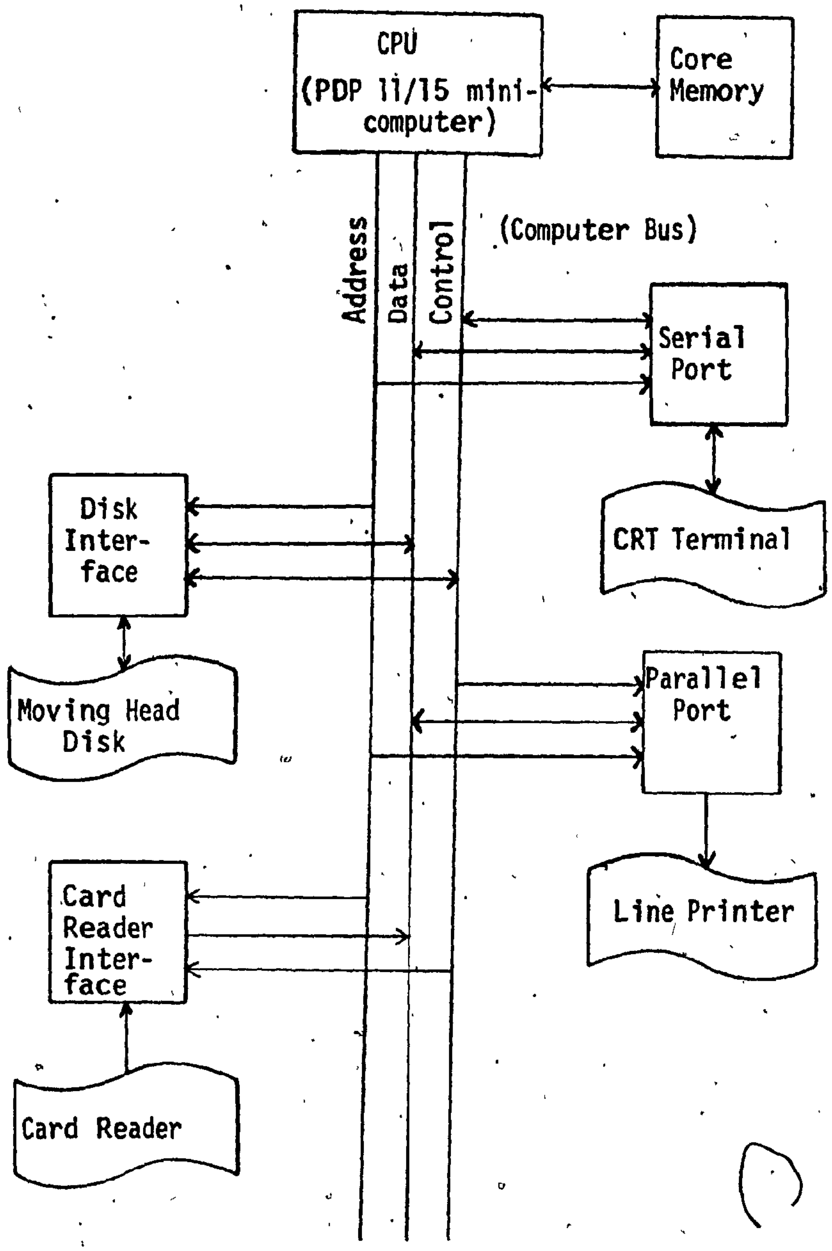


Fig. 3.3 — Minicomputer Implementation.

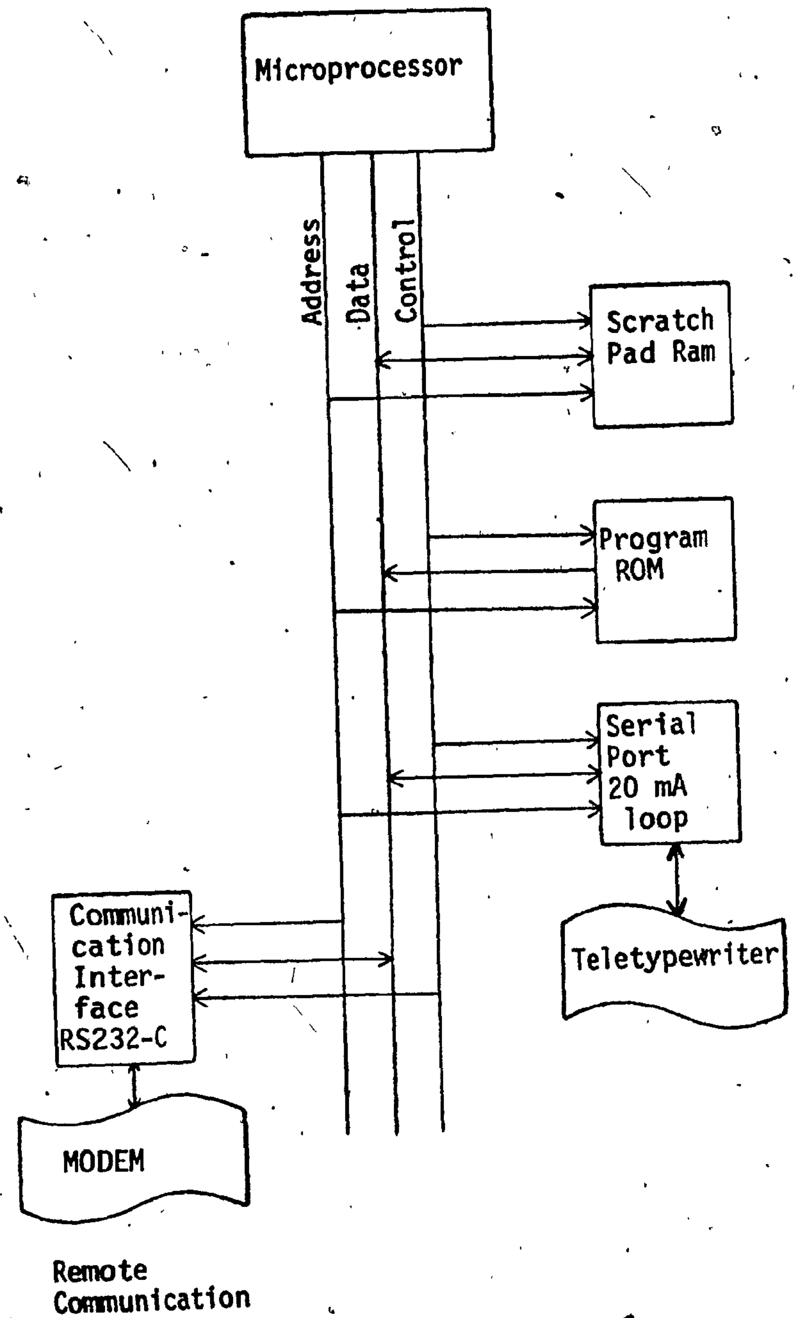


Fig. 3.4— Microprocessor Implementation.

3.3 Description of the Periodic & Residual Model

The model [15, 16] has two components. A periodic load component represented by a Fourier series, called the nominal load, and a residual component in the form of a linear difference equation. The residual term contains the stochastic component of the model, with a temperature function as the driving input. Fig. 3.5 shows the block diagram of the model. The total load is expressed as the sum of the two components as —

$$Z_t = Y_t + R_t \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.1)$$

where Y_t is the periodic term and R_t is the residual term. Y_t is modelled as —

$$Y_t = \alpha_0 + \sum_{k=1}^{N'} \alpha_\phi \sin kgt - \sum_{k=1}^{N'} \beta_k \cos kgt \quad \ldots\ldots\ldots\ldots (3.2)$$

where N' is the number of harmonics. The first 8 harmonics are used.

t is the basic time unit assumed to be 1 hour.

$g = 2\pi/T$, T being the period assumed to be 24 hours.

α_0, α_k, β_k are the estimated coefficients.

The residual term which is modelled by a general linear difference equation with a noise term added to it, is given by —

$$\left(1 + A(z^{-1})\right)R_t = \left(b_0 + B(z^{-1})\right)U_{t-h} + E_t \quad \ldots\ldots\ldots (3.3)$$

$$E_t = \frac{1 + C(z^{-1})}{1 + D(z^{-1})} W_t \quad \ldots\ldots\ldots\ldots\ldots\ldots (3.4)$$

where R is the system output, i.e., the residuals

U is the system input, a function of temperature[1]

[1] The determination of this function is described in reference [12] and will not be discussed in this report.

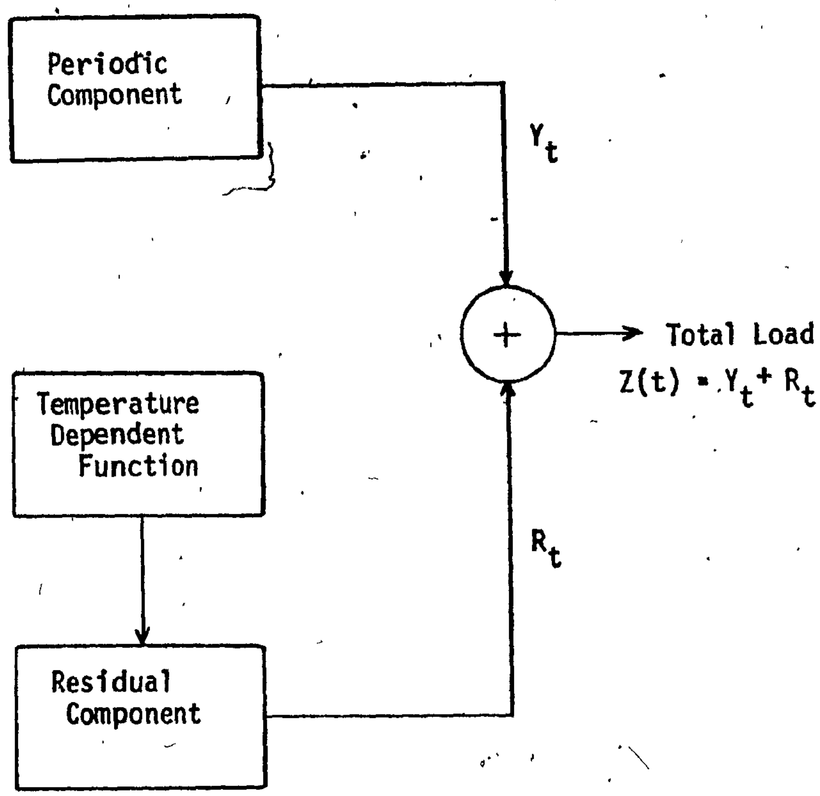


Fig. 3.5 — Block Diagram of the Load Model.

E is a correlated noise term

W is a zero mean gaussian process with variance σ_W^2, called white noise

h is a delay factor, assumed to be 7 hours

A, B, C, D are polynomials in the backward shift operator z^{-1} with coefficients a_i, b_i, c_i, d_i respectively, which are estimated from the data.

The representation of the preceding equations can be greatly simplified if we express them using Einstein notation: The occurrence of a superscript index with a variable and the same index with another variable indicates summation over that index [15]. Thus equation (3.2) can be rewritten as —

Let $\{p_i\} = \{\alpha_o, \alpha_k, \beta_k\}$

and $\{\gamma_{i,t}\} = (1, \{\sin kgt, \cos kgt\})$

where $k = 1, 2, 3 \ldots\ldots\ldots N'$

$i = 1, 2, 3 \ldots\ldots\ldots 2N'+1$

then $$Y_t = \sum_{i=1}^{2N'+1} p_i \gamma_{i,t} \quad \ldots\ldots (3.5)$$

Using Einstein's notation, we get —

$$Y_t = p^i \gamma_{i,t} \quad \ldots\ldots (3.6)$$

which indicates that the summation is over the index i. Similarly, equation (3.3) can be expressed as —

$$R_t + A(z^{-1})R_t = (b_o + B(z^{-1})U_{t-h} + E_t$$

$$R_t + \sum_{j=1}^{J} a_j R_{t-j} = \sum_{k=o}^{K} b_o U_{t-h-k} + E_t$$

where J, K are degrees of the polynomials $A(z^{-1})$ and $B(z^{-1})$ respectively.

Applying Einstein's notation we get —

$$R_t + a^j R_{t-j} = b^k U_{t-h-k} + E_t \quad \ldots\ldots\ldots\ldots (3.7)$$

Equation (3.4) can similarly be rewritten as —

$$E_t + \sum_{m=1}^{M} d_m E_{t-m} = W_t + \sum_{l=1}^{L} c_l W_{t-l}$$

where L, M are degrees of the polynomials $C(z^{-1})$, $D(z^{-1})$ respectively. From Einstein's notation we have —

$$E_t + d^m E_{t-m} = W_t + c^l W_{t-l}$$

so that

$$E_t = -d^m E_{t-m} + c^l W_{t-l} + W_t \quad \ldots\ldots\ldots\ldots (3.8)$$

Substituting equations (3.6), (3.7) and (3.8) into (3.1) we get the total load as —

$$Z_t = p^i Y_{i,t} - a^j R_{t-j} + b^k U_{t-h-k} - d^m E_{t-m} + c^l W_{t-l} + W_t \quad \ldots (3.9)$$

3.3.1 PREDICTION ALGORITHM

It now remains to be shown that equation (3.9) can be utilized to develop a prediction algorithm. Let $\hat{Z}_t(r)$ be the linear mean square error of prediction of Z_{t+r}, where r is the lead time. $\hat{Z}_t(r)$ can be computed as a conditional expectation [3] as follows:

$$\hat{Z}_t(r) = E[Z_{t+r} \mid Z_t, Z_{t-1}, Z_{t-2} \ldots\ldots]$$

Taking conditional expectance of both sides of equation (3.9) we get —

$$[Z_{t+r}] = p^{1}\gamma_{i,t+r} - a^{j}[R_{t+r-j}] + b^{k}U_{t+r-h-k} - d^{m}[E_{t+r-m}] + c^{l}(W_{t+r-l}] + \hat{[W}_{t+r}] \qquad \ldots\ldots\ldots\ldots (3.10)$$

where the square brackets in the expression denote conditional expectations. The functions $p^{1}\gamma_{i,t+r}$ and $U_{t+r-h-k}$ are not affected by the operation since these are deterministic terms that are assumed to be known for the lead time r. The remaining conditional expectations are evaluated as follows [3, 15].

For $n \geq 0$

$$
\begin{aligned}
[R_{t-n}] &= R_{t-n}, & n &= 0, 1, 2, 3, 4 \ldots\ldots \\
[R_{t+n}] &= \hat{R}_t(n), & n &= 1, 2, 3, 4 \ldots\ldots \\
[E_{t-n}] &= E_{t-n}, & n &= 0, 1, 2, 3 \ldots\ldots \\
[E_{t+n}] &= \hat{E}_t(n), & n &= 1, 2, 3, 4 \ldots\ldots \\
[W_{t-n}] &= Z_{t-n} - \hat{Z}_{t-n-1}(1), & n &= 0, 1, 2, 3 \ldots\ldots \\
[W_{t+n}] &= 0 & n &= 1, 2, 3, 4 \ldots\ldots
\end{aligned}
\qquad (3.11)
$$

The foregoing expressions indicate that the variables that are available at the time origin t are unchanged. Those that are not known can be replaced by their estimates at the origin.

3.4 Prediction Algorithm Based on the Residual Model

3.4.1 General Remarks

The prediction algorithm to be described in this section is based on Box & Jenkins [3] method. It utilizes the concept of seasonal differencing. The load data exhibits a periodic behaviour pattern every 24 hours.

The underlying principle of seasonal differencing is that the load values 24 hours apart are similar. The seasonal operator ∇_{24} when applied to the load data yields —

$$\nabla_{24}Z = Z_t - Z_{t-24} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (3.12)$$

The prediction algorithm is derived in the following section.

3.4.2 Prediction Algorithm

The differencing operator ∇_{24} is applied to equation (3.9) to get —

$$\nabla_{24} Z_t = \nabla_{24} p^i Y_{i,t} - \nabla_{24} a^j R_{t-j} + \nabla_{24} b^k U_{t-h-k} - \nabla_{24} d^m E_{t-m}$$

$$+ \nabla_{24} c^l W_{t-l} + \nabla_{24} W_t \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (3.13)$$

Since the first term on the R.H.S. of equation (3.9) is periodic, the first term in equation (3.13) vanishes, giving —

$$Z_t - Z_{t-24} = -\nabla_{24} a^j R_{t-j} + \nabla_{24} b^k U_{t-h-k} - \nabla_{24} d^m E_{t-m} + \nabla_{24} c^l W_{t-l} + \nabla_{24} W_t$$

$$\therefore \; Z_t = Z_{t-24} - \nabla_{24} a^j R_{t-j} + \nabla_{24} b^k U_{t-h-k} - \nabla_{24} d^m E_{t-m} + \nabla_{24} c^l W_{t-l} + \nabla_{24} W_t$$

$$\ldots\ldots\ldots\ldots \quad (3.14)$$

The predictions can be computed using the ideas developed in Section 3.3.1, by taking the conditional expectation of Z_{t+r} at time t as —

$$[Z_{t+r}] = [Z_{t+r-24}] - [\nabla_{24} a^j R_{t+r-j}] + [\nabla_{24} b^k U_{t+r-h-k}] - [\nabla_{24} d^m E_{t+r-m}]$$

$$+ [\nabla_{24} c^l W_{t+r-l}] + [\nabla_{24} W_{t+r}] \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (3.15)$$

All the terms of equation (3.15) can be evaluated using equation (3.11). Note, however, that —

$$[Z_{t+r-24}] = Z_{t+r-24}$$

the past values of the load are known. Also, the conditional expectation operator does not affect the third term on R.H.S. of equation (3.15) since U_t, the temperature dependent function is known for all t.

3.4.3 CONCLUDING REMARKS

The prediction algorithms described in the foregoing sections were selected on the basis of their computational simplicity. Each of the algorithms is algebraic in nature. The predictions are obtained recursively using each of the algorithms. In the periodic and residual model algorithm, the knowledge of the periodic component of the load and its associated parameters is required. For the residual model however, these parameters are not required.

CHAPTER 4

RESULTS OF PREDICTION

4.1 GENERAL

In order to implement the selected prediction algorithms of Chapter 3 on a minicomputer, two Fortran programs were devised. The first program uses the prediction of algorithm developed in section 3.3 and the second uses the algorithm developed in section 3.4. The program source listings are presented in Appendix A. The data used in this report is taken from McIntosh [12]. It consists of the Hydro-Quebec load data for January 1972, along with the corresponding temperature data provided by the Dorval Weather Bureau. The temperature dependent function U_t defined in Chapter 3, equation 3.3, is also taken from reference [12]. The relevant data is tabulated in Appendix B. The model parameters were determined by Panuska and Koutchouk [16]. Appendix C contains the model parameters in a tabulated form. The line printer outputs obtained from each of the algorithms is also contained in this Appendix. Appendix D contains the relevant mathematical formulae used to compute the error summary and the probabilty limits.

4.2 DETAILS OF THE RESULTS

For both algorithms, January 20, 1972, was arbitrarily chosen as the day of prediction. The time origin was placed at midnight of January 19.

The lead time of prediction is 24 hours: The results of the periodic and residual component model are presented first. The periodic component was computed over three cycles and is shown in Table 4.1. The residual component is shown in Table 4.2. Table 4.3 shows the total predicted load and its percent deviations from the actual load. Figures 4.1 and 4.2 show the time plots of the periodic and residual components respectively. Figure 4.3 indicates the variation of the temperature dependent function over the lead time of prediction. Table 4.4 shows the error statistics obtained using the periodic and residual component model. Note that the quantities indicated in the table are in 10MW units. Figure 4.4 shows both the predicted and actual load.

The results obtained from the Box and Jenkins [3] method using the residual model are presented next. The predicted and actual load are shown in Table 4.5. This table also indicates the percent deviations of the predicted load from their actual values.

Figure 4.5 shows the time plots of the predicted and actual load over the lead time of prediction. Table 4.6 shows the error figures obtained using the residual model. Figure 4.6 shows the ± 0.95 probability limit envelope. It should be noted that these limits apply to individual predictions and not jointly to the overall prediction.

Table 4.7 shows the memory maps for each of the algorithms.

4.3 CONCLUDING REMARKS

The predictions obtained from either algorithm show roughly the same error figures as indicated by Tables 4.4 and 4.6. The rms error as a percentage of the peak load of 8790 MW was less than 4% in both cases.

PERIODIC COMPONENT OF LOAD (IN 10MW UNITS)

HOUR:	1	2	3	4	5	6	7	8	9	10	11	12
PERIODIC COMP.:	662	637	640	612	593	612	634	668	720	761	797	810
HOUR:	13	14	15	16	17	18	19	20	21	22	23	24
PERIODIC COMP.:	781	770	774	748	771	856	875	825	798	791	771	725
HOUR:	25	26	27	28	29	30	31	32	33	34	35	36
PERIODIC COMP.:	662	637	640	612	593	612	634	668	720	761	797	810
HOUR:	37	38	39	40	41	42	43	44	45	46	47	48
PERIODIC COMP.:	781	770	774	748	771	856	875	825	798	791	771	725
HOUR:	49	50	51	52	53	54	55	56	57	58	59	60
PERIODIC COMP.:	662	637	640	612	593	612	634	668	720	761	797	810
HOUR:	61	62	63	64	65	66	67	68	69	70	71	72
PERIODIC COMP.:	781	770	774	748	771	856	875	825	798	791	771	725

Table 4.1 - Periodic Component.

RESIDUAL COMPONENT OF LOAD (IN 10MW UNITS)

HOUR:	1	2	3	4	5	6	7	8	9	10	11	12
RESIDUAL COMP.:	7	6	4	3	9	5	12	9	1	-6	-10	-3
HOUR:	13	14	15	16	17	18	19	20	21	22	23	24
RESIDUAL COMP.:	26	31	20	12	11	18	14	7	2	8	6	6
HOUR:	25	26	27	28	29	30	31	32	33	34	35	36
RESIDUAL COMP.:	-3	8	14	8	5	0	0	-4	-5	-4	-11	3

Table 4.2 — Residual Component.

LOAD PREDICTION USING PERIODIC & RESIDUAL COMPONENTS
HYDRO-QUEBEC LOAD DATA FOR JANUARY 20, 1972 (IN 10MW UNITS)

HOUR:	1	2	3	4	5	6	7	8	9	10	11	12
PREDICTED:	669	643	644	615	602	617	646	677	721	754	786	806
ACTUAL:	637	626	616	607	609	624	684	733	762	778	803	804
% ERROR:	-5	-2	-4	-1	1	1	5	7	5	3	2	0

HOUR:	13	14	15	16	17	18	19	20	21	22	23	24
PREDICTED:	807	801	794	760	782	874	889	832	800	799	777	731
ACTUAL:	776	782	773	796	879	876	852	834	820	788	747	692
% ERROR:	-3	-2	-2	4	11	0	-4	0	2	-1	-4	-5

Table 4.3 — Load Prediction Using Periodic and Residual Components.

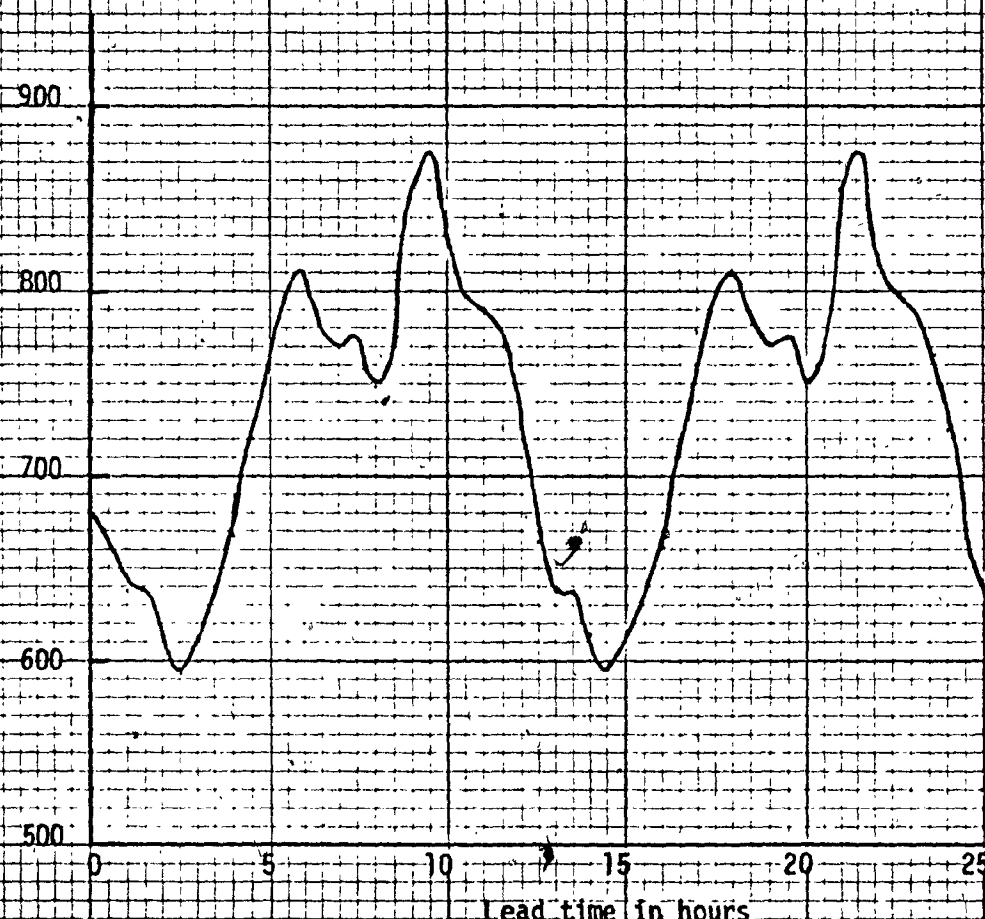


Figure 4.1 — Periodic Component (in 10MW units)

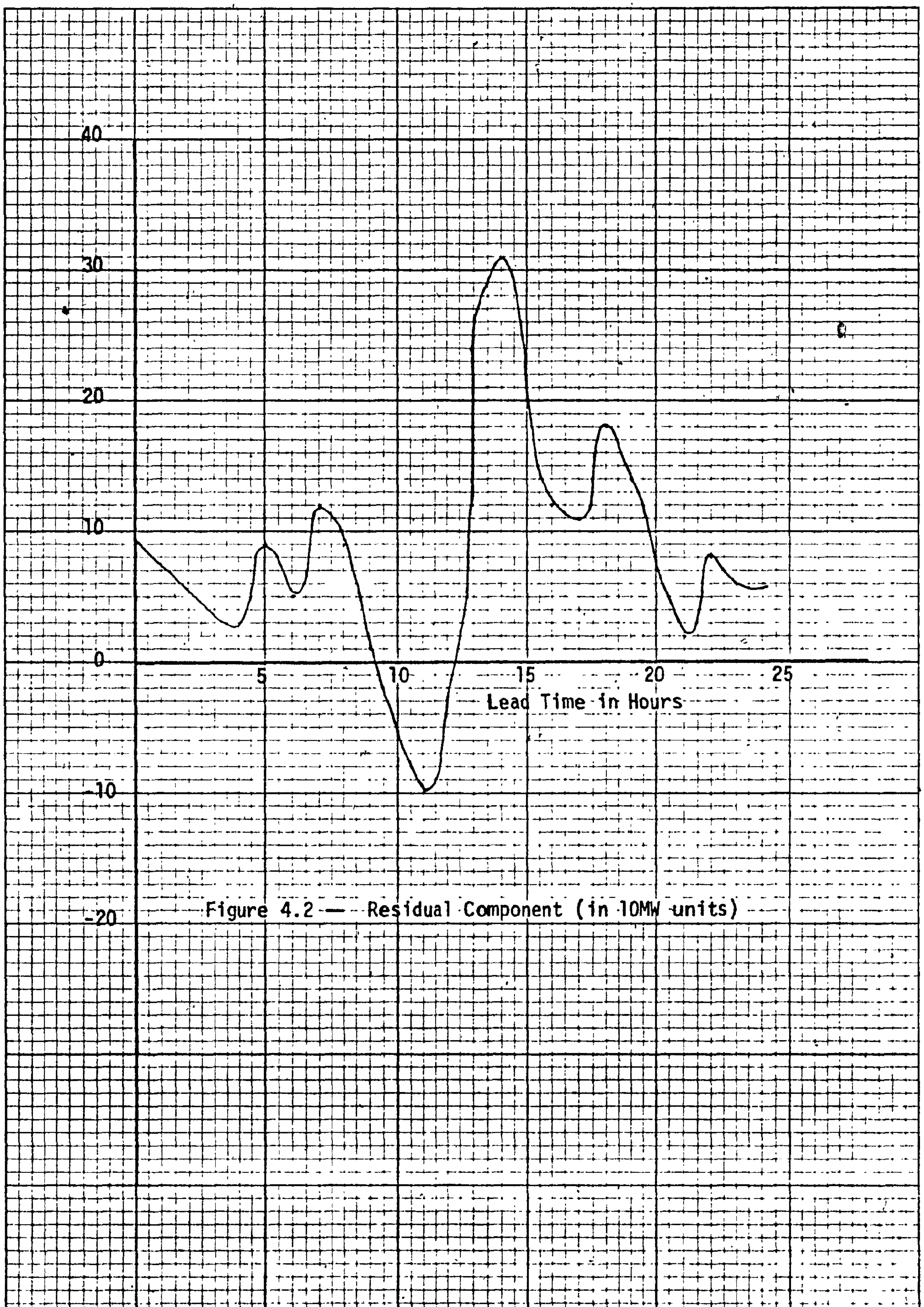


Figure 4.2 — Residual Component (in 10MW units)

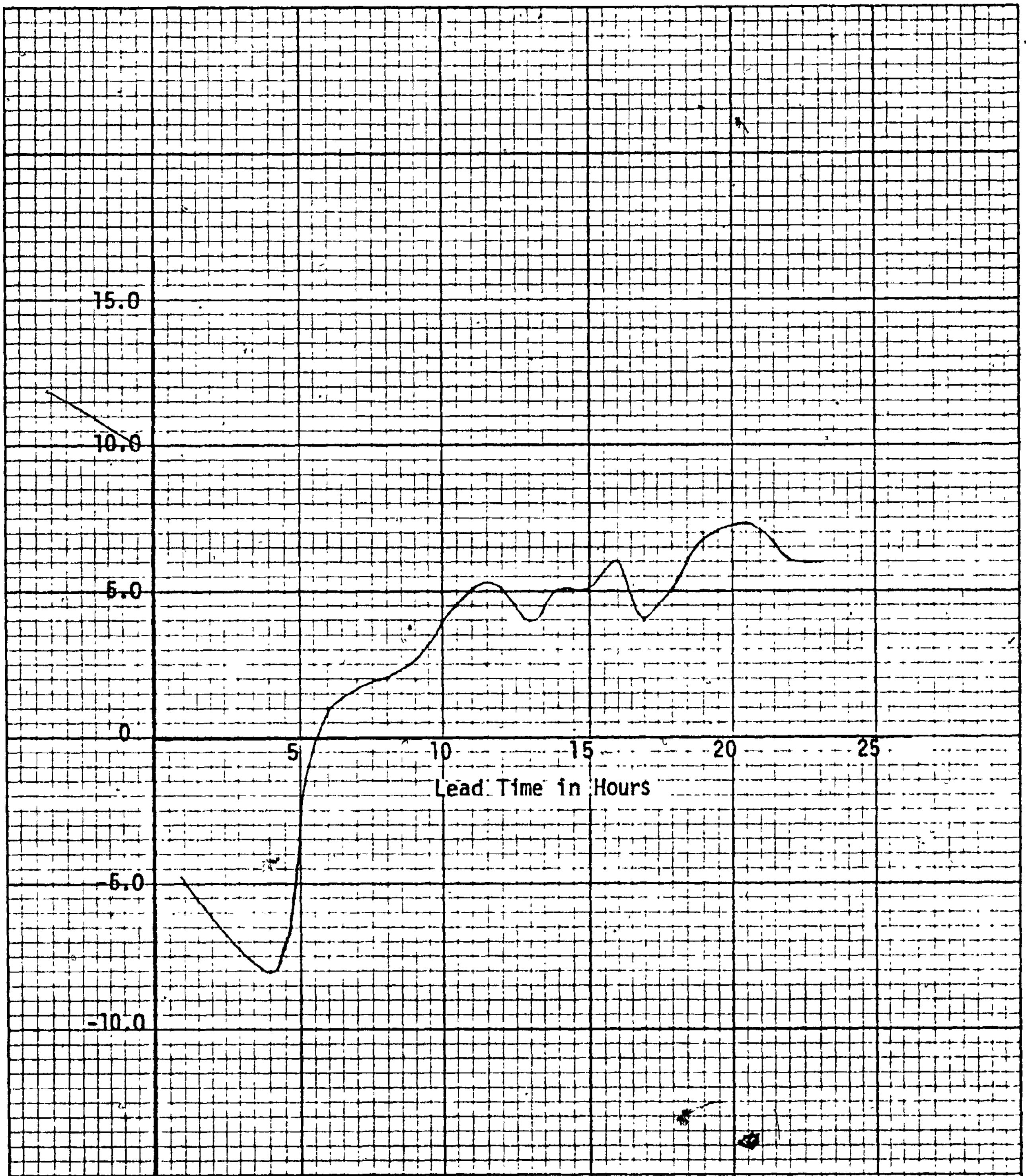


Figure 4.3 — Temperature Dependent Function U(T) (in °F)

```
              ERROR SUMMARY FOR JANUARY 20,1972 LOAD PREDICTION
    MEAN ERROR OF PREDICTION=25,92
     ACTUAL STANDARD DEVIATION=20,91
          RMS PREDICTION ERROR= 33,74
RMS ERROR PERCENT OF PEAK LOAD= 3,84
```

Table 4.4 — Results Based on Periodic and Residual Component Model.

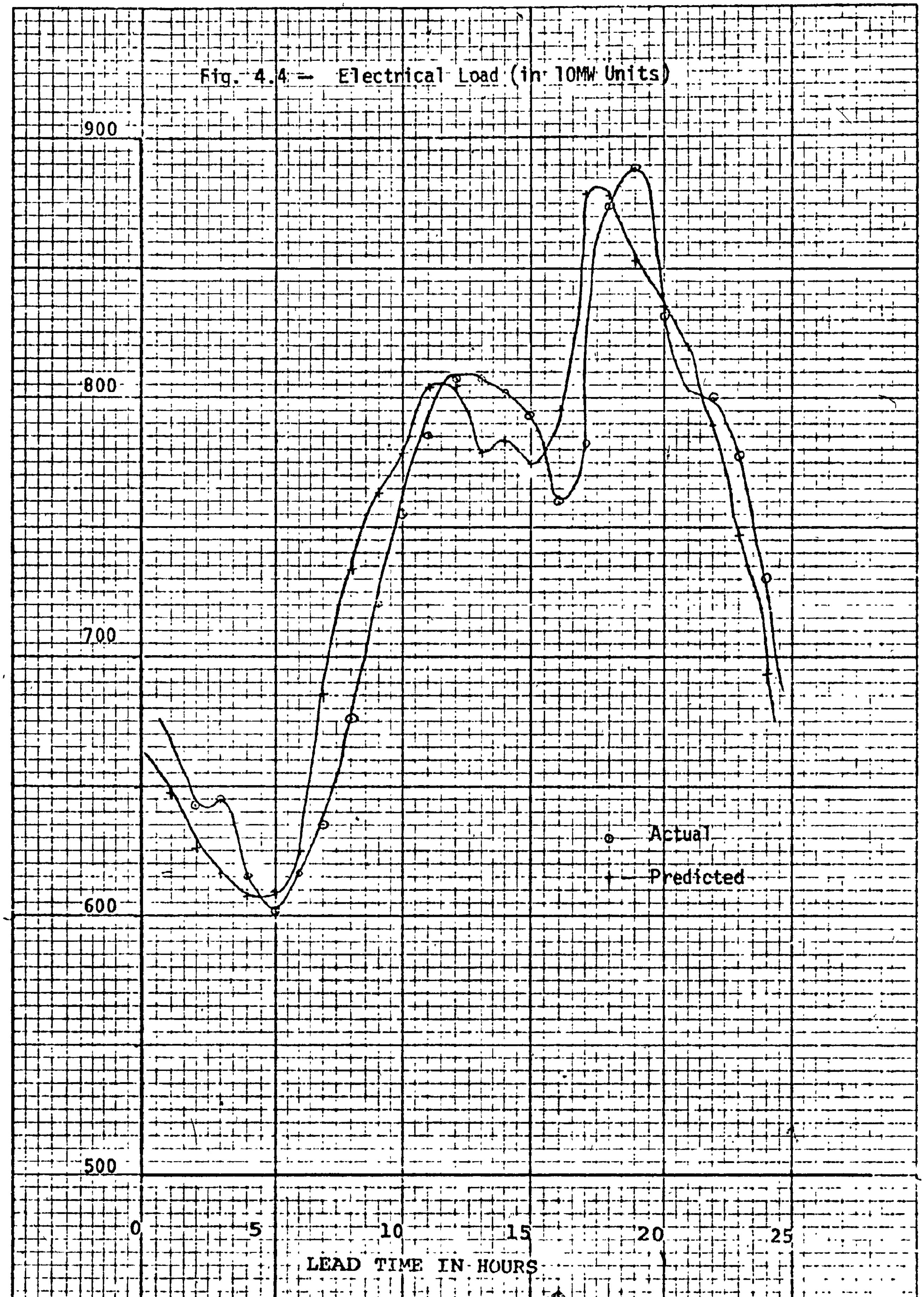


Fig. 4.4 — Electrical Load (in 10MW Units)
900
800
700
600
500
0
5
10
15
20
25
LEAD TIME IN HOURS
Actual
Predicted

LOAD PREDICTION USING BOX & JENKINS METHOD

HYDRO-QUEBEC LOAD DATA FOR JANUARY 20,1972 (IN 10MW UNITS)

HOUR:	1	2	3	4	5	6	7	8	9	10	11	12
PREDICTED:	646	636	631	626	639	644	712	810	786	779	794	780
ACTUAL:	637	626	616	607	609	624	684	733	762	778	803	804
% ERROR:	1	1	2	3	4	3	4	10	3	0	-1	-2

HOUR:	13	14	15	16	17	18	19	20	21	22	23	24
PREDICTED:	778	795	761	805	896	845	807	795	759	755	724	0
ACTUAL:	776	782	773	796	879	876	852	834	820	788	747	692
% ERROR:	0	1	-1	1	1	-3	-5	-4	-7	-4	-3	0

Table 4.5 — Load Prediction Using Box & Jenkins Method.

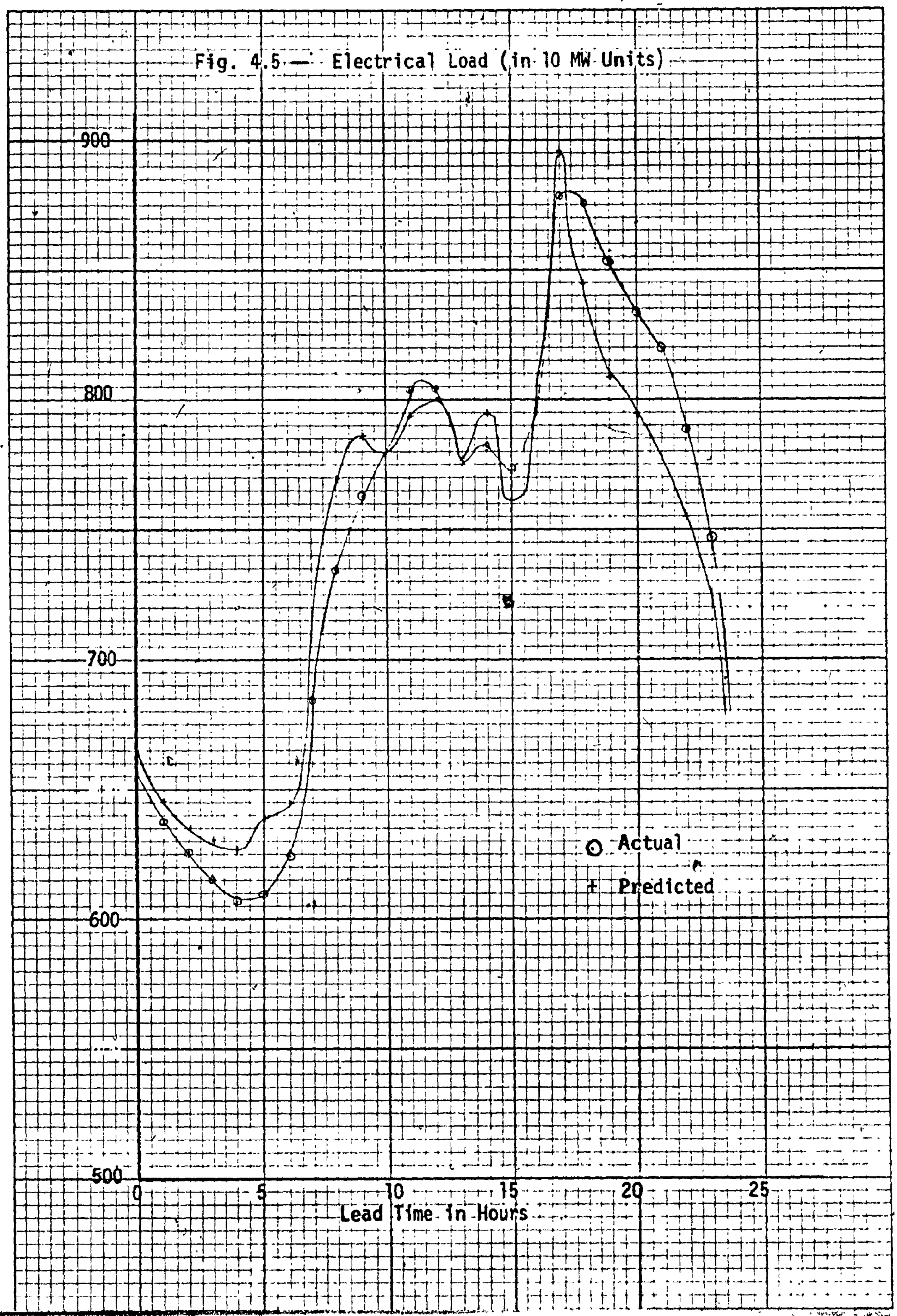


Fig. 4.5 — Electrical Load (in 10 MW Units)
900
800
700
600
500
0
5
10
15
20
25
Lead Time in Hours
Actual
Predicted

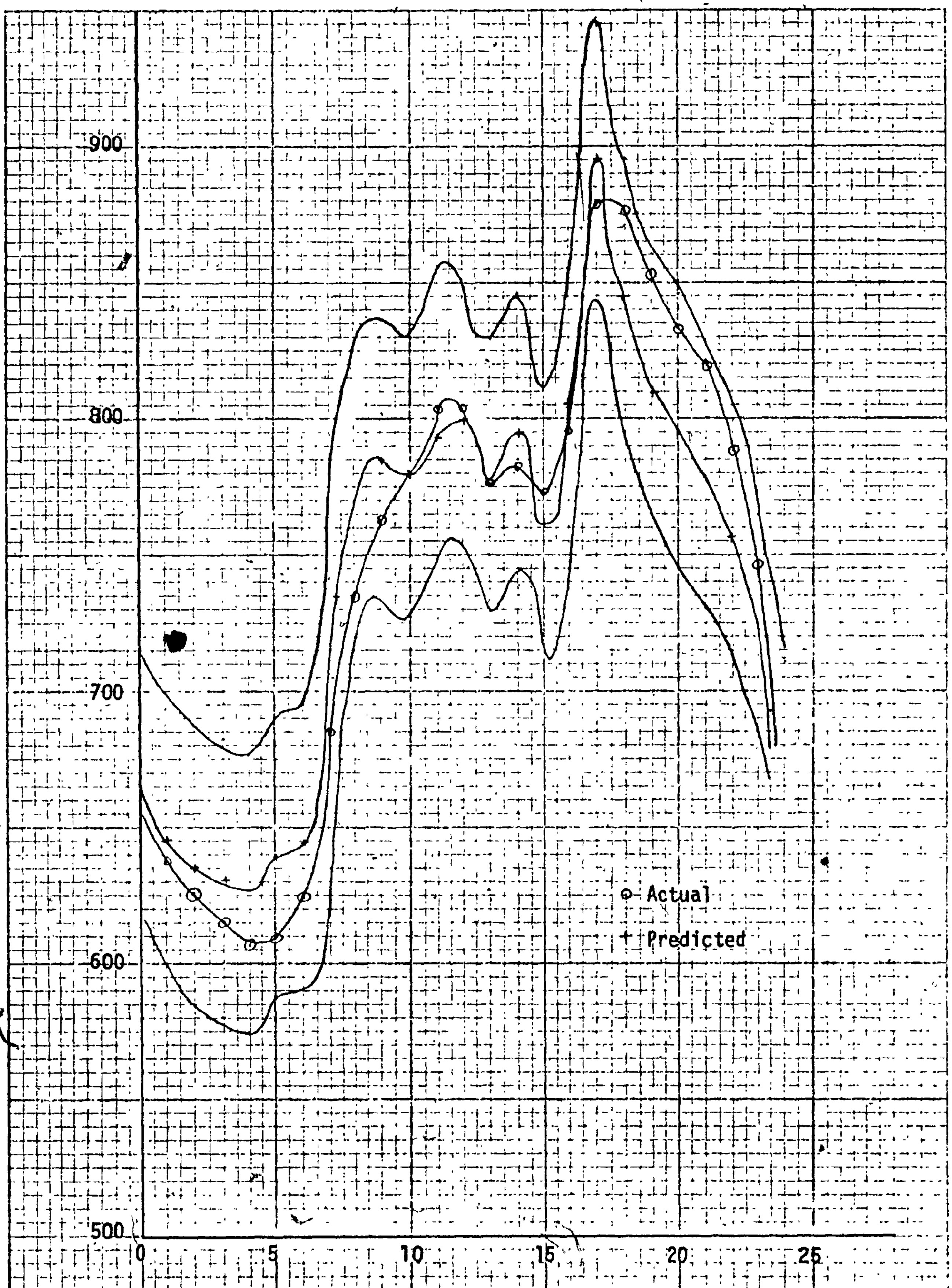


Figure 4.6 — Electrical Load (in 10MW units).

```
ERROR SUMMARY FOR JANUARY 20,1972 LOAD PREDICTION

      MEAN ERROR OF PREDICTION= 23.96
     ACTUAL STANDARD DEVIATION= 18.27
          RMS PREDICTION ERROR= 30.56
RMS ERROR PERCENT OF PEAK LOAD= 3.48
```

Table 4.6 — Results Based on Residual Model.

Memory Map PREDC1.LDA

Transfer Address*	113724
Low Limit	113724
High Limit	157460

(a)

Memory Map PREDC2.LDA

Transfer Address	115324
Low Limit	115324
High Limit	157460

(b)

*All addresses on this page are in octal, base 8.

Table 4.7 — a) Periodic & Residual Model (File Name: PREDC1)

b) Residual Model (File Name: PREDC2).

CHAPTER 5

CONCLUSIONS

The aims of this report have been met in that it has been successfully demonstrated that it is indeed feasible to implement selected prediction algorithms on micro and minicomputers. This is clearly shown by the memory maps (Table 4.7) of the two algorithms programmed in Fortran on a 16-bit minicomputer. Each of the algorithms require approximately 8K words of memory storage. Further reductions in storage requirement are possible by utilizing Assembly language. The compilation time for each of the algorithms is about 10 minutes. This includes the printing of the program listing on a 300 lpm line printer. The execution time is about 20 seconds.

The overall accuracy of the prediction is within acceptable limits [15].

REFERENCES

1. ALTMAN, L. and SCRUPSKI, S.E., "Applying Microprocessors", Electronics Magazine Book Series, McGraw Hill, New York, 1976.

2. BOX, G.E.P. and JENKINS, G.M., "Some Recent Advances in Forecasting and Control", Part I, Applied Statistics, Vol. 17, pp. 91-102, 1968.

3. BOX, G.E.P., and JENKINS, G.M., "Time Series Analysis - Forecast and Control", Holden-Day, San Francisco, 1970.

4. "Canadian Electronics Engineering", pp. 19-24, Maclean-Hunter Publication, Toronto, February 1978.

5. CAPECE, R.P., Electronics Magazine, pp. 96-107, McGraw Hill, New York, October 27, 1977.

6. CHATFIELD, C., "The Analysis of Time-Series: Theory and Practice", Chapman and Hall, London, 1975.

7. CHRISTIAANSE, W.R., "Short Term Load Forecasting Using General Exponential Smoothing", Paper No. 70, TP 681-PWR, IEEE Summer Power Meeting and EHV Conference, Los Angeles, California, July 1970.

8. GALIANA, F.D., HANDSCHIN, E., and FLETCHER, A.E., "Identification of Stochastic Electric Load Models from Physical Data", IEEE Transactions on Automatic Control, Vol. AC-19, pp. 887-893, 1974.

9. GUPTA, P.C., 'A Stochastic Approach to Peak Power Demand Forecasting in Electrical Utility Systems", IEEE Transaction on Power Apparatus and Systems, Vol. PAS-90, pp. 824-832, 1971.

10. LaFARRA, R., "Computer Methods in Science and Engineering", Hyden Book Company Inc., New Jersey, 1973.

11. LIJESEN, D.P. and ROSING, J., "Adaptive Forecasting of Hourly Loads Based on Load Measurements and Weather Information", IEEE Transactions of Power Apparatus and Systems, Vol. PAS-90, pp. 1757-1767, 1971.

12. McINTOSH, D.G., "Short Term Load Forecasting in an Electrical Power System", A Major Technical Report in the Faculty of Engineering, Concordia University, Montreal, 1973.

13. MEDITCH, J.S., "Stochastic Optimal Linear Estimation and Control", McGraw Hill, N.Y., 1969.

14. MORRISON, N., "Introduction to Sequential Smoothing and Prediction", McGraw Hill, N.Y. 1969.

15. PANUSKA, V., "Short-Term Forecasting of Electric Power System Load from a Weather Dependent Model", IFAC Symposium, The Institute of Engineers of Australia, Melbourne, 1977.

16. PANUSKA, V. and KOUTCHOUK, J.P., "Electrical Power System Load Modelling of a Two Stage Stochastic Approximation Procedure", Preprints, 6th Int. Fed. Automatic Control World Congress, paper No. 31.4, Boston, Mass, 1975.

17. SOLINAS, J., "Needs, Application and Techniques of Load Forecasting Applied to Operating and Controlling a Power Utility", A Major Technical Report in the Faculty of Engineering, Concordia University, Montreal, 1973.

18. SRINIVASAN, K. and PRONOVOST, R., "Short-Term Load Forecasting Using Multiple Correlation Models", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-94, pp. 1854-1858, 1975.

19. STAVROULAKIS, N.C., "Hybrid Computer Simulation of a Model for Electrical Load Short Term Forecasting", A Major Technical Report in the Faculty of Engineering, Concordia University, Montreal, 1974.

20. VEMURI, S., HILL, E.F. and BALASUBRAMANIAN, R., "Load Forecasting Using Stochastic Models", 8th Power Industry Application Conference, Minneapolis, Minn., 1973.

APPENDIX A

```
        FORTRAN V004A   00:01:38    06-FEB-78    PAGE    1
C
C       LOAD PREDICTION ALGORITHM BASED ON PERIODIC & RESIDUAL METHOD
C       DESCRIBED IN -- PANUSKA, V., 'SHORT-TERM FORECASTING OF
C       ELECTRICAL POWER LOAD FROM A WEATHER-DEPENDENT MODEL'
C
C
C       THE PROGRAM WAS EXECUTED ON A DISK BASED PDP11/15 MINICOMPUTER
C       SYSTEM USING DIGITAL EQUIPMENT 'S FORTRAN-4-VERSION A
C
        DIMENSION NU(72),AL(8),BE(8),Y(72)
        DIMENSION  NPLOT(100),X(72)
        DIMENSION Z(72),IHR(72),IER(72)
        DIMENSION U(80),T(72)
        DIMENSION F1(24),V(24)
        INTEGER  Z,Y,U,IER
C
C       DATA DEFINITIONS
C
C       TEMPERATURE DATA
C       U(1) = U(T-9), U(2) = U(T-8).... ETC
C
C       ACTUAL LOAD DATA
C       Z(1) = Z(T+1), Z(2) = Z(T+2)....ETC
C
        DATA U/-13,-11,-11,-10,-10,-8,-8,-6,-5,-5,-6,-7,-8,-3,1,2,2,2,4
       1,5,5,4,5,5,6,4,5,7,7,7,6,6,5,4,4,2,2,7,7,5,10,10,12,14,15,14
       1,14,12,12,13,13,16
       1,14,15,15,18,20,15,17,16,16,15,17,18,19,19,18,17,14,12,10,5,3,-5
       1,-14,-15,-15,-14,-14,-14/
        DATA Z/637,626,616,607,609,624,684,733,762,778,803,804,776,782
       1,773,796,879,876,852,834,820,788,747,692,652,637,626,616,621,
       1633,681,747,780,794,810,799,761,755,758,768,859,867,856,832,811
       1,779,752,660,628,616,606,604,604,633,685,733,779,797,821,808,77[illegible]
       1,778,777,794,876,863,846,839,819,791,755,708/
C
C       MODEL PARAMETERS
C
        READ(6,100)A1
100     FORMAT(F5.2)
        READ(6,110)A2
110     FORMAT(F6.3)
        READ(6,110)A3
        READ(6,120)B0
120     FORMAT(F4.2)
        READ(6,120)B1
        READ(6,100)B2
        BESUM=0
        NXF=0
        NSM=0
C
C
C       COEFFICIENTS
C.      VALUE OF ALPHA ZERO
```

```
      AL0=7351.0
      AL(1)=-965.6
      AL(2)=-398.8
      AL(3)=134.0
      AL(4)=4.18
      AL(5)=-101.2
      AL(6)=-137.7
      AL(7)=19.7
      AL(8)=21.7
      BE(1)=-273.1
      BE(2)=199.4
      BE(3)=-178.7
      BE(4)=90.8
      BE(5)=65.2
      BE(6)=-32.6
      BE(7)=-40.0
      BE(8)=74.1
C     LOAD AT THE ORIGIN
      Z0=6640
C     COMPONENTS OF THE PERIODIC TERM
      C1Y=AL0
      NT=1
55    C2Y=0.0
      C3Y=0.0
      DO 60 I=1,8
      ARG=I*2*3.14016*NT/24
      C2Y=C2Y+AL(I)*SIN(ARG)
      C3Y=C3Y+BE(I)*COS(ARG)
60    CONTINUE
      Y(NT)=C1Y+C2Y+C3Y
C     SCALING FACTOR = 10
      NSCAL=10
      Y(NT)=Y(NT)/NSCAL
      NT=NT+1
      IF(NT.LE.72)GOTO 55
C     WRITE HOURS IN ARRAY IHR
      DO 520 I=1,72
520   IHR(I)=I
      JJ=1
      JK=12
      WRITE(5,550)
550   FORMAT('1')
      WRITE(5,555)
555   FORMAT(35X,'PERIODIC COMPONENT OF LOAD (IN 10MW UNITS)'//)
802   WRITE(5,430)(IHR(M),M=JJ,JK)
      WRITE(5,560)(Y(M),M=JJ,JK)
560   FORMAT(25X,'PERIODIC COMP.:',12I4/)
      JJ=JJ+12
      JK=JK+12
      IF(JK.GE.84)GOTO 801
      GOTO 802
801   CALL PLOT(NPLOT,Z,Y,NSM)
C
C
C
C
C
```

```
C     COMPUTAION OF THE RESIDUAL TERM X(T)
C
C
C
C     VALUES OF PAST LOAD AT T=-1,-2,-3
      ZM1=7150
      ZM2=7520
      ZM3=7700
C
C     PAST VALUES OF X AT T= -1,-2,-3
      XM1=(ZM1/NSCAL)-Y(23)
      XM2=(ZM2/NSCAL)-Y(22)
      XM3=(ZM3/NSCAL)-Y(21)
      DO 65 I=1,8
65    BESUM=BESUM+BE(I)
      X0=A1*XM1+A2*XM2+A3*XM3
      X0=X0+B0*U(3)+B1*U(2)+B2*U(1)
      X(1)=-A1*X0-A2*XM1-A3*XM2+B0*U(4)+B1*U(3)+B2*U(2)
      X(2)=-A1*X(1)-A2*X0-A3*XM1+B0*U(5)+B1*U(4)+B2*U(3)
      X(3)=-A1*X(2)-A2*X(1)-A3*X0+B0*U(6)+B1*U(5)+B2*U(4)
      DO 185 K=4,72
      X(K)=-A1*X(K-1)-A2*X(K-2)-A3*X(K-3)+B0*U(K+2)+B1*U(K+1)+B2*U(K)
185   CONTINUE
C
C     PRINT X(T)
      JJ=1
      JK=12
      DO 552 I=1,48
552   NU(I)=X(I)
      WRITE(5,550)
      WRITE(5,600)
600   FORMAT(35X,'RESIDUAL COMPONENT OF LOAD (IN 10MW UNITS)'//)
602   WRITE(5,605)(IHR(M),M=JJ,JK)
605   FORMAT(35X,'HOUR:',12I6)
      WRITE(5,610)(NU(M),M=JJ,JK)
610   FORMAT(25X,'RESIDUAL COMP.:',12I6/)
      JJ=JJ+12
      JK=JK+12
      IF(JK.GE.48)GOTO 601
      GOTO 602
C
C     SAVE Y
601   DO 505 I=1,48
505   T(I)=Y(I)
C
C     PLOT X(T)
      DO 395 I=1,48
395   Y(I)=X(I)
      CALL PLOT(NPLOT,Z,Y,NSM)
C     TOTAL LOAD = PERIODIC + RESIDUAL COMPONENETS
402   DO 396 I=1,48
396   Y(I)=T(I)+X(I)
C     CALCULATE PERCENT ERROR
      DO 540 I=1,24
      NERR=Z(I)-Y(I)
540   IER(I)=(100*NERR)/Z(I)
```

```
C       STEP TO NEW PAGE
        WRITE(5,530)
530     FORMAT('1')
        WRITE(5,410)
410     FORMAT(35X,'LOAD PREDICTION USING PERIODIC & RESIDUAL
       1 COMPONENTS ')
        WRITE(5,420)
420     FORMAT(35X,'HYDRO-QUEBEC LOAD DATA FOR JANUARY 20, 1972
       1 (IN 10MW UNITS)'//)
C       PRINT DATA
        M1=1
        M2=12
82      WRITE(5,430)(IHR(M),M=M1,M2)
430     FORMAT(35X,'HOUR:',12I4)
        WRITE(5,440)(Y(M),M=M1,M2)
440     FORMAT(30X,'PREDICTED:',12I4)
        WRITE(5,450)(Z(M),M=M1,M2)
450     FORMAT(33X,'ACTUAL:',12I4)
        WRITE(5,465)(IER(M),M=M1,M2)
465     FORMAT(32X,'% ERROR:',12I4/)
        M1=M1+12
        M2=M2+12
        IF(M2.GE.36)GOTO 81
        GOTO 82
81      WRITE(5,470)
470     FORMAT(X,///)
        NSM=1
        CALL PLOT(NPLOT,Z,Y,NSM)
C
C       LEAD TIME OF PREDICTION
        LEAD=24
C
C       INITIALISE MEAN ERROR OF FORECAST
        XMEAN=0
C
        DO 501 I=1,LEAD
        MIN=IABS(Y(I)-Z(I))
        XMEAN=MIN+XMEAN
501     CONTINUE
        XMEAN=XMEAN/LEAD
C       ACTUAL STANDARD DEVIATION
        FSTD=0
        SSTD=0
        DO 502 I=1,LEAD
        STD=(Y(I)-Z(I))**2
        FSTD=FSTD+STD
        NFST=IABS(Y(I)-Z(I))
502     SSTD=SSTD+NFST
        SSTD=SSTD**2/(LEAD)
        SSTD=ABS(FSTD-SSTD)
        SSTD=SSTD/(LEAD-1)
        SSTD=SQRT(SSTD)
C       RMS ERROR CALCULATIONS
        RMS=SSTD**2+(XMEAN**2)*LEAD/(LEAD-1)
        RMS=SQRT(RMS)
C       PEAK LOAD
```

```
          PEAKLD=879.0
          PERCNT=(RMS*100)/PEAKLD
          WRITE(5,514)
  514     FORMAT('1')
          WRITE(5,504)
  504     FORMAT(30X,'ERROR SUMMARY FOR JANUARY 20,1972 LOAD PREDICTION'/)
          WRITE(5,506)XMEAN
  506     FORMAT(20X,'MEAN ERROR OF PREDICTION=',F5.2/)
          WRITE(5,508)SSTD
  508     FORMAT(19X,'ACTUAL STANDARD DEVIATION=',F5.2/)
          WRITE(5,510)RMS
  510     FORMAT(24X,'RMS PREDICTION ERROR=',F6.2/)
          WRITE(5,512)PERCNT
  512     FORMAT(14X,'RMS ERROR PERCENT OF PEAK LOAD=',F5.2/)
  700     H=T
          WRITE(5,300)
  300     FORMAT('1')
  C
  C       PROBABILITY LIMITS
  C
  C       VALUES OF PHI
          FI0=1
          FI(1)=A1
  C       ESTIMATED VARIANCE=
          VAR=104.1
          FI(2)=A1*FI(1)+A2
          FI(3)=A1*FI(2)+A2*FI(1)+A3
          DO 200 I=4,24
  200     FI(I)=A1*FI(I-1)+A2*FI(I-2)+A3*FI(I-3)
          DO 210 I=1,24
          WRITE(5,205)FI(I),I
  205     FORMAT(10X,F8.2,I6)
  210     CONTINUE
  C
  C       VARIANCE OF PREDICTION
  C
          V(1)=(1+FI(1)**2)*VAR
          DO 300 I=2,24
          V(I)=V(1)+(FI(I)**2)*VAR
          WRITE(5,630)V(I)
  630     FORMAT(10X,F8.2)
  300     CONTINUE
  405     STOP
          END

  ROUTINES CALLED:
  SIN    , COS    , PLOT   , IABS   , ABS    , SQRT

  BLOCK        LENGTH
  MAIN.     4225    (020402)*
)14 000000G.300:    $SEQ,000273

  **COMPILER ----- CORE**
      PHASE       USED  FREE
  DECLARATIVES  00527 08609
  EXECUTABLES   01407 07729
  ASSEMBLY      02501 09552
```

```
               FORTRAN V004A   00:03:37     06-FEB-78    PAGE     1
          SUBROUTINE PLOT(NPLOT,Z,Y,NSM)
          DIMENSION NPLOT(100),Y(72),Z(72)
          INTEGER Z,Y,U
C          PLOTTING ROUTINE
C         HEAD OF FORM
          WRITE(5,135)
135       FORMAT('1')
C         PUT ****S INTO ARRAY
139       DO 140 K=1,100
140       NPLOT(K)='*'
          WRITE(5,150)NPLOT
150       FORMAT(21X,100A1)
C         BLANK THE ARRAY
          DO  180  K=1,100
180       NPLOT(K)=' '
          N=1
99        J=Y(N)/10
          J=IABS(J)
          NK=Z(N)/10
          IF(J.LE.0)J=1
          IF(NSM.EQ.1)NPLOT(NK)='A'
          NPLOT(J)='+'
          NPLOT(1)='*'
          WRITE(5,101)NPLOT
101       FORMAT(20X,100A1)
          NPLOT(J)=' '
          NPLOT(NK)=' '
          N=N+1
          IF(N.NE.26)GOTO 99
          RETURN
          END
ROUTINES CALLED:
IABS

BLOCK        LENGTH
PLOT    328     (001220)*

**COMPILER ----- CORE**
   PHASE        USED  FREE
DECLARATIVES  00446  08690
EXECUTABLES   00607  08529
ASSEMBLY      01161  10892
```

```
      FORTRAN V004A   00:26:42    06-FEB-78   PAGE  1
C
C     LOAD PREDICTION ALGORITHM BASED ON BOX & JENKINS METHOD
C
C
      DIMENSION X(26),Z(30),U(100),T(100),W(75),WW(75)
      DIMENSION NPLOT(100),A(76),Z1(75),IER(75),IHR(75)
      DIMENSION NN(24),NU(24)
      INTEGER Z,Z0,WW,T
      INTEGER X,U,A,Z1,W,IER
C
C
C     DEFINITION OF DATA ELEMENTS
C
C     ARRAY A DEFINES THE ACTUAL LOAD DATA FOR JANUARY 20, 1972
C
C     PAST VALUES OF LOAD AND PREDICTED LOAD
C
C     Z(T-26) = X(1) , Z(T-25) = X(2) .... AND SO ON
C
C     TEMPERATURE FUNCTION  U(T)
C
C     U AT T-32 = T(1), U AT T031 = T(2) ..... AND SO ON
C
C     ORIGIN IS HOUR ZERO FOR ANY GIVEN DAY
C
      DATA A/637,626,616,607,609,624,684,733,762,778,803,804,776
     1,782,773,796,879,876,852,834,820,788,747,692,652,637.626,616
     1,621,633,681,747,780,794,810,799,761,755,758,768,859,867,856,83
     1,811,779,752,660,628,616,606,604,604,633,685,733,779,797,821,
     1808,772,778,777,794,876,863,846,839,819,791,755,708,684,657,648
     1,651/
      DATA X/729,693,647,616,603,592,579,589,607,651,701,738,771
     1,786,776,749,746,740,754,843,839,819,795,770,752,715/
      DATA T/-20,-25,-25,-22,-23,-24,-25,-26,-26,-25,-27,-33,-34,-33
     1,-32,-31,-31,-31,-24,-24,-22,-20,-15,-13,-11,-11,-11,-10,-10,-8
     1,-8,-6,-5,-5,-6,-7,-8,-3,1,2,2,2,4,5,5,4,5,5,6,4,5,7,7,7,6,6,5,
     1,3,2,2,7,7,5,7,10,12,14,15,14,14,12,13,13,16,14,15,15,18,20,-15
     1,-17,-22,-23,-22,-21,-23,-23,-28,-23,-20,-17,-13,-10,-8,-7,-3,1
     1,4,6/
      DATA NN/765,731,681,650,630,613,609,610,620,677,734,755,773,803
     1790,760,766,736,771,860,839,789,781,756/
C
C     U(T-56)= NU(1), U(T-55)= NU(2)     ....ETC
C
        DATA NU/1,2,2,0,-1,-2,-5,-4,-17,-19,-19,-19,-18,-16,-14,-14
     1,-16,-19,-19,-19,-19,-16,-17,-18/
C
C     MODEL PARAMETERS
C
      READ(6,600)A1
600   FORMAT(F5.2)
      READ(6,602)A2
602   FORMAT(F6.3)
      READ(6,602)A3
      READ(6,610)B0
```

```
610     FORMAT(F4.2)
        READ(6,610)B1
        READ(6,600)B2
C
C       SAVE DATA
        DO 140 I=1,24
140     IER(I)=NU(I)
        DO 160 I=25,56
160     IER(I)=T(I-24)
C
        DO 90 I=1,24
90      WW(I)=NN(I)
        DO 110 I=25,50
110     WW(I)=X(I-24)
C
C       COMPUTATION OF NOISE TERMS
C       WE REQUIRE 1-STEP AHEAD FORECASTS WITH ORIGINS T=-24,-23,-22,
C       AND SAVE THE RESULTS IN ARRAY W
C
C       Z(T-50)= WW(1), Z(T-49)= WW(2)  ETC
        DO 115 I=1,23
        W(I)=WW(I+3)-A1*(WW(I+25)-WW(I+2))-A2*(WW(I+26)-WW(I+1))
       1-A3*(WW(I+27)-WW(I))+B0*(IER(I+26)-IER(I+2))
       1+B1*(IER(I+25)-IER(I+1))+B2*(IER(I+24)-IER(I))
        W(I)=X(I+3)-W(I)
        WRITE(5,888)W(I),I
888     FORMAT(10X,I6,I6)
115     CONTINUE
        DO 112 I=1,50
112     U(I)=T(I)
C       VALUE OF Z AT THE ORIGIN
        Z0=6640
C       USE ARRAY NPLOT TO STORE PAST AND FORCASTED VALUES
C       TO BE USED LATER FOR NOISE CALCULATIONS
C
        CALL FORCST(X,U,Z,Z0,W,A1,A2,A3,B0,B1,B2)
C
C
C       SAVE THE FORECASTS TO THIS POINT IN ARRAY ZI
        DO 120 I=1,23
120     ZI(I)=Z(I)
C
C
C
C
C
C
C
C       PERCENT ERROR IN FORECASTING
        DO 180 I=1,24
        IERR=ZI(I)-A(I)
        IF(IERR.GT.100)IERR=20
        WRITE(5,778)ZI(I),A(I)
778     FORMAT(10X,I6,I6)
180     IER(I)=(100*IERR)/A(I)
C
```

```
C       PRINT OUT DATA
C       WRITE HOURS IN ARRAY IHR
        DO 520 I=1,48
520     IHR(I)=I
        WRITE(5,409)
409     FORMAT('1')
        WRITE(5,410)
410     FORMAT(35X,'LOAD PREDICTION USING BOX & JENKINS METHOD'//)
        WRITE(5,420)
420     FORMAT(35X,'HYDRO-QUEBEC LOAD DATA FOR JANUARY 20,1972 (
       1IN 10MW UNITS)'//)
        KI=1
        KJ=12
82      WRITE(5,430)(IHR(M),M=KI,KJ)
430     FORMAT(35X,'HOUR:',12I4)
        WRITE(5,440)(ZI(M),M=KI,KJ)
440     FORMAT(30X,'PREDICTED:'12I4)
        WRITE(5,450)(A(M),M=KI,KJ)
450     FORMAT(33X,'ACTUAL:',12I4)
        WRITE(5,460)(IER(M),M=KI,KJ)
460     FORMAT(32X,'% ERROR:',12I4/)
        KI=KI+12
        KJ=KJ+12
        IF(KJ.GE.36)GOTO 81
        GOTO 82
81      WRITE(5,525)
525     FORMAT(X,///)
        CALL BPLOT(ZI,A)
C
C
C       LEAD TIME OF FORECAST
        LEAD=23
C
C       SET MEAN ERROR OF FORECAST TO ZERO
        XMEAN=0
C
        DO 500 I=1,LEAD
        MIN=IABS(ZI(I)-A(I))
        XMEAN=MIN+XMEAN
500     CONTINUE
        XMEAN=XMEAN/LEAD
C       ACTUAL STANDARD DEVIATION
C       INITIALIZE
        FSTD=0
        SSTD=0
        DO 502 I=1,LEAD
        FSTD=(ZI(I)-A(I))**2+FSTD
        NFST=IABS(ZI(I)-A(I))
502     SSTD=SSTD+NFST
        SSTD=(SSTD**2)/LEAD
        SSTD=ABS(FSTD-SSTD)
        SSTD=SSTD/(LEAD-1)
        SSTD=SQRT(SSTD)
C       RMS ERROR CALCULATION
        RMS=SSTD**2+(XMEAN**2)*LEAD/(LEAD-1)
        RMS=SQRT(RMS)
```

```
C       PEAK LOAD FOR THE DAY
        PEAKLD=879.0
        PERCNT=(RMS*100)/PEAKLD
        WRITE(5,511)
511     FORMAT('1')
        WRITE(5,504)
504     FORMAT(30X,'ERROR SUMMARY FOR JANUARY 20,1972 LOAD PREDICTION'//)
        WRITE(5,506)XMEAN
506     FORMAT(20X,'MEAN ERROR OF PREDICTION','=',F6.2/)
        WRITE(5,508)SSTD
508     FORMAT(19X,'ACTUAL STANDARD DEVIATION=',F6.2/)
        WRITE(5,510)RMS
510     FORMAT(24X,'RMS PREDICTION ERROR=',F6.2/)
        WRITE(5,512)PERCNT
512     FORMAT(14X,'RMS ERROR PERCENT OF PEAK LOAD=',F5.2/)
C
        STOP
        END

ROUTINES CALLED:
FORCST, BPLOT , IABS  , ABS    , SQRT

BLOCK       LENGTH
MAIN.   3250   (014544)*

**COMPILER -----  CORE**
   PHASE        USED  FREE
DECLARATIVES 00607 08529
EXECUTABLES  00927 08209
ASSEMBLY     01937 10116
```

```
                                    00:28:08    06-FEB-78    PAGE    1
      SUBROUTINE FURCST(X,U,Z,Z0,W,A1,A2,A3,B0,B1,B2)
C
C     SUBROUTINE USED BY BOX & JENKINS METHOD
C     TO COMPUTE FORECASTS FOR THE NEXT 23 HOURS WITH HOUR 0 AS ORIGIN
C
      DIMENSION X(26),U(100),Z(30)
      DIMENSION W(75)
      INTEGER X,U,Z,Z0,W
C     COMPUTE FORECASTS
C
      Z(1)=X(4)-A1*((Z0/10)-X(3))-A2*(X(26)-X(2))-A3*(X(25)-X(1))
     1+B0*(U(27)-U(3))+B1*(U(26)-U(2))+B2*(U(25)-U(1))
     1-W(1)
      Z(2)=X(5)-A1*(Z(1)-X(4))-A2*((Z0/10)-X(3))-A3*(X(26)-X(2))
     1+B0*(U(28)-U(4))+B1*(U(27)-U(3))+B2*(U(26)-U(2))
     1-W(2)
      Z(3)=X(6)-A1*(Z(2)-X(5))-A2*(Z(1)-X(4))-A3*((Z0/10)-X(3))
     1+B0*(U(29)-U(5))+B1*(U(28)-U(4))+B2*(U(27)-U(3))
     1-W(3)
      DO 140 I=4,23
      Z(I)=X(I+3)-A1*(Z(I-1)-X(I+2))-A2*(Z(I-2)-X(I+1))
     1-A3*(Z(I-3)-X(I))+B0*(U(I+26)-U(I+2))+B1*(U(I+25)-U(I+1))+
     1B2*(U(I+24)-U(I))
     1-W(I)
140   CONTINUE
      RETURN
      END

BLOCK       LENGTH
FURCST  553    (002122)*

**COMPILER ----- CORE**
   PHASE         USED  FREE
DECLARATIVES  00366 08770
EXECUTABLES   00607 08529
ASSEMBLY      01065 10988
```

```
      SUBROUTINE BPLOT(ZI,A)
      DIMENSION A(75),ZI(75),NPLOT(100)
      INTEGER ZI,A
C     PLOTTING ROUTINE
      WRITE(5,220)
220   FORMAT('1')
      DO 200 I=1,100
200   NPLOT(I)='*'
      WRITE(5,300)NPLOT
300   FORMAT(25X,100A1)
      DO 210 I=1,100
210   NPLOT(I)=' '
      N=1
90    J=ZI(N)
      J=IABS(J)
      J=J/10
      NK=A(N)/10
      IF(J.GT.100)J=100
      IF(NK.GT.100)NK=99
      IF(J.LE.0)J=1
      NPLOT(NK)='A'
      NPLOT(J)='+'
      NPLOT(1)='*'
      WRITE(5,310)NPLOT
310   FORMAT(20X,100A1)
      NPLOT(J)=' '
      NPLOT(NK)=' '
      N=N+1
      IF(N.NE.26)GOTO 90
      RETURN
      END

ROUTINES CALLED:
IABS

BLOCK      LENGTH
BPLOT   543    (002076)*

**COMPILER ----- CORE**
   PHASE       USED  FREE
DECLARATIVES  00446 08690
EXECUTABLES   00607 08529
ASSEMBLY      01169 10884
```

APPENDIX B

DATE	HOUR	LOAD DATA in 10 MW UNITS	DORVAL TEMPERATURE (°F)	TEMPERATURE DEPENDENT FUNCTION (°F) U_t
Jan 18	0	681	28	-17
	1	650	29	-19
	2	630	29	-19
	3	613	28	-19
	4	609	27	-18
	5	610	25	-16
	6	620	23	-14
	7	677	23	-14
	8	734	25	-16
	9	755	29	-19
	10	773	30	-19
	11	803	32	-19
	12	790	33	-19
	13	760	31	-16
	14	766	32	-17
	15	736	34	-18
	16	771	35	-20
	17	860	39	-25
	18	839	39	-25
	19	789	36	-22
	20	781	36	-23
	21	756	37	-24
	22	729	37	-25
	23	693	38	-26
Jan 19	0	647	38	-26
	1	616	37	-25
	2	603	38	-27
	3	592	43	-33
	4	579	44	-34
	5	589	43	-33
	6	607	42	-32
	7	651	41	-31
	8	701	41	-31

DATE	HOUR	LOAD DATA in 10 MW UNITS	DORVAL TEMPERATURE (°F)	TEMPERATURE DEPENDENT FUNCTION (°F) U_t
Jan 19	9	738	42	-31
	10	771	37	-24
	11	786	38	-24
	12	776	37	-22
	13	749	36	-20
	14	746	32	-15
	15	740	30	-13
	16	754	28	-11
	17	843	27	-11
	18	839	26	-11
	19	819	25	-10
	20	795	24	-10
	21	770	22	-8
	22	752	22	-8
	23	715	20	-6
Jan 20	0	664	18	-5
	1	637	19	-5
	2	626	19	-6
	3	616	20	-7
	4	607	21	-8
	5	609	16	-3
	6	624	12	1
	7	684	11	2
	8	733	11	2
	9	762	12	2
	10	778	11	4
	11	803	12	5
	12	804	13	5
	13	776	14	4
	14	782	14	5
	15	773	14	5
	16	796	13	6

DATE	HOUR	LOAD DATA in 10 MW UNITS	DORVAL TEMPERATURE (^oF)	TEMPERATURE DEPENDENT FUNCTION (^oF) U_t
Jan 20	17	879	14	4
	18	876	12	5
	19	852	10	7
	20	834	9	7
	21	820	9	7
	22	788	9	6
	23	747	9	6
Jan 21	0	692	10	5
	1	652	11	4
	2	637	11	3
	3	626	12	2
	4	616	12	2
	5	621	7	7
	6	633	7	7
	7	681	9	5
	8	747	7	7
	9	780	5	10
	10	794	4	12
	11	810	4	14
	12	799	4	15
	13	761	5	14
	14	755	6	14
	15	758	8	12
	16	768	7	13
	17	859	6	13
	18	867	3	16
	19	856	4	14
	20	832	3	15
	21	811	3	15
	22	779	-1	18
	23	752	-3	20

APPENDIX C

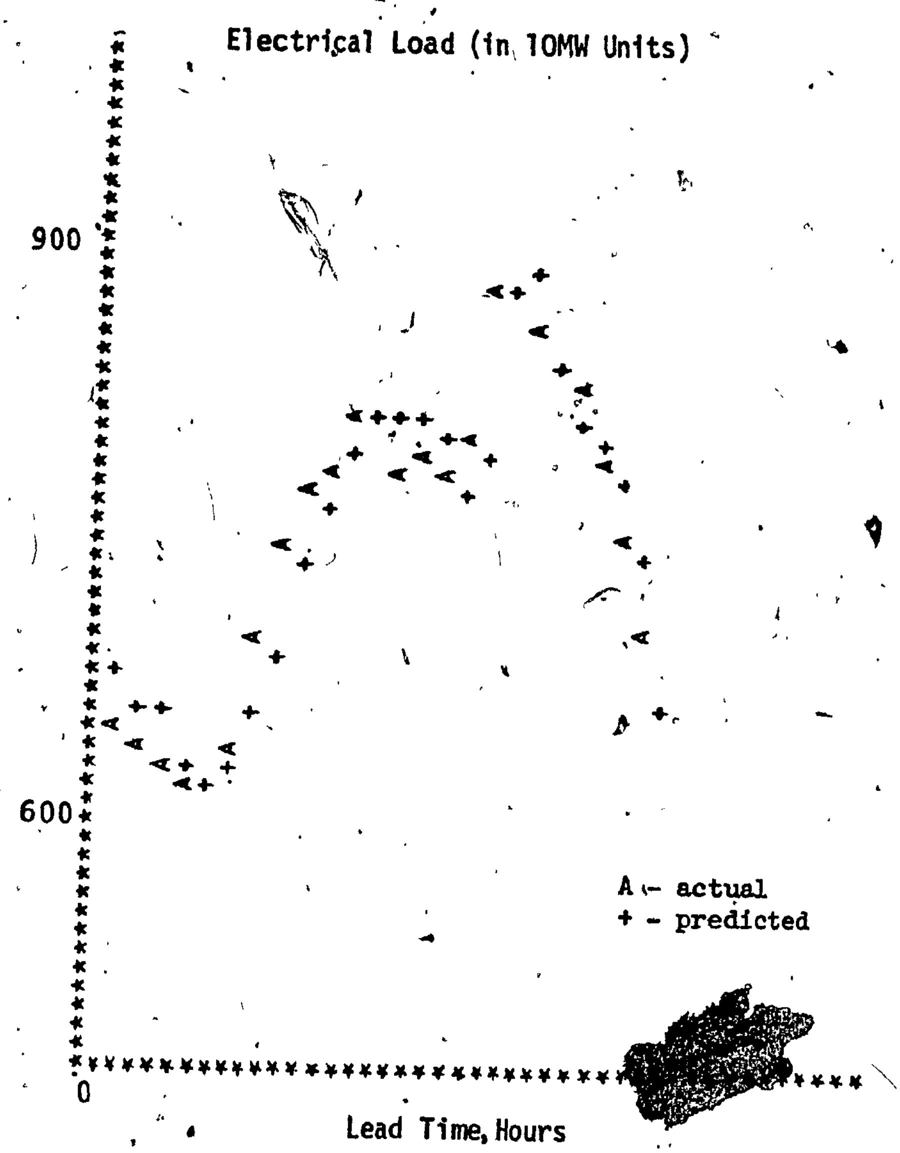


Fig. C-1 — Line Printer Plot of Prediction Obtained from Periodic and Residual Model.

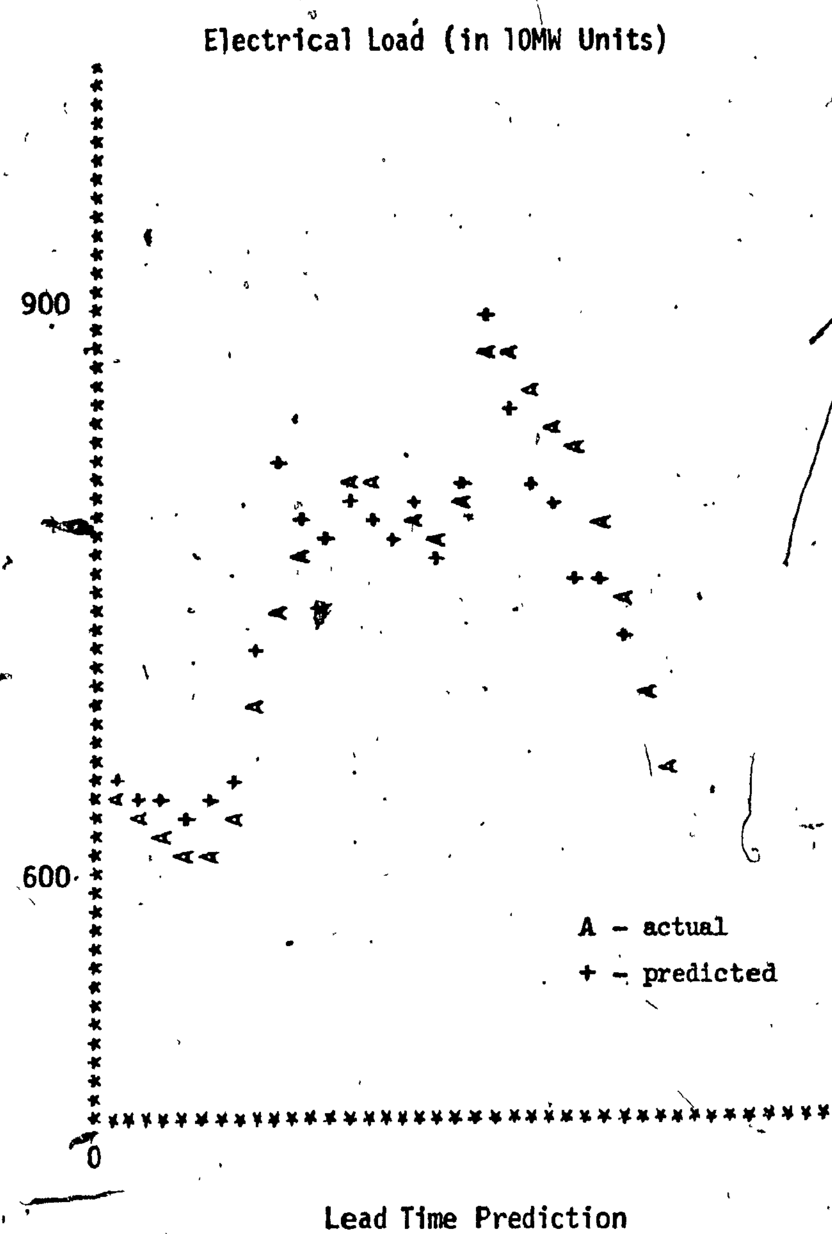


Fig. C-2 — Line Printer Plot of Prediction Obtained From Residual Model.

RESIDUAL COMPONENT PARAMETERS				
n	0	1	2	3
a_n	-	-0.65	-0.082	-0.018
b_n	2.70	4.25	-3.70	-
c_n, d_n	-	<0.01	<0.01	<0.01

PERIODIC COMPONENT PARAMETERS				
n	1	2	3	4
α_n	-965.6	-398.8	134.0	4.18
β_n	-273.1	199.4	-178.7	90.8
α_n	5	6	7	8
β_n	-101.2	-137.7	19.7	21.7
n	65.2	-32.6	-40.0	74.1

Table C-1 — Model Parameters.

APPENDIX D