APPLICATION OF CYBERNETICAL METHODS
IN
POWER INDUSTRY

Thesis

Submitted for the Degree of

Doctor of Philosophy (Engineering)

UNIVERSITY OF NORTH BENGAL

Jyotirmay Jhampati

Senior Assistant Engineer (Electrical)

WEST BENGAL STATE ELECTRICITY BOARD

JALPAIGURI : WEST BENGAL

P R E F A C E

We have inherited a very rich body of knowledge in cybernetics going back hundreds of years. Theories are numerous and tools are abundant. Ingenious adaptations of the tools are proliferating every day to real world problems. The present work is an experiment with the applications of the cybernetical tools in electrical power industry. Our endeavour deals with the iterations of observation - conjecture - experiment - theory - modelling - validation cycle[1] .

ACKNOWLEDGEMENT

The investigator expresses his deep sense of love and gratitude to Dr. A.S. Chaudhuri for suggesting the problem and for his invaluable help and encouragement in the day to day work as the supervisor of the research project.

The investigator is ever grateful to his respected teacher Professor N.Majumdar of Bengal Engineering College for introducing him in the fascinating discipline of Computer techniques for power networks.

The investigator also expresses his deep sense of respect to his teacher Dr. B. Sen, Director of Technical Education, West Bengal for his constant en-couragement throughout the execution of this research project.

The investigator expresses his indebtedness to all his teachers and Library staff of Jalpaiguri Government Engg. College who extended their whole hearted assistance whenever needed.

The investigator is also indebted to his organisation, West Bengal State Electricity Board and the Government of West Bengal for permitting him to work at Jalpaiguri Govt. Engineering College.

Thanks are expressed to Mr. Dulal Nandi who has taken all the troubles to type the manuscript of the present dissertation with much care and devotion.

Last but not least, the investigator is grateful to his mother. But for her blessings and advice at the moments of success and failure present dissertation might not be possible.

J. Jhampati,
Assistant Engineer,
Electrical,
West Bengal State Electricity Board.

// A B S T R A C T

Power industry is a part of the global system to support life. It comprises a number of subsystems whose interactions are effected through the use of direct links and large amount of feedbacks. Application of cybernetical methods to analyse information related processes in power industry would yield useful results. The present investigation deals with the experiments with the applications of cybernetical tools to the five different aspects of the electrical power industry.

For on-line operation of hydroelectric power plant on real time basis it is essential to have an accurate one step ahead estimation of river flow. The present investigation develops the hourly flow simulation technique with the cybernetical method of recursive least square instrument variable algorithm with parameter tracking adaptiveness. Effectiveness of the developed technique has been demonstrated with field data observed at different gauging stations of the hilly river Teesta in North Bengal. On line flow simulation has been done at Coronation Bridge point which has the potentiality of a large hydroelectric plant with an estimated generation capacity of about 1000 MW.

Growth models of Electrical Energy consumption has been developed with gross national product, gross domestic capital formation and other associated variables as exogenous ones. The model has been developed in the form of a polynomial of optimum complexity with the help of the multilayer group method of data handling algorithm.

A desired rate of growth of energy consumption has been assumed. On the basis of this growth rate the trend of energy consumption upto 2000 AD has been extrapolated. The model of energy consumption has been obtained in the form of a polynomial of optimal complexity by computer aided self organisation of mathematical models. A model for energy utilisation factor is also obtained. The models can be used as handy tools for planners of power industry.

State estimation technique provides a powerful tool to obtain a data base for on-line supervision and control of power system. In this work recursive type least square technique is used to obtain the state estimation of the power system parameters. The estimates of the states will help in selecting on-line contingency plan.
An illustration is given to show the application of the methods developed.

Dynamic programming technique of applied cybernetics has been used for optimal ordering of nodes for load flow analysis. The illustration shows that the developed method is capable of improving the computational efficiency of the load flow analysis.

Investigation, carried out in this work, has helped in developing necessary softwares for off-line planning and on-line control of electrical power industry. It also shows that cybernetical methods are powerful tools for analysing the different aspects of electrical power systems.

C O N T E N T S

CHAPTER III

CHAPTER I

I N T R O D U C T I O N

CHAPTER I

1.0 I N T R O D U C T I O N

The present investigation is an account of our experiments with the applications of different cybernetical tools to the real world problems of electrical power industry. The investigation deals with the five broad aspects of electrical power industry namely,

(i) On-line simulation of hourly river flows for run-of-the river hydroelectric plant ;

(ii) Electrical energy consumption model with interacting parameters by a learning identification algorithm ;

(iii) Medium-term and long-term prediction models of annual installed capacity and consumption of electrical energy by computer-aided self-organisation of mathematical models ;

(iv) States estimation of electrical power systems by a tracking algorithm ; and

(v) Gauss-seidel load flow with optimally ordered nodes by dynamic programming algorithm.

The practical implementability of different methods of applied cybernetics and pattern recognition such as recursive least square instrument variable algorithm with on-line adaptiveness in parameter variations, multilayer and combinatorial group method of data handling algorithms of computer-aided self-organised learning identification technique, least square recursive states tracking algorithm and dynamic programming algorithm of optimum nodes ordering technique for power networks have been demonstrated. Whereever necessary modification and improvement of the existing analytical techniques have been suggested for practical implementations. The computational procedures have been developed in well organised programme packages in high level software.

The report contains eight chapters. The contents and the scope of the individual chapter are briefly described below :

1.1 Scope of the Work

1.1.1 Chapter I

The chapter I deals with the introduction and the problem of investigation. It contains the scope of the work and the sources of data.

1.1.2 Chapter II

The chapter II deals extensively with the survey of the existing literature and the state of-the-art. It discusses the effectiveness and the shortcomings of the previous works. It exposes gradually the harmonious evolution of the scientific thoughts pertaining to the present investigation.

1.1.3 Chapter III

The chapter III deals with the on-line simulation of the hourly river flows for run-of-the river hydro-electric plant. It develops the hourly flow simulation technique with the cybernetical method of recursive least square instrument variable algorithm with on-line parameter tracking adaptiveness. The effectiveness of the developed technique has been demonstrated with field data observed at the different gauging stations of the hilly river Teesta in North Bengal.

1.1.4 Chapter IV

The chapter IV develops a mathematical description of annual energy consumption in India with population, gross national product, gross domestic saving and gross domestic capital formation as exogenous variables in the form of a polynomial of optimum complexity with the help of a learning identification technique known as multilayer group method of data handling algorithm. The developed model is found to simulate adequately the effects of interactions of different technoeconomic parameters on annual electrical energy consumption.

1.1.5 Chapter V

In chapter V a model of annual installed plant capacity of electrical energy of India has been obtained in the form of polynomial of optimum complexity by computer -aided self-organisation of mathematical models. Desired rates of growth of annual installed plant capacity and annual energy consumption have been assumed. On the basis of the growth rates the polynomial models of optimum complexity have been obtained for annual installed plant capacity and energy consumption. A model for plant annual load factor has also been obtained in the form of a polynomial with harmonic terms. The developed models can be used as handy tools for planners of power industry.

1.1.6 <u>Chapter VI</u>

It is observed that the state estimation technique provides a powerful tool to obtain a data base for on-line supervision and control of power system. In this chapter recursive type least square technique with parameter tracking algorithm has been used to obtain the estimation of the power system state variables. Incorporation of the parameter tracking algorithm makes the state estimator amenable to on-line operation. The estimates of the states will help in selecting on-line contingency plan. An illustration is given to show the application of the method.

1.1.7 Chapter VII

This chapter deals with the Gauss-Seidel load flow technique of electrical power networks with optimally ordered nodes by dynamic programming algorithm of applied cybernetics. The algorithm has been illustrated with IEEE 14 Bus System. It has been observed that the computational efficiency is improved with optimal ordering by dynamic programming algorithm.

1.1.8 Chapter VIII

This chapter concludes the report. The specific areas of further research are suggested. At the end a list of referenced bibliography is enclosed.

1.2 Sources of Data

Data pertaining to the Teesta river system have been obtained from Jalpaiguri Field Division of the Central Water Commission, Government of India.

Data relating to the electrical energy consumption models with interacting parameters have been obtained from the Economic Survey, 1981 - 82, Government of India.

Data pertaining to medium term and long term prediction models of installed plant capacity and electrical energy consumption have been obtained from the Sixth Five Year Plan, 1980 - 81, Government of India.

Data relating to the state estimation of electrical power system have been obtained from the 5 - Bus network given in Computer Methods in Power System Analysis - G.W. Stages and A. H El- Abiad, McGraw Hill, 1968.

Data for illustration of dynamic programming algorithm for optimum ordering of nodes have been obtained from IEEE 14 Bus system.

CHAPTER II

THE SURVEY OF THE EXISTING LITERATURE AND THE STATE-OF-THE-ART

CHAPTER II

THE SURVEY OF THE EXISTING LITERATURE AND THE STATE-OF-THE-ART

2.0 Introduction

The outcome of the present investigation is the results of our experiments with the applications of different methods of cybernetics to the electrical power industry. The methods which are relevant to the present investigation are discussed with the associated state-of-the-art.

2.1.0 On-line Simulation of Hourly River Flows

The chapter III deals with the on-line simulation of hourly river flows for run-of-the-river hydroelectric plant. For on-line operation of hydroelectric power plant on real-time basis it is essential th have an accurate one step ahead estimation of river flow. The present investigation develops the hourly flow simulation technique with the cybernetical method of recursive least square instrument variable algorithm with parameter tracking adaptiveness.

There are many ways of obtaining recursive algorithms. Some of the early references on recursive identification methods are given in [2]. It is not attempted to present all its variants in their wide spectrum of use. The discussion is limited to that part which is relevant to the present investigation and it deals with more than just the subject of estimation algorithm : it treats also the subjects of system identification and forecasting. This is due to the fact that the techniques of estimation derive in part from the broader field of system identification which incorporates estimation with model structure identification, model verification and model validation. The investigator is heavily debted to Beck [3] for his excellent treatment of the subject in a highly understandable tutorial fashion for practical use. Excellent treatment of the time series by Young [4] has acted as a guide.

In the present investigation black box models have been assumed and therefore only such models are discussed. This type of models is often encountered in physical system. When suitably transformed the model becomes amenable to recursive techniques. Soderstrom et al [5], and Ljung [6,7,8,9 & 10] has given a good coverage on recursive identification methods.

Recursive technique has been defined by Young [11] as "a technique in which an estimate is updated on receipt of fresh informations." Steps of development of the Recursive parameter estimation algorithm has been depicted in Fig. 2.1.0.

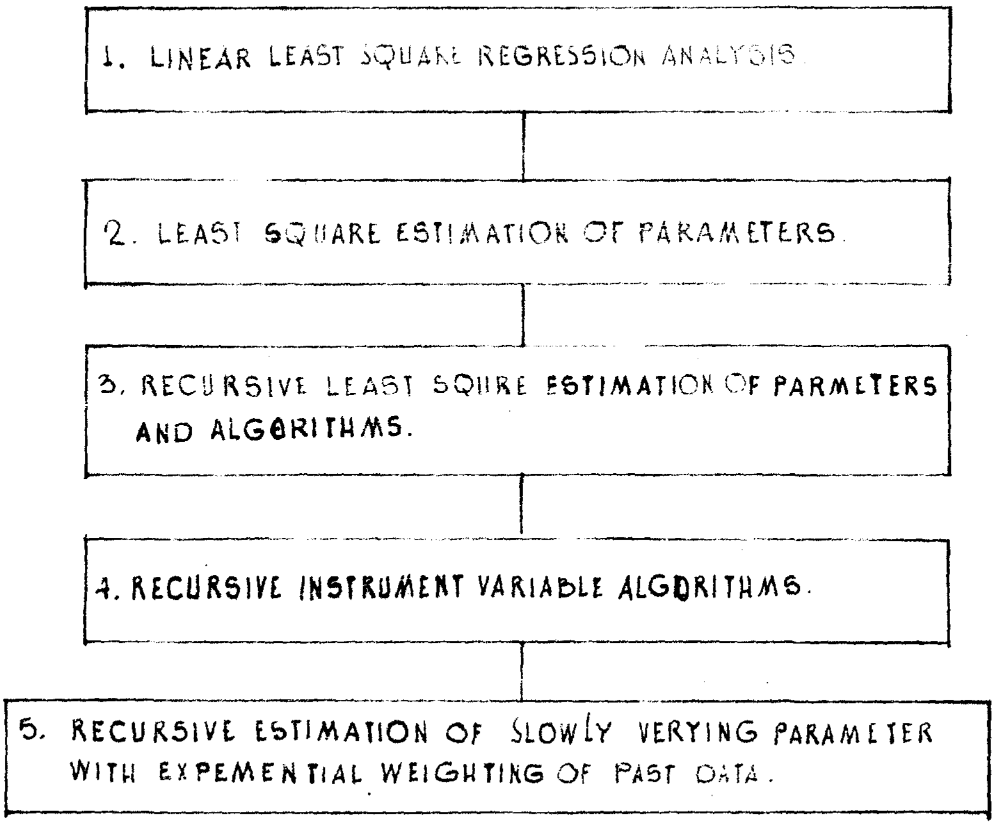


FIG. 2.1.0

AN OUTLINE OF THE DERIVATION OF RECURSIVE ESTIMATION ALGORITHMS.

2.1.1 Linear Least Square Regression Analysis

Without making any assumption about the physical phenomena of the system which are thought to govern the system's dynamic behaviour the system is often viewed as a black box. Identification means that a model is fitted to measured U_i [i = 1,2,....n] of the input signal and y_i [i = 1,2,... n] of the output signal. The analysis of time series suggests that there is a probability that the current values of the output $Y(t_k)$ is a function of the previous output observations, the autoregressive terms $Y(t_{k-1})$, $Y(t_{k-2})$, ... ; and the past observations of inputs $U_1(t_{k-1})$, $U_2(t_{k-1})$, $U_1(t_{k-2})$, $U_2(t_{k-2})$ together with the current unknown realistic noise process $V(t_k)$. Therefore the system may be assumed as

$$Y(t_k) = \sum_{i=1}^{n} \beta_i Y(t_{k-i}) + \sum_{j=1}^{m} \sum_{i=1}^{n} \delta_{ji} U_j(t_{k-i}) + V(t_k) \qquad \ldots (2.1.1)$$

Determination of n is known as model order determination. Chaudhuri [12] has suggested output- output and output- input correlation as an intuitive consideration for model order determination which is also the model structure identification.

In the polynomials of backward shift operator, the equation (2.1.1) can be rearranged as

$$A(q^{-1})\,Y(t_k) = \sum_{j=1}^{m} B_j(q^{-1})U_j(t_k) + V(t_k) \qquad (2.1.2)$$

where the backward shift operator q^{-1} is defined by

$$q^{-1}\,Y(t_k) = Y(t_{k-1}) \qquad (2.1.3)$$

and

$$A(q^{-1}) = 1 - \beta_1 q^{-1} - \beta_2 q^{-2} \;\cdots\; - \beta_n q^{-n} \qquad (2.1.4)$$

$$B_j(q^{-1}) = \delta_{j1} q^{-1} + \delta_{j2} q^{-2} \;\cdots\; + \delta_{jn} q^{-n} \qquad (2.1.5)$$

This model is quite flexible since it requires that the equations be linear in parameters [13].

Equation (2.1.2) can be represented as

$$Y(t_k) = Z^T(t_k).\alpha + V(t_k) \qquad (2.1.6)$$

where

$$Z^T(t_k) = [\,Y(t_{k-1}) \ldots Y(t_{k-n}),\ U_1(t_{k-1}) \ldots U_1(t_{k-n}) \ldots U_m(t_{k-1}) \ldots U_m(t_{k-n})\,] \qquad (2.1.7)$$

and

$$\alpha = [\,\beta_1 \ldots \beta_n,\ \delta_{11} \ldots \delta_{1n} \ldots \delta_{m1} \ldots \delta_{mn}\,] \qquad (2.1.8)$$

2.1.2 Least Square Estimation of Parameters

Least square estimate of the parameters is obtained by minimising the loss function defined as the sum of the squared errors,

$$J \triangleq \sum_{k=1}^{N} \left(Y(t_k) - Z^T(t_k)\hat{\alpha} \right)^2 \qquad (2.1.9)$$

in which the estimates $\hat{\alpha}$ of α that minimises J are called the least square estimates. The model response errors $V(t_k) = Y(t_k) - Z^T(t_k)\hat{\alpha}$ are not in general identical with $V(t_k)$ but converges to $V(t_k)$ as $\hat{\alpha}$ converges to true value of α.

Differentiating J with respect to parameter vector and then setting the vector of derivatives equal to zero we have the well known equations for the least square parameter estimates,

$$\hat{\alpha} = \left[\sum_{k=1}^{N} Z(t_k)\, Z^T(t_k) \right]^{-1} \left[\sum_{k=1}^{N} Z(t_k)\, Y(t_k) \right] \qquad (2.1.10)$$

2.1.3 Recursive Least Square Estimation of Parameters

Recursive form of least square estimation of parameters is an elegant way of updating estimates $\hat{\alpha}$ which changes as it converges to true value α.

Consider the equation

$$Y(t_k) = Z^T(t_k)\alpha + V(t_k)$$

The least square estimate $\hat{\alpha}$ of α is given by the equation (2.1.10). With direct analogy from equation (2.1.10) we may write.

$$\hat{\alpha}(t_k) = \left[\left(\sum_{j=1}^{k} Z(t_j)Z^T(t_j) \right)^{-1} \sum_{j=1}^{k} Z(t_j)Y(t_j) \right] \qquad (2.1.11)$$

Equation (2.1.11) can be written in coincise form as

$$\hat{\alpha}(t_k) = P(t_k)\, b(t_k) \qquad (2.1.12)$$

where

$$P(t_k) \triangleq \left[\sum_{j=1}^{k} Z(t_j)Z^T(t_j) \right]^{-1}$$

and

$$b(t_k) \triangleq \left[\sum_{j=1}^{k} Z(t_j)Y(t_j) \right]$$

Recursive relationship for $P(\cdot)$ and $b(\cdot)$ can be set as

$$\left[P(t_k) \right]^{-1} = \sum_{j=1}^{k} Z(t_j)Z^T(t_j)$$

$$= \left[P(t_{k-1}) \right]^{-1} + Z(t_k)Z^T(t_k) \qquad (2.1.13)$$

Similarly

$$b(t_k) = b(t_{k-1}) + Z(t_k)\, Y(t_k) \qquad (2.1.14)$$

Premultiplying by $P(t_k)$ and post-multiplying by $P(t_{k-1})$ we get from equation (2.1.13)

$$P(t_{k-1}) = P(t_k)+P(t_k)Z(t_k)Z^T(t_k)P(t_{k-1}) \qquad (2.1.15)$$

Post multiplying by $Z(t_k)$ equation (2.1.15) gives,

$$P(t_{k-1})Z(t_k) = P(t_k)Z(t_k)\left[1+Z^T(t_k)P(t_{k-1})Z(t_k)\right] \qquad (2.1.16)$$

Post multiplying by $\left[1+Z^T(t_k)P(t_{k-1})Z(t_k)\right]^{-1} Z^T(t_k)P(t_{k-1})$ equation (2.1.16) gives,

$$P(t_{k-1})Z(t_k)\left[1+Z^T(t_k)P(t_{k-1})Z(t_k)\right]^{-1} Z^T(t_k)P(t_{k-1})$$
$$= P(t_k)Z(t_k)Z^T(t_k)P(t_{k-1}) \qquad (2.1.17)$$

From equation (2.1.15) we get

$$P(t_k) = P(t_{k-1}) - P(t_k)Z(t_k)Z^T(t_k)P(t_{k-1}) \qquad (2.1.18)$$

Finally substitution of equation (2.1.17) in equation (2.1.18) gives,

$$P(t_k)= P(t_{k-1})- P(t_{k-1})Z(t_k)\left[1+Z^T(t_k)P(t_{k-1})Z(t_k)\right]^{-1} Z^T(t_k)P(t_{k-1}) \qquad \ldots (2.1.19)$$

Now from equation (2.1.12)

$$\hat{\alpha}(t_k) = P(t_k).\ b(t_k)$$

i.e.

$$\hat{\alpha}(t_k) = [P(t_{k-1}) - P(t_{k-1})Z(t_k)[1+Z^T(t_k)P(t_{k-1})Z(t_k)]^{-1} Z^T(t_k)P(t_{k-1})](\ b(t_{k-1})+Z(t_k)Y(t_k))$$

and since $P(t_{k-1})\ b(t_{k-1}) = \hat{\alpha}(t_{k-1})$

$$\hat{\alpha}(t_k) = \hat{\alpha}(t_{k-1}) - P(t_{k-1})Z(t_k)[1+Z^T(t_k)P(t_{k-1})Z(t_k)]^{-1} Z^T(t_k)\hat{\alpha}(t_{k-1})$$

$$+ P(t_{k-1})Z(t_k)Y(t_k) - P(t_{k-1})Z(t_k)[1+Z^T(t_k)P(t_{k-1})Z(t_k)]^{-1}$$

$$x\ Z^T(t_k)P(t_{k-1})Z(t_k)Y(t_k)$$

Since $Z^T(t_k)\ P(t_{k-1})\ Z(t_k)$ is scaler and hence

$[1 + Z^T(t_k)\ P(t_{k-1})\ Z(t_k)]^{-1}$ is also scalar. Therefore

$$\hat{\alpha}(t_k) = \hat{\alpha}(t_{k-1}) - P(t_{k-1})Z(t_k)[1+Z^T(t_k)P(t_{k-1})Z(t_k)]^{-1} Z^T(t_k)\ \hat{\alpha}(t_{k-1})$$

$$+\ P(t_{k-1})Z(t_k)\ [1+Z^T(t_k)P(t_{k-1})Z(t_k)]^{-1} Y(t_k)$$

And hence we get the recursive least square parameter estimation algorithm as

$$\hat{\alpha}(t_k) = \hat{\alpha}(t_{k-1}) + P(t_{k-1}) Z(t_k) \left[1 + Z^T(t_k) P(t_{k-1}) Z(t_k) \right]^{-1}$$

$$\left[Y(t_k) - Z^T(t_k) \hat{\alpha}(t_{k-1}) \right] \qquad (2.1.20)$$

where

$$P(t_k) = P(t_{k-1}) - P(t_{k-1}) Z(t_k) \left[1 + Z^T(t_k) P(t_{k-1}) Z(t_k) \right]^{-1} Z^T(t_k) P(t_{k-1})$$

... (2.1.21)

with

$$P(t_{k-1}) = \left[\sum_{j=1}^{k-1} Z(t_j) Z^T(t_j) \right]^{-1} \qquad (2.1.22)$$

Least square technique is of great importance for parameter estimation [14]. But this does not overcome the problem of bias.

The problem of bias is discussed below.

The non-recursive least square estimation of parameter is given by

$$\hat{\alpha}(t_k) = \left[\sum_{j=1}^{k} Z(t_j) Z^T(t_j) \right]^{-1} \left[\sum_{j=1}^{k} Z(t_j) Y(t_j) \right] \qquad (2.1.23)$$

where the system equation is taken as

$$Y(t_j) = Z^T(t_j) \alpha + V(t_j) \qquad (2.1.24)$$

Upon substitution of equation (2.1.24) in equation (2.1.23) we have

$$\hat{\alpha}(t_k) = \left[\sum_{j=1}^{k} Z(t_j) Z^T(t_j) \right]^{-1} \left[\sum_{j=1}^{k} Z(t_j) \left[Z^T(t_j)\, \alpha + V(t_j) \right] \right]$$

$$= \left[\sum_{j=1}^{k} Z(t_j) Z^T(t_j) \right]^{-1} \left[\sum_{j=1}^{k} Z(t_j)\, Z^T(t_j)\, \alpha \right]$$

$$+ \left[\sum_{j=1}^{k} Z(t_j) Z^T(t_j) \right]^{-1} \left[\sum_{j=1}^{k} Z(t_j)\, V(t_j) \right]$$

Therefore,

$$\hat{\alpha}(t_k) = \alpha + \left[\sum_{j=1}^{k} Z(t_j) Z^T(t_j) \right]^{-1} \left[\sum_{j=1}^{k} Z(t_j) V(t_j) \right] \quad (2.1.25)$$

Equation (2.1.25) shows that for estimate $\hat{\alpha}(t_k)$ to be unbiased the following condition must hold

$$E\left\{ Z(t_j) V(t_j) \right\} = 0 \quad \text{for all} \quad K \qquad (2.1.26)$$

This is only possible if $V(t_k) = e(t_k)$ where $e(t_k)$ is a white noise sequence [3].

As pointed out by Beck [15] for most practical cases of interest $V(t_k)$ is not a white gassian sequence and the estimate $\hat{\alpha}$ is not unbiased. To overcome the problem of bias many variants of recursive parameter estimation algorithms have been suggested of which recursive instrument variable algorithm of Young [11] and generalised least square formulation of Hastings and James et al [16] are important. The essential components of these algorithms are similar. The recursive instrument variable algorithm is easily amenable to computation and as observed by Young [17,18,19,20,21 & 22] it may well offer and unified and comprehensive approach to system identification.

2.1.4 Recursive Instrument Variable Algorithm

Most likely source of biased estimate is the presence of auto correlated noise process $E\{V(t_k)\ V(t_{k-1})\} \neq 0$ for all k which implies $E\{Y(t_{k-1})\ V(t_k)\} \neq 0$, i.e. there is a significant correlation between the noise sequence and the past values of output. Referring to equation (2.1.2) and with suitable estimate of the parameters in $A(q^{-1})$ and $B_j(q^{-1})$ as $\hat{A}(q^{-1})$ and $\hat{B}_j(q^{-1})$ respectively a deterministic time series denoted as auxiliary model can be computed as

$$X(t_k) = [1 - \hat{A}(q^{-1})]X(t_k) + \sum_{j=1}^{m} \hat{B}_j(q^{-1})U_j(t_k) \qquad (2.1.27)$$

Equation (2.1.2) and (2.1.27) suggest that (i) variation in $\hat{Y}(t_k)$ should be strongly correlated with variations in the noise corrupted output observations $Y(t_k)$ but (ii) these variations in $\hat{Y}(t_k)$ should be uncorrelated with $V(t_k)$ provided $V(t_k)$ is not correlated with the measured input sequence $U_j(t_k)$ i.e. $E\{U_j(t_k)\ V(t_l)\} = 0$ for all j,k,l.

Sequence of $\hat{Y}(t_k)$ is called the sequence of instrumental variables. Consequently the vector $Z(t_k)$ is modified as $\hat{Z}(t_k)$ defined by

$$\hat{Z}(t_k) = \left[\ \hat{Y}(t_{k-1}), \ldots \hat{Y}(t_{k-n}),\ U_1(t_{k-1}), \ldots U_1(t_{k-n}), \ldots U_m(t_{k-n})\ \right]^T \qquad (2.1.28)$$

Conditions of unbiased estimates are modified as

$$E\left\{\hat{Z}(t_k)\ V(t_k)\right\} = 0 \quad \text{for all } k$$

Replacing $Z(t_k)$ by $\hat{Z}(t_k)$ and not $Z^T(t_k)$ by $\hat{Z}^T(t_k)$ heuristically. Recursive Instrument variable Algorithm is given by

$$\hat{\alpha}(t_k) = \hat{\alpha}(t_{k-1}) + \hat{P}(t_{k-1})\hat{Z}(t_k)\left[1+Z^T(t_k)\hat{P}(t_{k-1})\hat{Z}(t_k)\right]^{-1} \ Y(t_k) - Z^T(t_k)\ \hat{\alpha}(t_{k-1}) \qquad (2.1.29)$$

$$\hat{P}(t_k) = \hat{P}(t_{k-1}) - \hat{P}(t_{k-1})\,\hat{Z}(t_k)$$

$$\left[1+Z^T(t_k)\hat{P}(t_{k-1})\hat{Z}(t_k)\right]^{-1} Z^T(t_k)\hat{P}(t_{k-1}) \qquad (2.1.30)$$

with

$$\hat{P}(t_k) = \left[\sum_{j=1}^{k} \hat{Z}(t_j)Z^T(t_j)\right]^{-1} \qquad (2.1.31)$$

This recursive algorithm is used to estimate the parameters of a regression relationship which varies with time by passing through time series data and attempting to track the parameter variations. The data may be processed iteratively, each time using a data set in order to further refine the estimates to obtain better statistical efficiency. For a given block of N data elements the recursive method terminate after N steps where as the iterative procedure continues until parameters no longer changes with further iteration. For on-line process this can also be used as the basis for continuous updating of the auxiliary model parameters [22].

Sometimes large errors are found between the predicted and the observed output. This is not so much a consequence of spurious errors in the measurement but are due primarily to changing values of the model parameters.

2.1.5 Determination of Instrument Variables

The instrument variables $Y(.)$ in $z(.)$ are obtained through a separate parameter tracking algorithm as detailed below.

$$Y(t_k) = \sum_{i=1}^{n} \beta_i Y(t_{k-i}) + \sum_{j=1}^{m} \sum_{i=0}^{n} \delta_{ji} U_j(t_{k-rj-i})$$

$$+ \sum_{q=1}^{Q} C_{t_{k-q}} \left(Y(t_{k-q}) - \hat{Y}(t_{k-q}) \right) + \sigma(t_k) \qquad (2.1.32)$$

where the third component is the moving average component (t_k) is the error sequence.

$\hat{Y}(t_k)$, the estimate of $Y(t_k)$ can be written as

$$\hat{Y}(t_k) = a^T(t_{k-1}) \, z(t_{k-1}) \qquad (2.1.33)$$

where

$$a^T(t_{k-1}) = [\, \beta_1, \beta_2, \ldots, \delta_{10}, \delta_{11}, \ldots\ldots, \delta_{mn}, C_{t_{k-1}}, C_{t_{k-2}}, \ldots \,]$$

$$z(t_{k-1}) = [\, Y(t_{k-1}), Y(t_{k-2}), \ldots, U_1(t_{k-r1}) \ldots\ldots, U_m(t_{k-rm-n}) \,] \qquad (2.1.34)$$

The coefficient vector 'a' can be estimated by minimizing the quadratic performance criterion $J_k(a)$ [24,25,26,27], defined as,

$$J_k(a) \triangleq \sum_{j=1}^{k} (Y(t_j) - a^T z(t_{j-1}))^2 +$$

$$+ (a - a (t_0))^T S^{-1} (t_0) (a - a(t_0)) \qquad (2.1.35)$$

where $a(t_0)$ is the available a priori estimate of the coefficient vector 'a' and $S(t_0)$ is the positive definite weighting matrix of the order $ml \times ml$ where $ml = n+m(n+1)+Q$.

For minimisation,

$$\frac{\delta J_k(a)}{\delta a} = -2 \sum_{j=1}^{k} z(t_{j-1})(Y(t_j) - a^T z(t_{j-1}))$$

$$+ 2 S^{-1}(t_0)(a - a(t_0)) \qquad (2.1.36)$$

It follows from equation (2.1.36)

$$\sum_{j=1}^{k} z(t_{j-1})Y(t_j) + S^{-1}(t_0)a(t_0) = \sum_{j=1}^{k} z(t_{j-1})z^T(t_{j-1}) + S^{-1}(t_0)a$$

$$\ldots \quad (2.1.37)$$

Let

$$S^{-1}(t_k) = \sum_{j=1}^{k} z(t_{j-1})z^T(t_{j-1}) + S^{-1}(t_0) \qquad (2.1.38)$$

and

$$d(t_k) = \sum_{j=1}^{k} z(t_{j-1})Y(t_j) + S^{-1}(t_0)a(t_0) \qquad (2.1.39)$$

Denoting the coefficient vector 'a' as a (t_k) at the time instant t_k,

$$S^{-1}(t_k)\, a(t_k) = d(t_k) \qquad (2.1.40)$$

or

$$a(t_k) = S(t_k)\, d(t_k)$$

From equations (2.1.38) and (2.1.39) the following recursive equations are obtained,

$$S^{-1}(t_{k+1}) = S^{-1}(t_k) + z(t_k)\, z^T(t_k) \qquad (2.1.41)$$

$$d(t_{k+1}) = d(t_k) + z(t_k)\, Y(t_{k+1}) \qquad (2.1.42)$$

By matrix inversion lemma the recursive parameter estimation algorithms to obtain the instrument variables $Y(t_k)$ are,

$$a(t_{k+1}) = a(t_k) + S(t_{k+1})z(t_k)\left[Y(t_{k+1}) - a^T(t_k)z(t_k) \right] \qquad (2.1.43)$$

$$S(t_{k+1}) = S(t_k) - S(t_k)z(t_k)z^T(t_k)S(t_k)\left[1 + z^T(t_k)S(t_k)z(t_k) \right]^{-1} \qquad \ldots (2.1.44)$$

The algorithms are initialised with

$$P(t_0) \triangleq I \quad (\text{unit matrix}); \quad a(t_0) = 0$$

$$X(t_j) = 0 \quad \text{for } j = 0, -1, -2, \ldots,$$

and $$\hat{\delta}(t_j) = 0 \quad \text{for } j = 0, -1, -2, \ldots).$$

2.1.6 Application of Recursive least Square Technique

In the foregoing discussion a comprehensive methodologies of recursive estimation have been presented. Whitehead and Young [28] have demonstrated the feasibility of constructing realistic dynamic stochastic water quality (BOD - DO) models for non-tidal river systems. The models are satisfactorily identified and statistically validated by reference to practical field data of flow BOD - DO in a 55 Km stretch of Bedford Ouse River System in England. Whitehead and Young [29], Young, Whitehead and Beck [30] have been able to demonstrate the particular utility of recursive methods of time series analysis both for identification and estimation of water resources systems models. The recursive algorithms have proved to be valuable aids for obtaining relatively efficient estimates of various model parameters in a straight forward and simple manner. Recursive algorithms provide a powerful general methods of data processing well suited to the modelling problem of water resources systems.

Sen Gupta and Chaudhuri [31] have used recursive least square nonstationary time series analysis technique for on-line forecasting of daily dissolved oxygen levels of a non-tidal river. Sen Gupta, Maulik and Chaudhuri have described an application [32] of the dynamic least square estimation algorithms for on-line modelling of dissolved oxygen levels of a non-tidal river passing through a highly industrialized region. The mathematical description of the dissolved oxygen levels allows for the real time monitoring of water quality. They have modelled the bio-chemical oxygen demand of a non-tidal river by recursive least square instrument variable algorithm. They have verified with observed data that recursive instrument variable [33] technique is amenable to on-line computation provided adequate real time data are available in time. They have used a separate parameter tracking algorithm for estimating the instrument variables. The present investigator has used this technique in the investigation.

Maulik, Sen Gupta and Chaudhuri have obtained [34] a simple dynamic model of daily flows of a non-tidal river by recursive least square non-stationary time series technique. They have also used recursive least square instrument variable algorithm [35] for on-line estimation of hourly flow of a non-tidal river. Instead of a separate parameter tracking algorithm they have estimated the instrument variables in the form of a memory sequence estimated apriori from an observed sequence of past data. The errors of the model are found to be quite high.

2.2.0 Annual Electrical Energy Consumption Model

In chapter IV a mathematical description of annual electrical energy consumption in India has been developed with population, gross national product, gross domestic saving and gross domestic capital formation as exogenous variables in the form of a polynomial of optimum complexity with the help of a learning identification technique known as multilayer group method of data handling algorithm.

2.2.1 Multilayer Group Method of Data Handling Algorithms

Ivakhnenko's [36] multilayer group method of data handling is a heuristic method of self-organisation of different partial models. This method involves the generation and comparison of all possible combinations of input output and to select the best possible ones according to the criterion of integral square error.

In multilayer group method of data handling algorithms, polynomials are used as the basic means of investigation of complex dynamical systems. The polynomials of prediction are regression equations which connect the current values of output with the current and/or past values of input variables. Regression analysis allows to evaluate the co-efficients of the polynomial by criterion of minimum mean square error. Then the polynomials are treated in the same manner as are seeds in the agricultural selection, an unique mathematical concept propagated and established by Academician A. G. Ivakhnenko and his co-workers [37,38,39] of the Institute of Cybernetics, Kiev, USSR.

Volterra series [40] introduced to non-linear system analysis by Wiener [41], learning filter of Gabor, Wilby and Woodcock [42] and the perception of Rosenblatt [43] have provided the conceptual basis for multilayer GMDH. Astrom and Eykhoff [44] pointed out that problems may arise with the use of volterra series or high degree polynomial to approximate non-linear functions because of the fact that there are many co-efficients to estimate, many data are needed and the computation with the resulting large matrices may be prohibitive. Ivakhnenko's multilayer GMDH algorithms are free of these problems. He models the input output relationships of complex processes using multilayer network structure of Rosenblatt's [45] perception type, who designed the model of brain's perception.

Salient features of multilayer GMDH as applicable in multilayer selection process which is used in the present investigation are briefly described here.

Each output element in the network implements a non-linear function of its inputs. The function implemented is usually a second order polynomial of its inputs. Since each element generally takes two inputs, the implemented function by an element in one of the layers is given by

$$Y = A_2(x) = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_1^2 + a_5x_2^2 \qquad (2.2.1)$$

Only those elements whose performance indices exceed the threshold at that layer are allowed to pass to the next layer. Therefore, the network represents a feed forward transformation whereby each succeeding layer in the network increases by two the degree of the multipolynomial fit to the input properties of x_i. Figure 2.2.1 depicts the concept of feed forward transformation.

The selection hypothesis employed by Ivakhnenko to select the elements to be used in the succeeding layers involves two basic conclusions ; the compositm character of a system must be based on the use of the signals which control the totality of the elements of the system, and the long history of the art of selection as observed in the case of plants and animals can to successfully extended to the science of engineering cybernetics.

Let us explain the two conditions. To get, say, plants in the agricultural sense with certain specific propentics, a large number of plants are sown which may have these properties, and the plants are crossed. From the harvest of the first generation, the plants are chosen which better our requirement (the first self selection) as compared to others.

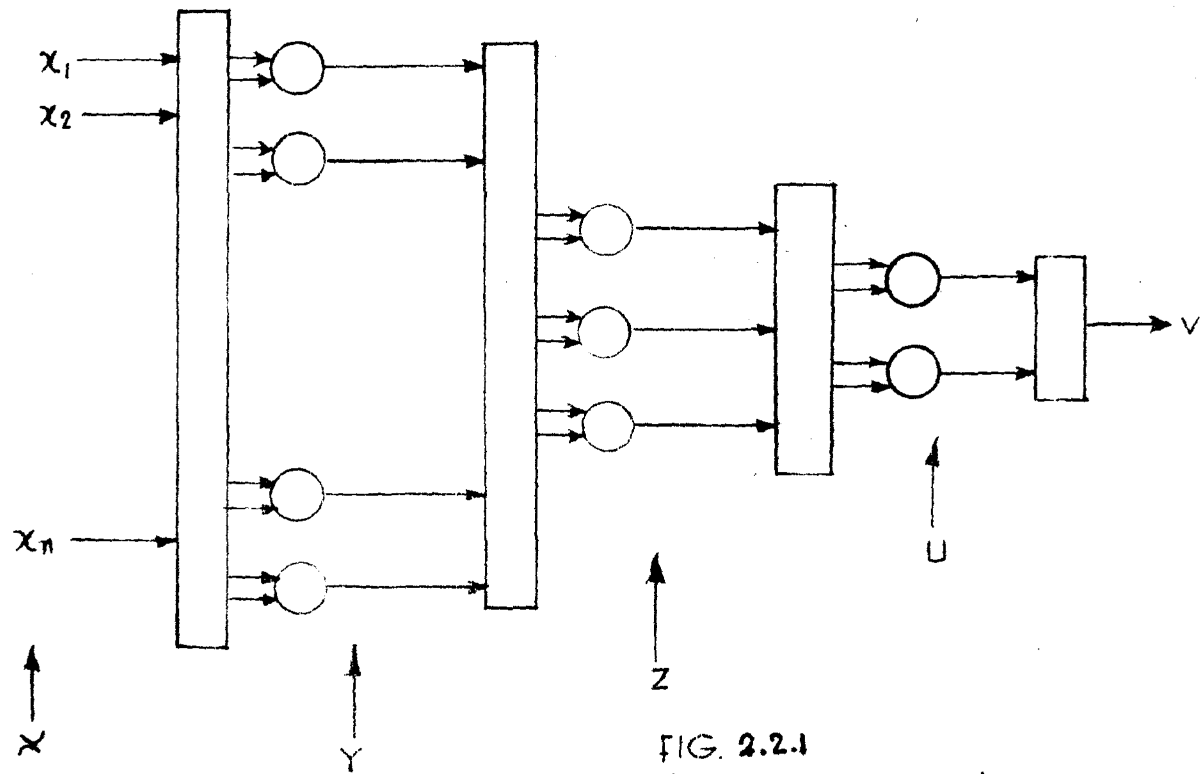


FIG. 2.2.1
Concept of feed forward transformation.

The seeds of the selected plants are again sown and crossed. From the second harvest we select certain seeds and the seeds are sown, and so on.
Rules employed in the process of mass selection are as follows :

i) For each generation certain optimal number of seeds are sown.

ii) The selection process cannot be completed in a single generation (at least 3 to 4 generations are needed).

Perception algorithms [43] of Rosenblatt duplicate the above mentioned process. Perception can be used for identification of extremal processes, in control theory sense. The complex surface of extremal hump is approximated by polynomials. The signals applied to the perception input contain information about the surface of interest to us. The surface is usually described by a number of experimental points and simple function of their co-ordinates. In accordance with the selection hypothesis, the simple polynomials of second degree that are easiest to inscribe in the surfaces are taken. The combination of data are subjected to the first threshold selection, in accordance with the integral square error criterion on a separate checking set. Only some of the polynomials which fit best into the sought surface are allowed to pass into the second layer where they form more complex combinations of polynomials of fourth degree. From the second layer again the polynomials which fit best into the sought surface are singled out and are allowed to pass into the third layer and so on. The process continues so long as minimum of a selection criterion is obtained. This constitutes Ivakhnenko's multilayer group method of data handling algorithms [46,47,48].

The co-efficients of each layer in the network are determined in the following manner.

Consider one element in the first layer. It implements the function $A_g(x)$ shown in equation (2.2.1). The data are divided into two sets — training and checking sets). Assume that these are N- input vectors in the training set each one of them is composed of p-property values.

$$X_n = (x_{n1}, x_{n2}, \ldots, x_{np})$$

$$n = 1,2, \ldots, N \qquad (2.2.2)$$

Denote the n th desired output as ϕ_n. A set of six co-efficients for the elements (which has inputs x_{ni} and x_{nj}) must be obtained so that the integral square error between the outputs of this element X_n and the true output ϕ_n is minimized. The co-efficients are obtained by solving Gauss Normal equations. The system of equations are written as

$$\phi_1 = a_0 + a_1 x_{1i} + a_2 x_{1j} + a_3 x_{1i} x_{1j} + a_4 x_{1i}^2 + a_5 x_{1j}^2$$

$$\vdots \qquad (2.2.3)$$

$$\phi_N = a_0 + a_1 x_{Ni} + a_2 x_{Nj} + a_3 x_{Ni} x_{Nj} + a_4 x_{Ni}^2 + a_5 x_{Nj}^2$$

in the matrix form $\phi = YA$ where ϕ, Y and A are $N \times 1$, $N \times 6$ and 6×1 matrices respectively (the first element of each row of x- matrix is unity). Vector A contains a set of six co-efficients which enables this element to be approximated to the true outputs in accordance with the method of least square. While estimating the co-efficients it has been intuitively assumed that the equation error is a white noise process with zero mean, constant variance and uncorrelated with inputs, and it is significantly small. The second assumption is that the inputs and outputs are exactly known without any measurement error [49].

This process is repeated for each element in the first layer with the components in matrix x changing each time depending on the identity of two inputs to the particular elements. The same technique is repeated to find the six co-efficients of each element in the succeeding layers. After the values of the co-efficients are obtained the performance index of a given element is determined by computing the integral square error between the output of each element and the true output on the whole data set.

Only those elements whose performance index are satisfactory are allowed to pass to the next layer. Figure 2.2.2 shows a flow chart of multilayer GMDH algorithm - the Ivakhnenko's theory of self-organisation.

Computational method for multilayer GMDH algorithms has been briefly described in the ensuing section.

The complete description of any process is given by

$$\phi = f\ (x_1, x_2, x_3, \ldots, x_n) \qquad (2.2.4)$$

The process is to be constructed of several layers of partial description of two input variables taken at a time.

$$y_1 = f(x_1, x_2),\ y_2 = f(x_3, x_4),\ \ldots,\ y_m = f(x_{n-1}, x_n)$$

$$m = \frac{n!}{2!\,(n-2)!} \qquad (2.2.5)$$

$$z_1 = f(y_1, y_2),\ z_2 = f(y_3, y_4),\ \ldots\ z_p = f(y_{m-1}, y_m)$$

$$p = \frac{m!}{2!\,(m-2)!}$$

and so on, where m and p are the number of pairwise combinations of first and second layer respectively.

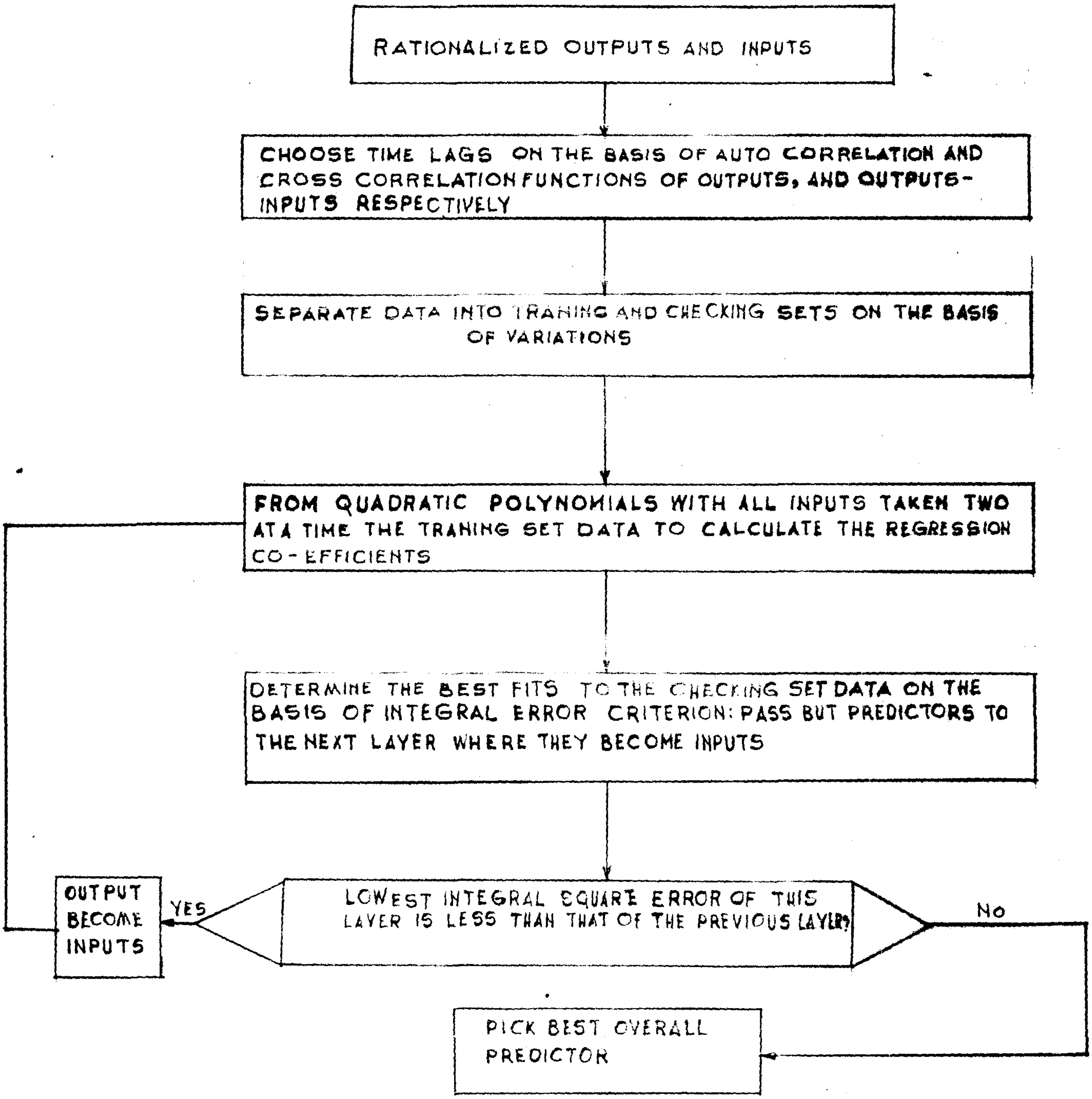


FIG. 2.2.2 FLOW CHART OF MULTILAYER GMDH ALGORITHMS.

Inputs which have strong correlation with the output are selected. Correlation functions are defined as

$$K_{yy}(\lambda) = \frac{\sum_{i=1}^{N-\lambda} (y(i) - \bar{y})\,(y(i+\lambda) - \bar{y})}{\left[\sum_{i=1}^{N-\lambda} (Y(i) - \bar{Y})^2 \sum_{j=1+\lambda}^{N} (Y(j) - \bar{Y})^2 \right]^{-\frac{1}{2}}} \qquad (2.2.6)$$

$$K_{yx}(\lambda) = \frac{\sum_{i=1}^{N-\lambda} (Y(i) - \bar{y})\,(x(i+\lambda) - \bar{x})}{\left[\sum_{i=1}^{N-\lambda} (Y(i) - \bar{y})^2 \sum_{j=1+\lambda}^{N} (x(j) - \bar{x})^2 \right]^{\frac{1}{2}}} \qquad (2.2.7)$$

where $K_{yy}(.)$ and $K_{yx}(.)$ are autocorrelation and cross correlation of output and output-input respectively for different lag λ , $\lambda = 0,1,2,\ldots N$; N = number of data points.

Data are rationalised in the form

$$x(k) = \frac{X(k) - X(min)}{X(max) - X(min)} \qquad (2.2.8)$$

where $X(k)$ is the actual value of data at the k-th instant of time.

The co-efficients of the first layer of partial description to given as shown in the equation

$$y_a = a_{oa} + a_{1a}x_b + a_{2a}x_c + a_{3a}x_bx_c + a_{4a}x_b^2 + a_{5a}x_c^2 \qquad (2.2.9)$$

where a, is the number of combinations and b,c are indices of combinations of input variables taken two at a time. The co-efficients are computed by solving a system of normal Gaussian equations. The left hand sides of the equation are set equal to the values of output at every points. After finding the values of the co-efficients the values of the intermediate variables are obtained. Then using the data set the integral square error between the intermediate variables and the true output is determined. Only the variables which give low error are selected for subsequent use. These variables are retained variables with high error figure are discarded. The number of intermediate variables should be kept same as the number of input variables. In the second layer of selection the co-efficients of the partial description,

$$z_a = b_{oa} + b_{1a}y_b + b_{2a}y_c + b_{3a}y_by_c + b_{4a}y_b^2 + b_{5a}y_c^2 \qquad (2.2.10)$$

of the layer are calculated and the accuracy is checked again to select the accurate intermediate variables of the layer. The process of selection continues so long as the integral square error comes to a minimum and in the next layer starts increasing. Thus multilayer GMDH comes to practical convergency.

The integral square error criterion is defined as

$$\sigma^2 = \frac{\sum_{i=1}^{N1} (\, Yi\ (observed) - Yi\ (model))^2}{\sum_{i=1}^{N1} (\, Yi\ (observed))^2} \qquad (2.2.11)$$

Every intermediate variable is examined for its effect on prediction accuracy. The training set is used for finding the co-efficients of the partial description, whereas the checking set is used to evaluate the quality of partial description. Thus multilayer GMDH has inherent decision regularisation.

Polynomial description of the process is obtained in the form of partial description of intermediate variables of different layers. Eliminating the intermediate variables the complete polynomial description of the process is obtained in the form of Gabor-Kolmogorov type of polynomial as

$$Y = a_0 + \sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j x_i x_j$$

$$+ \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k}^{n} a_i a_j a_k x_i x_j x_k + \ldots \qquad (2.2.12)$$

2.2.2 Application of Multilayer GMDH

With the help of multilayer GMDH algorithms Ivakhnenko obtained the polynomial description of British economy for prediction and control on the basis of characteristic variables established by Parks and Pyatt [50]. Heuristic self-organisation method proposed by Ivakhnenko in GMDH algorithms has been used in a modified form by Ikeda, Ochiai and Savargi [51,52] for developing a non-linear river flow model from the available data of river flows and mean areal precipitation. It is observed from the numerical comparisons made between the prediction model by GMDH and by elaborate hydraulic methods, that there are significant improvements in the heuristic prediction algorithms for real time computation. Tamura and Konda [53] have used GMDH algorithms for identifying spatial pattern of air pollution concentration in a large area. The heuristic GMDH algorithms have been used by Duffy and Franklin [54,55] to model an environmental system producing high nitrate level in agricultural drain water in the corn belt in the United States. The method amounts to fitting a polynomial th the multiinput single output response surface. They observed that the GMDH is advantageous with systems characterised by complexity with many variables and parameters, ill defined mathematical structures and limited data. These algorithms are useful for empirically generating hypothesis about which relatively little is known. R.K. Mehra [56] has employed GMDH for forecasting wheat crop yield using weather data. A comparison of the results with Baier [57] shows that crop prediction using GMDH compares favourably with the results obtained using theoretical-empirical models based on over ten years of research. The structural information obtained from GMDH as to which input variables have significant effect on wheat crop yield is also quite significant. Mehra has suggested the use of all data points alternatively as training and checking sets. This technique is expected to give good results. Chandhuri [58] has used GMDH for identification of the interactions of meteorological processes on monthly tea crop production. It is observed that multilayer GMDH gives good prediction results, identifies the significant variables, and gives an insight into the controlling aspects to adhere to a desired level of tea crop production.

Maulik, Sen Gupta and Chaudhuri [59] have developed a dynamic model for sixth hourly predistion of river flows, by multilayer group method of data handling algorithm, correlating the different up-stream flows and the rainfall at the different gauging stations in the catchment region of a river with the flow at the point of forecasting. They have also obtained a real time predistion model [60,61] for hourly flow at a point in a river system correlating the hourly flows at different gauging stations in the up-stream region. The models are found the simulate adequately the major variations observed in the field measurements. Sen Gupta, Maulik and Chaudhuri [62] have reported that the multilayer GMDH is quite capable of modelling on real time basis the dissolved oxygen levels, incorporating interacting parameters of a non-tidal river passing through a highly industrialised region.

2.3.0 Combinatorial Group Method of Data Handling

In chapter V the model of annual installed plant capacity of electrical energy of India has been obtained in the forms of polynomial of optimum complexity by computer aided self-organisation of mathematical models.

With the theory of self-organisation [63,64] using heuristic learning algorithm commonly known as group method of data handling it has been possible to formulate mathematical models for complex processes with prediction optimisation.

The concept of self-organisation can be illustrated as follows. When the model complexity gradually increases the computer finds by shifting the different models, the minimum of a selection criterion which the computer has been ordered to look for. Thus the computer indicates to the operator the model of optimum complexity.

2.3.1 Process Equation

The process equation has been developed from the illustration given in [66].

The physical process involved in a storm period is stochastic in nature. The process can be represented in the form of a finite order stochastic difference equation of the type as

$$y_{(k)} = f(x_1(k-n),\ x_2(k-m),\ x_3(k-p),\ldots) \qquad (2.3.1)$$

$n, m, p, \ldots,$ are the instants of the

$x_1,\ x_2,\ x_3,\ \ldots,$ respectively

which have highest correlation with $y_{(k)}$. We write

$y_{(k)} = y,$ flow at the k-th instant

$x_1(k-n) = x_1,$ flow at a up-stream gauging station 1 which has highest correlation at lag instant n

$x_2(k-m) = x_2,$ flow at a up-stream gauging station 2 which has highest correlation at lag instant m, and so on.

Let us assume

$x_i(k-j)$ as $i = 1,2,3,4,5$; $j = m,n,p,q,r$

so the process equation becomes

$$y = f(x_1,\ x_2,\ x_3,\ x_4,\ x_5) \qquad (2.3.2)$$

The function f(.) in equation (2.3.2) is sought in the class of quadratic polynomials on the basis of a table of polynomials of gradually increasing complexity of eight variables as shown in Table 2.3.1 with the theory of self-organisation of different mathematical models.

The model of optimum complexity is selected on the basis of minimum of integral square criterion. Integral square error is defined as

$$I^2 = \frac{\sum_{i=1}^{N} (y_{tab}(i) - y_{d.m}(i))^2}{\sum_{i=1}^{N} (y_{tab}(i))^2} \qquad (2.3.3)$$

where $y_{tab}(i)$, $i = 1,2,\ldots, N$ hours, are the tabulated values of the output variable in the interpolation region and $y_{d.m}(i)$ are the values of the variable obtained from the model.

2.3.2 Application of Combinatorial Group Method of Data Handling Algorithm

Chaudhuri [64] has used combinatorial GMDH algorithm to obtain the medium term and long term prediction models of annual Indian tea production. Different types of models of

TABLE 2.5.1

POLYNOMIALS OF GRADUALLY INCREASING COMPLEXITY OF EIGHT VARIABLES

$y = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_3 + \ldots$ [illegible]

x_1	x_2	x_1x_2	x_3	x_1x_3	x_3x_4	x_8
$y_1 = a_0 + a_1x_1$	$y_2 = a_0 + a_2x_2$; $y_3 = a_1 + a_1x_1 + a_2x_2$	$y_4 = a_0 + a_1x_1x_2$; $y_5 = a_0 + a_1x_1 + a_2x_1x_2$; $y_6 = a_0 + a_1x_2 + a_2x_1x_2$; $y_7 = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2$	[illegible] $\ldots + a_4x_3$	$y_{26} = a_0 + a_1x_1x_3$; $y_{27} = a_0 + \ldots$	$y_{312} = a_0 + \ldots$ [illegible] $\ldots + a_{10}x_3x_4$	$\ldots$; $y_{2^{36}-1} = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + \ldots + a_{36}x_7x_8$

The General rule of total number of combination is 2^n-1 where n is the total number of terms containing the variables in the general form of the polynomial.

polynomials of increasing complexity have been tested. The polynomials which give minimum of a selection criterion have been found. It is found that annual tea crop production is a nonstationary process. It is observed that the law of annual tea crop production varies with time. Maulik, Sen Gupta and Chaudhuri [65] have obtained a simple dynamical model of hourly flow of a river with a minimum of sixth lag instance in the measurements of up-stream flows at different up-stream gauging stations using combinatorial GMDH algorithm. The model has been verified by simulation against field data. Sen Gupta, Maulik and Chaudhuri [66] have obtained a dynamic model of optimum complexity for daily forecasting of dissolved oxygen levels of a non tidal river with the help of the combinatorial GMDH technique. The model has been verified by field measurement of the dissolved oxygen collected over a 80 day period from the river Cam in Eastern England. The distinct periodicity has been observed in the daily dissolved oxygen levels. The sinusoidal terms have been incorporated in the polynomial model.

2.4.0 States Estimation of Electrical Power System

The realiable operation of a power network depends on a real time data base for monitoring, security and control of power system. States estimation programme can enhance the data base for on line real time operation of power networks. The basic function of an estimation programme is to convert the telemetered raw measurement data into a reliable information base. The information base contains all complex bus voltages, power and current flows as well as injunctions along with the network status and parameters errors.

2.4.1 A Brief Review of Currently Available States Estimators

The power system state estimation results from two big fields, load flow analysis [67] and estimation theory. F.C. Schweppe [68,69,70,71] appears to be first scientist to initiate the application of the modern control technique of state estimation, detection and parameter identification, originally developed for aerospace systems, to meet the power system needs. Dopazo et.al. [72,73,74] have developed A.E.P. State estimators for real time monitoring of power system state variables. Arafeh et.al [75] have given a good converage providing assessment and comparison of different state estimation techniques.

The weighted least square method is the general basis of state estimation algorithms actually used. An iterative procedure based on Newton-Raphson's method is used to achieve convergence of the state variables.

The state estimation algorithms are divided into two categories, namely, static state estimation algorithm and tracking state estimation algorithm. Static state estimation is defined [77] as the process of computing the network node voltages which are the states of the power systems from a set of measurements made upon the network at a sampling instant (i.e., snap shot measurements). The set of measurements include active and reactive node injections or line flows current and voltages etc. In real time on-line operation quasi-dynamic tracking state estimator is used for recursive estimation of the state variables. Recursive estimation is a process of updating the estimated state each sampling instant on the receipt of fresh information.

In the method presented in chapter IV a recursive type tracking algorithm [76] is used. Though a snap shot of measurements is considered to illustrate the application of the developed method in an iterative sequence, the method is quite usable for on-line discrete time control operation for the power network. In chapter VI a complete derivation of the power system states estimator with a tracking algorithm has been given in details with the necessary illustration.

2.5.0 Gauss-Seidel Load Flow with Optimally Ordered Nodes by Dynamic Programming Algorithm

It is desired that transmission system should be able to transmit electric energy economically and reliably from generation centres to all load centres at a generally acceptable voltage level. This necessiates the study of the load flow in a power system to determine steady operating states. Results of the load flow analysis are used for stability analysis and for power system planning operation and control. A large number of numerical algorithms have been developed over the last 25 years [78,67]. The most of the algorithms are variations of two numerical technique such as (i) Gauss-Seidel method and (ii) Newton Raphson method. The present effort is an exposure of the Gauss-Seidel method under different bus conditions with optimal ordering of buses by Dynamic Programming algorithm.

2.5.1 Optimal Ordering of Nodes

The computational efficiency of load flow analysis depends on the order in which the Gaussian elimination is performed on sparse matrices and total number of new non zero elements are generated in course of elimination. It is observed that the computational efficiency is greatly improved if the nodes are ordered in an optimal way.

The Principle of solution of sparsity oriented node ordering problem can be stated as follows [79,80,81,82,83,84,85].

An initial segment of an optimal ordering is a group of nodes of a network which has the property that their optimally ordered elimination of the remaining nodes in a network constitutes an optimally ordered elimination of all the nodes in a network.

The principle of optimality as stated above is applied to the problem of optimal ordering of sparsity oriented nodes in power system network. This optimisation problem is solved in an iterative procedure by Dynamic Programming algorithm [86,87] following an optimum decision policy. The objective of the sparsity oriented optimum ordering of nodes is to determine the best possible way of performing Gaussian elimination, so that the amount of fill in or the valency of the elimination is minimum ; the valency of a node is the number of new paths added among the remaining set of nodes as a result of elimination of the node and the valency of an ordering is the total number of new paths generated in the process of performing the node elimination in the order specified.

In chapter VII a complete derivation of the dynamic programming algorithm has been given with an illustration on IEEE 14 BUS system.

CHAPTER III

ON-LINE SIMULATION OF HOURLY RIVER FLOWS FOR RUN-OF-THE RIVER HYDRO-ELECTRIC PLANT

CHAPTER III

ON-LINE SIMULATION OF HOURLY RIVER FLOWS FOR RUN-OF-THE RIVER HYDRO-ELECTRIC PLANT

3.0 Introduction

Assessment of the present state of the hydropower generation in India indicates that the huge water power of the rivers flowing through the Himalayan regions is virtually untapped. A realistic estimate of the availability of water power of the river Teesta in North Bengal in the Himalayan region in the up-stream of Coronation Bridge point as a single generating station of the run-of-the river type suggests the generation in the range of 1000 MW. Complexities of the hydrological characteristics of the Teesta watershed region in the mountain necessitates the on-line monitoring of hourly river flow and real time control of hydraulic structures of the power plant in conjunction with the monitored variables.

Computer based digital supervisory system needs a highly realistic hourly flow simulation model to be incorporated in the ROM of the respective processor. Small general purpose microcomputer can be used as peripheral controllers with preprogrammed estimation algorithm in conjunction with computer control supersystem. The present investigation develops the hourly flow simulation technique with the cybernetical method of recursive least square instrument variable algorithm with parameter tracking adaptiveness in the estimation of instrument variables. The effectiveness of the developed technique has been demonstrated in obtaining an on-line simulation model of the hourly flows of the hilly river Teesta in North Bengal at Coronation Bridge Point on the basis of the observed data at different up-stream gauging stations.

3.1.0 Modelling of Hourly River Flows by Recursive Least Square Instrument Variable Algorithm

3.1.1 Development of Recursive Algorithms

The recursive technique has been defined as a technique in which an estimate is updated on receipt of fresh information.

Systems Dynamic Equation

A strong correlation of the down-stream flows with that at up-stream flows, particularly at the confluence of the tributaries, suggests that the process may be represented in the form of a non-stationary time series with a probability that the current value of the output $Y(t_k)$ is a function of the previous output observations, the autoregressive terms $Y(t_{k-1})$, $Y(t_{k-2})$, ..., the past values of the highly correlated deterministic inputs

$$U_1(t_{k-r1}), U_2(t_{k-r2}), \ldots, U_1(t_{k-r1-1}), U_2(t_{k-r2-1}), \ldots$$

together with a current unknown realization of the noise process $\eta(t_k)$. The process can be represented as

$$Y(t_k) = \sum_{i=1}^{n} \beta_i Y(t_{k-i}) +$$

$$\sum_{j=1}^{m} \sum_{i=0}^{n} \delta_{ji} U_j(t_{k-rj-i}) + \eta(t_k) \qquad (3.1.1)$$

Determination of n is known as the model order determination. It is suggested that the output correlation is an intuitive consideration for model determination which is also the model structure identification.

With backward shift operator $q^{-1}(.)$ defined as $q^{-1}Y(t_k) = Y(t_{k-1})$, equation (3.1.1) transforms to

$$Y(t_k) = \left[\sum_{i=1}^{n} \beta_i q^{-i} \right] Y(t_k) +$$

$$\sum_{j=1}^{m} \left[\sum_{i=0}^{n} \delta_{ji} q^{-i} \right] U_j(t_{k-rj-i}) + \eta(t_k) \qquad (3.1.2)$$

Here, $\eta(t_k)$ may be expressed in a moving average sequence as

$$\eta(t_k) = \sum_{p=1}^{P} \eta(t_{k-p}) + v(t_k) \qquad (3.1.3)$$

where $v(t_k)$ is a white noise innovation process with

$$\xi \{ v(t_k) \; v(t_j) \} = 0 \quad \text{for } j \neq k$$

$$\xi \{ v(t_k) \; v(t_j) \} \simeq \rho^2 \quad \text{for } j = k \qquad (3.1.4)$$

This model is quite flexible since it requires that the equations should be linear in parameters.

Equation (3.1.2) may be represented as

$$Y(t_k) = z^T(t_k) \alpha + \eta(t_k) \qquad (3.1.5)$$

where α is the parameter vector with the property of slowly varying with time and amenable to recursive adaptiveness, and

$$z^T(t_k) = \left[Y(t_{k-1}),\ldots, Y(t_{k-n}), U_1(t_{k-r_1}),\ldots \right.$$

$$\left. U_1(t_{k-r_1-n}),\ldots,U_m(t_{k-r_m}),\ldots,U_m(t_{k-r_m-n}) \right] \qquad (3.1.6)$$

where r_j, $j = 1,2,\ldots\ldots\ldots, m$ stands for the lag time instant of up-stream flows which have the strongest correlation with the down-stream flow $Y(t_k)$.

3.1.2 Least Square Estimation of Parameters

The parameter values of equation (3.1.5) are estimated by minimising a loss function defined as the sum of the square errors as

$$J \triangleq \sum_{k=1}^{N} \left[Y(t_k) - z^T(t_k)\hat{\alpha} \right]^2 \qquad (3.1.7)$$

The estimates $\hat{\alpha}$ of α that minimise J are called least squares estimates. Thus, for minimization,

$$\frac{\delta J}{\delta \hat{\alpha}} = 0$$

$$= 2\left[\sum_{k=1}^{N} z(t_k)z^T(t_k) \right] \quad -2\left[\sum_{k=1}^{N} z(t_k)Y(t_k) \right]$$

Hence the well known equations for least squares parameter estimation as

$$\hat{\alpha} = \left[\sum_{k=1}^{N} z(t_k) z^T(t_k) \right]^{-1} \left[\sum_{k=1}^{N} z(t_k) Y(t_k) \right] \qquad (3.1.8)$$

3.1.3 Algorithms for Recursive Least Square Estimation of Parameters

If it is assumed that the estimate $\hat{\alpha}$ of α is a slowly varying function of time, the new value of $\hat{\alpha}$ defined as $\hat{\alpha}(t_k)$ will appear as each information is serially processed recursively.

The algorithms for least square recursive estimation of parameters have been obtained as

$$\hat{\alpha}(t_k) = \hat{\alpha}(t_{k-1}) + P^*(t_{k-1})\, z(t_k) \left[1 + z^T(t_k) P^*(t_{k-1}) z(t_k) \right]^{-1} \left(Y(t_k) - z^T(t_k)\, \hat{\alpha}(t_{k-1}) \right) \qquad (3.1.9)$$

$$P^*(t_k) = P^*(t_{k-1}) - P^*(t_{k-1}) z(t_k) \left[1 + z^T(t_k) P^*(t_{k-1}) z(t_k) \right]^{-1} z^T(t_k) P^*(t_{k-1}) \qquad (3.1.10)$$

with $$P^*(t_k) \triangleq \left[\sum_{j=1}^{k} z(t_j) z^T(t_j) \right]^{-1}.$$

From least square estimation $\hat{\alpha}(.)$ and $P^*(.)$ may be initialized for a block of data or for the whole data set.

The parameter estimation algorithms of equations (3.1.9) and (3.1.10) do not overcome the problem of bias. The conditions for unbiased estimates have been stated in equation (3.1.4). For most cases of practical interest, $v(t_k)$ is not a white Gaussian sequence and estimates of α are not unbiased.

To overcome the problem of bias many variants of recursive parameter estimation algorithms have been suggested. Of which the recursive instrument variable algorithms are found to be easily amenable to simulation with rapid convergent properties.

3.1.4 Recursive Instrument Variable Algorithms for Parameter Estimation

The most likely source of biased estimate is the presence of significant noise sequence correlation between noise sequence and the past values of the output.

Referring to equation (3.1.2) with suitable estimates of the parameters, an auxiliary time series model can be computed as

$$\hat{Y}(t_k) = \left[\sum_{i=1}^{n} \beta_i q^{-i} \right] \hat{Y}(t_k) +$$

$$\sum_{j=1}^{m} \left[\sum_{i=0}^{n} \delta_{ji} q^{-i} \right] U_j(t_{k-r_j-i}) \qquad (3.1.11)$$

Equations (3.1.2) and (3.1.11) suggest that any variation in $\hat{Y}(t_k)$ should be strongly correlated with variations in the noise corrupted output observations $Y(t_k)$, but the variations in $\hat{Y}(t_k)$ should be uncorrelated with $\eta(t_k)$ provided $\eta(t_k)$ is not correlated with the measured input sequences $U_j(t_{k-r_j})$, i.e.,

$$\mathcal{E}\left\{ U_j(t_{k-r_j})\, \eta(t_l) \right\} = 0 \quad \text{for all } j, k \text{ and } l.$$

The sequence $\hat{Y}(t_k)$ is called the sequence of instrument variables. Consequently, the vector $z^T(t_k)$ is modified as $\hat{z}^T(t_k)$, defined by

$$\hat{z}^T(t_k) = \left[\hat{Y}(t_{k-1}), \ldots, \hat{Y}(t_{k-n}), U_1(t_{k-r_1}), \ldots, U_1(t_{k-r_1-n}), \ldots, U_m(t_{k-r_m}), \ldots, U_m(t_{k-r_m-n}) \right]$$

The conditions of unbiased estimates are modified as

$$\mathcal{E}\{z(t_k)\ \eta(t_k)\} = 0 \quad \text{for all} \quad k$$

with $$\mathcal{E}\{v(t_k)\ v(t_j)\} = 0 \quad \text{for} \quad k \neq j$$

$$= \rho^2 \quad \text{for} \quad k = j$$

Replacing $z(t_k)$ by $\hat{z}(t_k)$ and not $z^T(t_k)$ by $\hat{z}^T(t_k)$, the recursive instrument variable algorithms are given as

$$\hat{\alpha}(t_k) = \hat{\alpha}(t_{k-1}) + \hat{P}^*(t_{k-1})\,\hat{z}(t_k)\,[1 + z^T(t_k)\,\hat{P}^*(t_{k-1})\,\hat{z}(t_k)]^{-1}\,\{Y(t_k) - z^T(t_k)\,\hat{\alpha}(t_{k-1})\}$$

$$\hat{P}^*(t_k) = \hat{P}^*(t_{k-1}) - \hat{P}^*(t_{k-1})\,\hat{z}(t_k)\,[1 + z^T(t_k)\,\hat{P}^*(t_{k-1})\,\hat{z}(t_k)]^{-1}\,z^T(t_k)\,\hat{P}^*(t_{k-1}) \qquad (2.1.12)$$

with

$$\hat{Y}(t_k) = \left[\sum_{i=1}^{n} \beta_i\, q^{-i}\right] \hat{Y}(t_k) + \sum_{j=1}^{m} \left[\sum_{i=0}^{n} \delta_{ji}\, q^{-i}\right] U_j(t_{k-rj-1})$$

and $$\hat{P}^*(t_k) \triangleq \left[\sum_{j=1}^{k} \hat{z}(t_j)\,z^T(t_j)\right]^{-1}.$$

$\hat{\alpha}(.)$ and $\hat{P}^*(.)$ have been initialised by the least square method using the whole data set or a block of data.

The instrument variables $\hat{Y}(.)$ in $\hat{z}(.)$ are obtained through a separate parameter tracking algorithm as detailed below.

$$Y(t_k) = \sum_{i=1}^{n} \beta_i Y(t_{k-i}) + \sum_{j=1}^{m} \sum_{i=0}^{n} \delta_{ji} U_j(t_{k-rj-i})$$

$$+ \sum_{q=1}^{Q} C_{t_{k-q}} (Y(t_{k-q}) - \hat{Y}(t_{k-q})) + \sigma(t_k) \qquad (3.1.13)$$

where the third component is the moving average component, $\sigma(t_k)$ is the error sequence.

$\hat{Y}(t_k)$, the estimate of $Y(t_k)$ can be written as

$$\hat{Y}(t_k) = a^T(t_{k-1})\, z(t_{k-1}) \qquad (3.1.14)$$

where

$$a^T(t_{k-1}) = \left[\beta_1, \beta_2, \ldots, \delta_{10}, \delta_{11}, \ldots, \ldots, \delta_{mn}, C_{t_{k-1}}, C_{t_{k-2}}, \ldots \right]$$

$$z(t_{k-1}) = \left[Y(t_{k-1}), Y(t_{k-2}), \ldots, U_1(t_{k-r1}), \ldots, \ldots, U_m(t_{k-rm-n}) \right] \qquad (3.1.15)$$

The co-efficient vector 'a' can be estimated by minimizing the quadratic performance criterion $J_k(a)$, defined as,

$$J_k(a) \triangleq \sum_{j=1}^{k} (Y(t_j) - a^T z(t_{j-1}))^2 +$$

$$+ (a - a(t_0))^T S^{-1}(t_0) (a - a(t_0)) \qquad (3.1.16)$$

where $a(t_0)$ is the available a priori estimate of the co-efficient vector 'a' and $S(t_0)$ is the positive definite weighting matrix of the order $ml \times ml$ where $ml = n+m(n+1)+Q$.

For minimization,

$$\frac{\delta J_k(a)}{\delta a} = -2 \sum_{j=1}^{k} z(t_{j-1}) (Y(t_j) - a^T z(t_{j-1})$$

$$+ 2 S^{-1} (t_0) (a - a(t_0)) \qquad (3.1.17)$$

It follows from equation (3.1.17)

$$\sum_{j=1}^{k} z(t_{j-1}) Y(t_j) + S^{-1}(t_0) a(t_0) = \sum_{j=1}^{k} z(t_{j-1}) z^T(t_{j-1})$$

$$+ S^{-1}(t_0)\, a \qquad (3.1.18)$$

Let

$$S^{-1}(t_k) = \sum_{j=1}^{k} z(t_{j-1})z^T(t_{j-1}) + S^{-1}(t_0) \qquad (3.1.19)$$

and

$$d(t_k) = \sum_{j=1}^{k} z(t_{j-1})Y(t_j) + S^{-1}(t_0)a(t_0) \qquad (3.1.20)$$

Denoting the co-efficient vector 'a' as a (t_k) at the time instant t_k,

$$S^{-1}(t_k)\, a(t_k) = d(t_k)$$

or $\qquad (3.1.21)$

$$a(t_k) = S(t_k)\; d(t_k).$$

From equations (3.1.19) and (3.1.20) the following recursive equations are obtained,

$$S^{-1}(t_{k+1}) = S^{-1}(t_k) + z(t_k)z^T(t_k) \qquad (3.1.22)$$

$$d(t_{k+1}) = d(t_k) + z(t_k)\, Y(t_{k+1}) \qquad (3.1.23)$$

By matrix inversion lemma the recursive parameter estimation algorithms to obtain the instrument variables $Y(t_k)$ are,

$$a(t_{k+1}) = a(t_k) + S(t_{k+1})z(t_k)\left[Y(t_{k+1}) - a^T(t_k)z(t_k)\right] \qquad (3.1.24a)$$

$$S(t_{k+1}) = S(t_k) - S(t_k)z(t_k)z^T(t_k)S(t_k)\left[1 + z^T(t_k)S(t_k)z(t_k)\right]^{-1}$$

... (3.1.24b)

The algorithms are initialised with

$$S(t_0) \stackrel{\Delta}{=} I \quad (\text{ unit matrix }) \; ; \quad a(t_0) = 0$$

$$Y(t_j) = 0 \quad \text{for} \quad j = 0, -1, -2, \ldots,$$

and $$\hat{S}(t_j) = 0 \quad \text{for} \quad j = 0, -1, -2, \ldots).$$

3.2.0 Details of Investigation Sites

The river Teesta, its catchment and the observation sites are described below.

3.2.1 The Teesta, its Catchment and the Observation Sites

The river Teesta rising from the Himalayan ranges in north Sikkim and passing through the deep gorges for nearly 138 Kms. debouches upon the plain of West Bengal near Sevoke. The Teesta is a very fast flowing river. Its average velocity is 6.2 metre per second. In winter its water is seagreen. In summer and in rainy season when the ice in the glacier melts quickly and when its catchment is bathed in torrential rains the milky white water flows through the river surging its narrow Himalayan fluvial course.

The accompanying map, Fig. 3.2.1 depicts the Teesta river and its catchment. A brief description of the river is given. The river Lohnak originates from the snow line in North Sikkim at a height of about 6401 metres.

The river Zemu originates from the Zemu glacier at a height of 4968 metres. These two rivers combine at Lachen, after which it is known as the Zemuchu river. At Chungthang Lachen Chu river joins the Zemu Chu from the north eastern side. This combined flow is further augmented by the river Lachung Chu at down-stream of Chungthang to form the river Teesta. Thus the Teesta, in Bengali Language called Trisrota meaning that three flows have combined together, is formed by the rivers Zemu Chu, Lachen Chu and the Lachung Chu. At Sankalan the river Talung Chu originating from the Talung gracier in north western Sikkim at a height of about 5873 metres, joins the river Teesta. Upto Sankalan length of the river from the origin is about 70 K.m. and the catchment area is about 4200 sq. K.m.

From Sankalan the river Teesta flows through the narrow Himalayan gorges and comes to Singtam. About 15 K.m. up-stream of Singtam the river Dik Chu joins the Teesta. At Singtam from the easternside the river Rongni Chu joins the Teesta. Upto Kantitar the length of the river is approximate 114 K.m. from the origin and the catchment area is approximate 4874 sq.K.m. from the origin.

At Rongpo the river Rongpo Chu from the eastern catchment region joins with the Teesta. The length of the river from the origin upto Rongpo is approximate 116 K.m. and the total catchment area of the river including its tributaries upto Rongpo is approximately 5405 sq.K.m. from the origin.

Near Singlabazar the river Great Rangit combines with the river Raman and the river Little Rangit and flows as the Great Rangit river. This combination of three rivers bring in an addition of about 1966 sq. K.m. of catchment area to the Teesta catchment. At about 3 K.m. up-stream of Teestabazar the Great Rangit joins the Teesta. The confluence of the Great Rangit and the Teesta is unforgettable.
The clear sea green water of the Great Rangit mixes with the milky white water of the Teesta and thus creating a wonderful cocktail of nature. The length of the river from the origin upto Teestabazar is about 134 K.m. and the catchment is (approx.) 7714 sq. K.m.

Upto Coronation Bridge the length of the Teesta from the origin is (approx.) 188 K.m. and the catchment area is about 8147 sq.K.m. Upto Sevoke its length from the origin is (approx.) 160 K.m. and the catchment area is about 8179 sq. K.m.

In the plains the important tributaries of the Teesta are the Lish, Ghish, Chel, Neora and the Karla. Upto Domohani Road Bridge the length of the Teesta from the origin is (approx.) 206 K.m. and the catchment area is (approx.) 9432 sq. K.m.

The Teesta mixes with the Brahmaputra.

3.2.2 The Main Observation Stations

The main observation stations from where the data for the investigation reported in this chapter were collected have been shown in the map. A brief description of the stations are given below.

Sitewise brief note of various observation stations

1. **SANKALAN BRIDGE**

a) Location	:	Lat.-27° 30'N, Long.-88° 38'E, on river Teesta in hilly terrain of North Sikkim down-stream of confluence of Lachenchu, Lachungchu and Talungchu. National Highway 31 A is about 3 K.m. from the site.
b) Nature of Station	:	Gauge and Discharge with 25 Watt H.F. Wireless facilities.
c) Mode of observation	:	Hourly gauge observed round the clock during monsoon. Discharge observation thrice a day at 0800, 1200 and 1800 hrs. using wooden float and a float run of 30 m. Cross-section taken once a month using sounding of 15 Kg.wt.

d) Length of river from origin up to site : 70 K.m. (approx.)

e) Catchment area up to site from origin : 4900 sq. K.m. (approx.)

f) Date of commencement of gauge/discharge : 12.10.72/ 2.12.72

g) Maxm. observed gauge/ Discharge during monsoon of 1979 : 759 m. on 2.7.79 (1900 Hrs.) / 1432.64 Cumecs on 12.7.79 (1800 hrs.)

h) Maxm. ever recorded gauge/Discharge : 762.30 m on 12.6.73 (0100 hrs.) / 1729.07 Cumecs on 18.7.73 (0700 hrs.)

i) Type of Raingauge : NIL.

2. <u>KHANITAR</u>

a) Location : Lat.- 27° 10.5'N, Long.- 88° 30'E in Sikkim on Teesta. NH-31 A is about 1 K.m. from the site.

b) Nature of Station : Gauge and Discharge with 15 Watt. H.F. wireless facilities.

c) Mode of observation : Hourly gauge observed round the clock during monsoon. Discharge observation thrice a day at 0800, 1200 and

1600 hrs. using Wooden float and a float run of 30 M. Cross-section taken once a week using sounding of 25 Kg.wt.

d) Length of river from origin upto site : 114 K.m. (approx.)

e) Catchment area upto site from origin : 4874 Sq.K.m. (approx.)

f) Date of commencement of observation of Gauge/ Discharge : 12.6.70/12.6.70

g) Maxm.observed gauge/Discharge during monsoon of 1979 : 296.79 m.on 2.7.79 (2000 hrs.) / 1784.49 Cumecs on 23.7.79 (0700 hrs.)

h) Maxm. ever recorded Gauge/Discharge : 297.50 m.on 12.10.73 (0500 hrs.) / 2197.74 Cumecs on 18.6.79 (1200 hrs.)

i) Type of raingauge : One S.R. Raingauge and one ordinary raingauge.

3) **RONGPO**

(i) **ON RIVER RANGPO CHU :**

(a) Location : Lat.-27° 10' N, Long.-88° 32' E, on river Rongpo chu i.e. tributary of river Teesta near the junction of L.R.P. Road at NH - 31 A is about 1 K.M. from the site.

(b) Nature of Station : Gauge and Discharge with 15 Watt. H.F. Wireless facilities.

(c) Mode of observation : Hourly Gauge observed round the clock during monsoon. Discharge observation thrice a day at 0800, 1200 and 1600 hrs. using wooden float and a float run of 30 m. Cross-section taken once a week using sounding of 20 Kg.wt.

(d) Date of commencement of observation of Gauge/Discharge : 5.6.70/ 1.7.70

(e) Maxm.observed Gauge/Discharge during monsoon of 1979 : 304.608 Mtr.on 3.8.79 (0800 hrs.) / 603.25 Cumecs on 23.7.79 (1200 hrs.)

(f) Maxm. ever recorded
Gauge/Discharge : 304.608 Mtr.on 3.8.79 (0300 hrs.) / 732.016 Cumecs on 26.9.78 (1200 hrs.)

(g) Type of Raingauge : One ordinary raingauge.

4) SINGLABAZAR

(a) Location : Lat.- 27° 07' N and Long. 88° 14' E, on river Great Rangit i.e. tributary of river Teesta, is 2 K.m. from the site.

(b) Nature of Station : Gauge with 15 Watt. H.F. Wireless facilities.

(c) Mode of observation : Hourly gauge observed round the clock during the monsoon.

(d) Date of commencement of observation of Gauge/Discharge : 11.12.69

(e) Maxm. observed Gauge/Discharge during monsoon '79 : 310.00 Mtr. on 24.7.79 (0300 hrs.)

(f) Maxm. ever recorded Gauge/Discharge : 310.35 Mtr. on 12.10.73 (2100 hrs.)

(g) Type of raingauge : One S.R. Raingauge and one ordinary raingauge.

(ii) <u>ON RIVER RAMAN</u>

(a) Location : Lat.-27°7.5'N, Long. 88° 18.5'E on river Raman i.e. tributary of river Teesta, is 1 K.m. from the site.

(b) Nature of Station : Gauge and Discharge.

(c) Mode of Observation : Hourly gauge observed round the clock during monsoon. Discharge observation thrice a day at 0800, 1200 and 1600 hrs. using wooden float and a float run of 30 Mtr. Cross-section taken once a week using sounding of 25 Kg.wt.

(d) Length of river from origin upto site : 32 K.m. (approx.)

(e) Catchment area upto site from origin : 385 Sq.K.m. (approx.)

(f) Date of commencement of observation of Gauge/Discharge : 11.12.69/ 1.6.73

(g) Maxm. observed Gauge/Discharge during monsoon of 1979 : 320.00 Mtr.on 29.7.79 (1000 hrs.) / 140.63 Cumecs on 29.7.79 (1200 hrs.)

(h) Maxm. ever recorded Gauge/Discharge : 320.15 Mtr. on 12.10.73 (2200 hrs.) / 421.488 Cumecs on 16.8.78 (0700 hrs.)

(i) Type of Raingauge : NIL.

(iii) ON RIVER LITTLE RANGIT

(a) Location : Lat.- $27^{\circ}\ 5\frac{5}{6}$ 'N, Long. $88^{\circ}\ 15\frac{1}{2}$ 'E on river Little Rangit i.e. tributary of river Teesta, is about 3 K.m. from the site.

(b) Nature of Station : Gauge and Discharge with Non-exchange telephone facilities with Singlabazar.

(c) Mode of observation : Hourly gauge observed round the clock during monsoon. Discharge observation thrice a day at 0800, 1200 and 1600 hrs. using wooden float and a float run of 30 Mtr. Cross-section taken once a day using sounding of 25 Kg.wt.

(d) Length of river from origin upto site : 35 K.m. (Approx.)

(e) Catchment area upto site from origin : 184 Sq. K.m. (Approx.)

(f) Date of commencement of observation of Gauge/Discharge : 21.6.73/ 1.7.73.

(g) Maxm. of observed Gauge/Discharge during monsoon of 1979 : 321.53 Mtr.on 8.8.79 (2400 hrs.) / 83.60 Cumecs on 30.8.79 (0700 hrs.)

(h) Maxm. ever recorded Gauge/Discharge : 314.30 Mtr.on 12.10.73 (1800 hrs.) / 868.57 Cumecs on 7.9.75 (1700 hrs.)

(i) Type of Raingauge : NIL.

5) <u>TEESTABAZAR</u>

(a) Location : Lat.- 27° 03' N, Long. 88° 25' E on river Teesta down-stream of the confluence of the Great Rangit with river Teesta NH-31A is about 3 K.m. from the site.

(b) Nature of Station : Gauge and Discharge with 15 Watt. Wireless facilities observation made on the suspension bridge on river Teesta on NH-31A. Also having non-exchange telephone line between camp shed and bridge point.

(c) Mode of observation : Hourly Gauge observed round the clock during the monsoon. Discharge observation thrice a day at 0800, 1200 and 1600 hrs. using wooden float and a float run of 50 Mtr. Cross-section taken once a month through sounding of 25 Kg.wt.

(d) Length of river from origin upto site : 134 K.m. (approx.)

(e) Catchment area upto site from origin: 7714 Sq.K.m. (approx.)

(f) Date of commencement of observation of Gauge/Discharge : 1.5.69/ 22.8.74

(g) Maxm.observed Gauge/ Discharge during monsoon of 1979 : 211.80 Mtr.on 24.7.79 (0400 hrs.) / 2906.17 Cumecs on 24.7.79 (0700 hrs.)

(h) Maxm. ever recorded Gauge/Discharge : 217.00 Mtr.on 13.10.73 (0200 hrs.) / 7642.45 Cumecs on 13.10.73 (0600 hrs.)

(i) Type of Raingauge : One S.R. Raingauge and one ordinary raingauge.

6) <u>CORONATION</u>

(a) Location : An important forecasting point on river Teesta on NH-31.

(b) Nature of Station : Gauge and Discharge with 15 Watt. Wireless facilities and non-exchange telephone line between Camp shed and bridge point and camp shed to Sevoke site.

(c) Mode of observation : Hourly gauge observed round the clock during monsoon. Discharge observation thrice a day at 0800, 1200 and 1600 hrs. using wooden float and a float run of 70 Mtr. Cross-section taken once a month through sounding of 25 Kg.wt.

(d) Length of river from origin upto site : 188 K.m. (approx.)

(e) Catchment area upto site from origin : 8147 Sq.K.m. (approx.)

(f) Date of commencement of observation of Gauge/Discharge : 10.8.74/ 1.8.74.

(g) Maxm. observed Gauge/Discharge during monsoon of 1979 : 151.40 Mtr. on 24.7.79 (0600 hrs.) / 2751.36 Cumecs on 24.7.79 (0700 hrs.)

(h) Maxm. ever recorded Gauge/Discharge : 156.500 Mtr. on 12.10.73 (0800 hrs.) / 5090.00 Cumecs on 12.10.73 (0700 hrs.)

(i) Type of Raingauge : NIL.

Table showing the distance of various gauge observation in various basin from the forecasting stations and their travel time.

Sl. No.	Base Station	F/C Station	River distance between base station to F/C station in K.M.	Approx. Travel time
	TEESTA CATCHMENT			
1.	Sankalan Bridge	Coronation Bdg.	84	10 Hrs.
2.	Kantitar	Coronation Bdg.	44	4 Hrs.
3.	Rongpo	Coronation Bdg.	42	4 Hrs.
4.	Singlabazar	Coronation Bdg.	54	4 Hrs.
5.	Mayabazar	Coronation Bdg.	54	4 Hrs.
6.	Teestabazar	Coronation Bdg.	22	2 Hrs.

3.2.3 Results of Investigation

Data from 14.00 hr. on July 23, 1979 to 18.00 hr. on July 26, 1979 were observed at the gauging stations at Sankalan, Great Rangit (Singlabazar), Rongo (Teesta), Teestabazar and Coronation Bridge. The stations are shown in Fig. 3.2.2.

Table 3.2.1 shows the correlation co-efficients of the hourly river flow at Coronation Bridge with flows at the up-stream gauging stations. Input variables are selected on the basis of the strongest co-efficient of correlation with flows at Coronation Bridge.

Table 3.2.2 shows the input-output variables rationalized in accordance with

$$x(k) = \frac{X(k) - X_{min}}{X_{max} - X_{min}} \quad \text{(ii)}$$

where $X(.)$ is the observed value and the values and X_{max} and X_{min} are the maximum and the minimum values of the data sequence. The co-efficient of correlation has been defined as

$$\phi_{yx}(\lambda) = \frac{\sum_{i=1}^{N-\lambda} \left[y(i) - \frac{1}{N}\sum_{i=1}^{N} y(i) \right] \left[x(i+\lambda) - \frac{1}{N}\sum_{i=1}^{N} x(i) \right]}{\left[\sum_{i=1}^{N-\lambda} \left[y(i) - \frac{1}{N}\sum_{i=1}^{N} y(i) \right]^2 \sum_{j=1+\lambda}^{N} \left[x(j) - \frac{1}{N}\sum_{i=1}^{N} x(i) \right]^2 \right]^{\frac{1}{2}}} \quad \ldots (3.2.1)$$

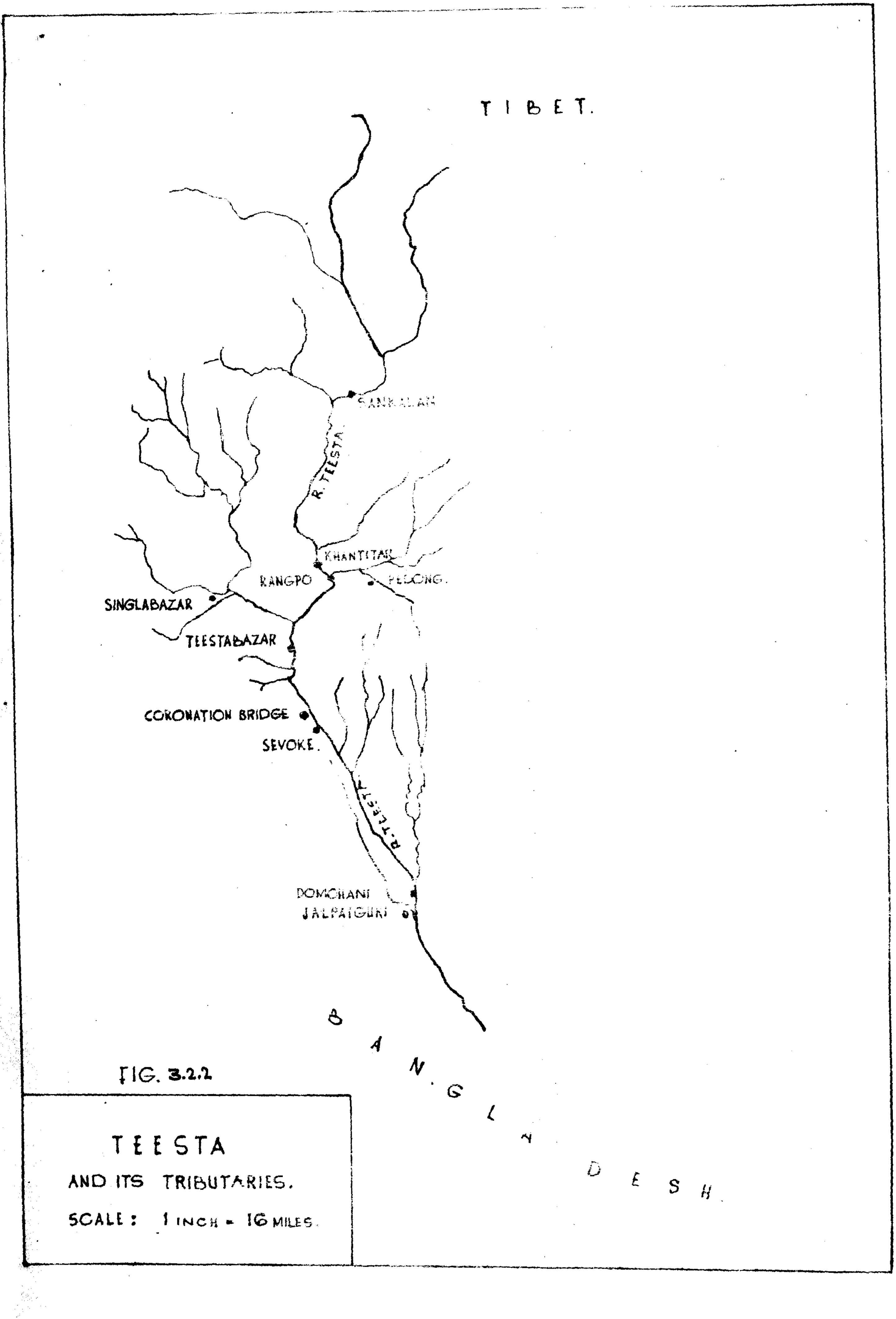


FIG. 3.2.2

TEESTA

AND ITS TRIBUTARIES.

SCALE : 1 INCH = 16 MILES.

TABLE 3.2.1

Correlation Co-efficient against Timeshift Coronation Bridge, Teestabazar, Rango (Teesta), Great Rangeet, Sankalan.

Instant	Coronation Teestabazar	Coronation Rango (Teesta)	Coronation Great Rangeet	Coronation Sankalan
0	0.9421	0.7586	0.7574	0.6534
1	0.9726	0.8306	0.7765	0.7078
2	0.9540	0.8748	0.7908	0.7482
3	0.8886	0.8879	0.7854	0.7828
4	0.7891	0.8628	0.7426	0.7886
5	0.6737	0.8066	0.6649	0.7473
6	0.5516	0.7342	0.5728	0.6690
7	0.4343	0.6493	0.4697	0.5698
8	0.3263	0.5609	0.3584	0.4531
9	0.2300	0.4760	0.2494	0.3301
10	0.1550	0.4015	0.1503	0.2088

TABLE 3.2.2

OUTPUT-INPUT VARIABLES IN RATIONALIZED UNITS

DATE	TIME, hr	CORONATION $Y(t_k)$	TEESTA-BAZAR $U_1(t_{k-1})$	RONGPO (TEESTA) $U_2(t_{k-3})$	GREAT RANGEET $U_3(t_{k-2})$	SANKALAN $U_4(t_{4-4})$
July 23,	14.00	0.318681	0.266666	0.444444	0.086956	0.068181
1979	15.00	0.450549	0.444444	0.644444	0.086956	0.090909
	16.00	0.604395	0.711111	0.800000	0.217391	0.090909
	17.00	0.736263	0.822222	1.000000	0.391304	0.090909
	18.00	0.692307	0.688888	1.000000	0.347826	0.090909
	19.00	0.637362	0.666666	1.000000	0.304347	0.113636
	20.00	0.582417	0.622222	0.977777	0.304347	0.113636
	21.00	0.560439	0.577777	0.888888	0.282608	0.090909
	22.00	0.472527	0.555555	0.800000	0.304347	0.090909
	23.00	0.417582	0.422222	0.622222	0.304347	0.068181
	24.00	0.362637	0.377777	0.533333	0.304347	0.068181

TABLE 3.2.2 (Continued)

DATE	TIME, hr	CORONATION $Y(t_k)$	TEESTA - BAZAR $U_1(t_{k-1})$	RONGPO (TEESTA) $U_2(t_{k-3})$	GREAT RANGEET $U_3(t_{k-2})$	SANKALAN $U_4(t_{4-4})$
July 24, 1979	01.00	0.340659	0.377777	0.488888	0.282608	0.068181
	02.00	0.340659	0.355555	0.466666	0.282608	0.090909
	03.00	0.318681	0.355555	0.444444	0.434782	0.090909
	04.00	0.428571	0.444444	0.400000	0.434782	0.136363
	05.00	0.527472	0.533333	0.444444	0.434782	0.181818
	06.00	0.571428	0.577777	0.577777	0.456521	0.272727
	07.00	0.604395	0.600000	0.622222	0.521739	0.318181
	08.00	0.692307	0.644444	0.644444	0.565217	0.409090
	09.00	0.747252	0.777777	0.644444	0.782608	0.500000
	10.00	0.714285	0.933333	0.666666	1.000000	0.636363
	11.00	0.956043	1.000000	0.800000	1.000000	0.863636
	12.00	1.000000	0.911111	0.800000	0.782608	1.000000
	13.00	0.912087	0.822222	0.844444	0.565217	0.772727
	14.00	0.824175	0.711111	0.800000	0.521739	0.636363
	15.00	0.736263	0.666666	0.711111	0.478260	0.579545
	16.00	0.659340	0.600000	0.622222	0.434782	0.488636
	17.00	0.571428	0.555555	0.622222	0.434782	0.409090
	18.00	0.494505	0.511111	0.577777	0.434782	0.295454
	19.00	0.450549	0.488888	0.502222	0.413043	0.181818
	20.00	0.428571	0.444444	0.422222	0.434782	0.181818
	21.00	0.384615	0.433333	0.400000	0.434782	0.181818
	22.00	0.373626	0.422222	0.400000	0.478260	0.181818
	23.00	0.417582	0.422222	0.355555	0.478260	0.261363
	24.00	0.439560	0.466666	0.311111	0.456521	0.272727

TABLE 3.2.2 (Continued)

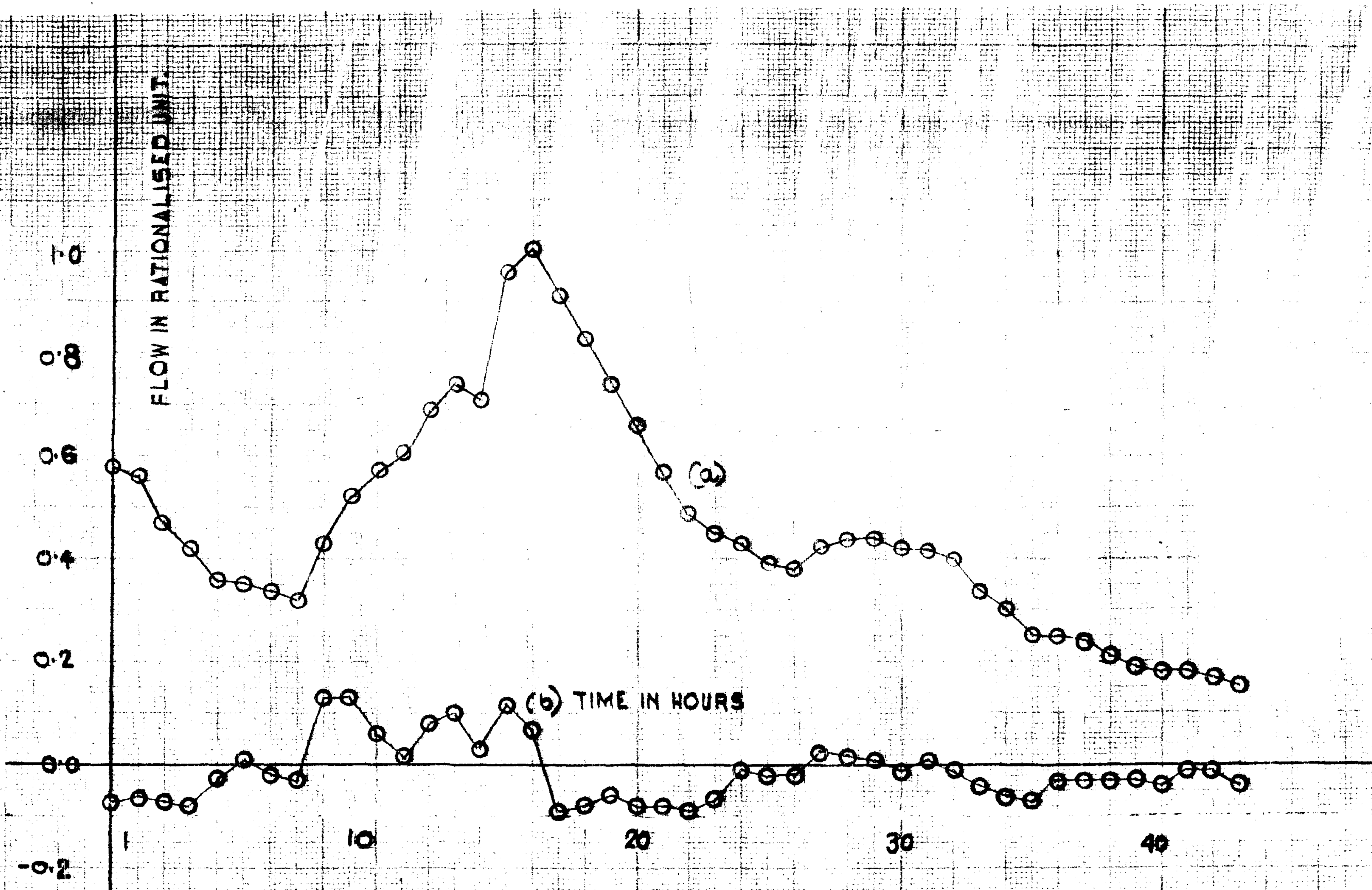
DATE	TIME, hr	CORONATION $Y(t_k)$	TEESTA - BAZAR $U_1(t_{k-1})$	RONGPO (TEESTA) $U_2(t_{k-3})$	GREAT RANGEET $U_3(t_{k-2})$	SANKALAN $U_4(t_{4-4})$
July 25, 1979	01.00	0.439560	0.466666	0.266666	0.391304	0.272727
	02.00	0.428571	0.466666	0.266666	0.391304	0.272727
	03.00	0.423076	0.444444	0.222222	0.347826	0.238636
	04.00	0.395604	0.422222	0.222222	0.364565	0.238636
	05.00	0.340659	0.400000	0.222222	0.413043	0.238636
	06.00	0.296703	0.377777	0.222222	0.434782	0.238636
	07.00	0.252747	0.377777	0.222222	0.434782	0.181818
	08.00	0.252747	0.377777	0.200000	0.434782	0.181818
	09.00	0.241758	0.333333	0.200000	0.423043	0.181818
	10.00	0.219780	0.288888	0.177777	0.369565	0.181818
	11.00	0.197802	0.266666	0.177777	0.326086	0.181818
	12.00	0.186813	0.244444	0.177777	0.304347	0.181818
	13.00	0.186813	0.244444	0.155555	0.282608	0.136363
	14.00	0.164835	0.222222	0.133333	0.260869	0.136363
	15.00	0.142857	0.222222	0.133333	0.260869	0.102272

Using recursive instrument variable algorithm with parameter tracking adaptiveness for estimation of instrument variables the mean error and the integral square error for $n = 2$ and $Q = 2$ were found as 0.013475 and 0.014917 respectively. The integral square error has been defined as,

$$ISE = \frac{\sum_{j=1}^{N} \left[Y(t_j) - \hat{Y}(t_j) \right]^2}{\sum_{j=1}^{N} \left[Y(t_j) \right]^2} \qquad (3.2.2)$$

For a moving average sequence of $P = 2$, $\eta(t_k)$ was found to have a mean error of 9.250282 E-03 and a variance of the error sequence $v(k)$ as 0.934175. Observed and the errors between the observed and modelled values of the hourly river flows are shown in Fig. 3.2.3a and 3.2.3b respectively. Parameter vector '$\hat{\alpha}$' and the matrix P^* have been initialised for block of data using least square technique. Table 3.2.3 and Table 3.2.4 show the initial values of the parameter vector '$\hat{\alpha}$' and the P^* matrix respectively.

Thus, the recursive instrumental variable algorithms have been found to simulate adequately the observed hourly flows for real time operation [33]. The computer programmes in BASIC language are given in the Appendix as A1.1 - A1.5.



FLOW OF RIVER TEESTA AT CORONATION BRIDGE FROM 20'00 HOUR ON JULY 23, 1979 - 15'00 HOUR ON JULY 25, 1979

FIG 3.2.3a & 3.2.3b

TABLE 3.2.3

INITIAL VALUES OF THE PARAMETER VECTOR " $\hat{\underset{\sim}{\alpha}}$ "

Element	Value
1	0.399774
2	0.014683
3	0.488142
4	-0.088370
5	-0.247403
6	0.213479
7	0.082624
8	-0.128437
9	-0.209974
10	0.241120
11	0.168889
12	0.220721
13	-0.388690
14	0.166041

TABLE 3.2.4

INITIAL VALUES OF P*(. , .) MATRIX

Column / Row	1	2	3	4	5
1	58.880664	-8.889063	-40.030619	-33.521285	-38.186045
2	-8.889063	42.177748	5.355676	13.017608	-22.537688
3	-40.030619	5.355676	73.677836	22.761898	54.834963
4	-33.521285	13.017608	22.761898	71.448512	7.330765
5	-38.186045	-22.537688	54.834963	7.330765	94.575968
6	37.562788	-13.953667	-68.268940	-45.479030	-39.470863
7	-16.391836	9.196837	15.698047	0.450839	-2.882897
8	15.003694	-11.202067	-14.154664	-2.339096	-12.157318
9	0.987203	-10.415648	-17.542502	-2.414112	1.880186
10	39.029081	0.235924	-37.433986	-46.712746	-40.959851
11	-14.271545	6.368086	9.629415	19.934608	-3.078808
12	6.120561	1.193557	-7.605047	-2.480554	-2.551219
13	-6.585247	4.076049	-2.720300	-3.768386	0.046091
14	6.494609	-12.928262	-8.330788	-8.911984	1.032974

TABLE 3.2.4 (Continued)

Column / Row	6	7	8	9	10
1	37.562788	-16.391936	15.003694	0.987203	39.029081
2	-13.953667	9.196837	-11.202057	-10.415648	0.236924
3	-62.262940	15.692047	-14.154664	-17.543808	-37.432986
4	-45.479030	0.450839	- 2.338026	- 2.414112	-46.712746
5	-39.470863	- 2.582897	-12.157318	1.880156	-40.969551
6	86.771294	-30.972607	18.657976	15.318287	47.790582
7	-30.972607	47.886238	-22.121170	- 1.996982	- 7.449749
8	18.657976	-22.121170	19.164287	1.333767	8.713446
9	15.318287	- 1.996982	1.333767	16.445804	- 7.199783
10	47.790582	- 7.449749	8.713446	- 7.199783	65.210142
11	-16.687083	10.421918	- 5.727901	0.572292	-22.645643
12	6.034589	- 1.466709	3.596923	- 1.512034	- 1.167621
13	2.840898	1.696162	- 0.275096	5.385988	3.787456
14	14.064357	- 3.282177	4.766636	3.231064	7.722280

TABLE 3.2.4 (Continued)

Column / Row	11	12	13	14
1	-14.271545	6.120561	-6.585247	6.494409
2	6.358026	1.193587	4.076049	-12.925282
3	9.629415	-7.605047	-2.720300	-8.320788
4	12.934606	-2.420654	-3.763396	-8.911984
5	-3.078808	-8.551819	0.046091	1.032974
6	-16.687083	6.034539	3.840898	14.064257
7	10.421918	-1.466709	1.696162	-3.282177
8	-5.727901	3.596983	-0.275096	4.766638
9	0.872293	-1.518034	5.385928	3.231054
10	-22.645643	-1.167621	3.787456	7.782280
11	20.258990	1.900406	-8.207861	-0.987570
12	1.900406	18.380408	-17.247785	4.283845
13	-8.207861	-17.247785	37.880206	-16.887294
14	-0.987570	4.283845	-16.887294	20.010860

CHAPTER IV

ELECTRICAL ENERGY CONSUMPTION MODEL WITH INTERACTING PARAMETERS BY A LEARNING IDENTIFICATION ALGORITHM

CHAPTER IV

ELECTRICAL ENERGY CONSUMPTION MODEL WITH INTERACTING PARAMETERS BY A LEARNING IDENTIFICATION ALGORITHM

4.0 Introduction

The levels of energy consumption reveal the levels of cultural and economic development of different countries. Techno-economic and socio-economic parameters of energy system are interrelated. The energy system has distinct cybernetic features. Venikov [93] has pointed out that deep-lying feedback paths exist in energy system. Thus it is understandable that significant interaction exists between energy consumption and different techno-economic parameters.

This paper presents a mathematical description of annual energy consumption with population, gross national product, gross domestic saving and gross domestic capital formation as exogenous variables in the form of a polynomial of optimum complexity with the help of a method of applied cybernetics commonly known as multilayer group method of data handling algorithm (GMDH). The method has the potentiality of identifying, implicitly in a learning identification environment, the interactions and feedbacks of interactions of different input parameters on the output of the process.

To give a mathematical description of the annual energy consumption as a function of a set of exogenous variables interrelated with one another through deep-lying feedback paths is a complex process. Modern Control theory based on differential or difference equations is not adequate to describle the process. In view of this difficulty, the method of modelling applied here uses a technique of self-organisation. This GMDH algorithm of self-organisation involves generation and comparison of different regression polynomials by using all possible combinations of input variables and selection therefrom of the best possible ones according to the criterion of minimum integral square error defined in equation (4.1.11). The GMDH technique is found to simulate adequately the input-output relationship of the complex process of annual electrical energy consumption as a function of a set of input varieables (input set is not exhaustive).

4.1.0 Brief Description of GMDH

The multilayer group method of data handling involves the use of regression polynomials as the basic means of investigation of complex dynamical systems. The relevant polynomial is a regression equation which connects a value of an output variable with past or current values of output and input variables. The regression analysis in this case helps in evaluating the co-efficients of the polynomial by using the criterion of minimum integral square error. The polynomials are then treated in the same manner as that used for selection of seeds in agriculture as per a unique mathematical concept propagated and established by Ivakhnenko [36,37].

The salient features of GMDH as applicable in the case of multilayer selection process used in the present work are now briefly described here :

The process can be described by

$$\theta = f(x_1, x_2, \ldots, x_n) \tag{4.1.1}$$

and it involves the construction of several layers of partial descriptions using two input variables at a time, e.g., the first layer can be represented as

$$y_i = f(x_j, x_k) \tag{4.1.2}$$

for $j = 1, 2, \ldots, n$

with $k = 1, 2, \ldots, n \; (j \neq k)$

and $i = 1, 2, \ldots, m$ where $m = \binom{n}{2}$

Likewise the second layer can be represented as

$$z_{i'} = g(y_{j'}, y_{k'}) \tag{4.1.3}$$

for $j' = 1, 2, \ldots, m$

with $k' = 1, 2, \ldots, m \; (j' \neq k')$

and $i' = 1, 2, \ldots, p$ where $p = \binom{m}{2}$

and so on, it being noted that m and p are the numbers of pairwise combinations of the first and second layers respectively. The first step concerns the selection of input variables on the basis of strong correlation (defined in equation (4.1.7).

The co-efficients of the first layer of partial description are calculated by solving a system of normal Gaussian equations. The left hand side of the equations are set equal to the values of output at every point. After finding the values of the co-efficients, the values of the intermediate variebles are obtained. Then using the observed data the integral square error is determined for each of the variables. Only those variables which give low error (self selection threshold) are selected for subsequent use. These variables are retained and other variables are discarded. In the second layer of selection, the co-efficient of the partial description of the layer are calculated and the accuracy is checked again to select the accurate intermediate variables of the layer ; $z_1, z_2, \ldots z_p$. The process of selection continues so long as the integral square error on the whole data set comes to a minimum and then starts increasing in the next layer or, the integral square error approaches an asymptotic minimum.

The polynomial description of the process is obtained in the form of partial description of the intermediate variables of different layers. Eliminating the intermediate variables, the complete polynomial description of the process is obtained in the form of Gabor-Kolmogrov type polynomial as

$$\Theta = a_0 + \sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j x_i x_j$$

$$+ \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{n} a_i a_j a_k x_i x_j x_k + \ldots \qquad (4.1.4)$$

Use of GMDH in formulating the electrical energy consumption process :

4.1.1 Input Data :

The input data from the Indian techno-economic scenario consisted of the annual energy generated (gross) in Mn.Kwh, population in million, gross national product at factor cost (Rs. Crores) at 1970-71 prices, gross domestic saving and gross domestic capital formation as percent of gross domestic product at market prices from 1960-61 to 1980-81. The complete data are shown in Table 4.1.1 . All the relevant data X(k) for the k-th year were rationalised as

$$x(k) = \frac{X(k) - X_{min}}{X_{max} - X_{min}} \qquad (4.1.5)$$

TABLE 4.1.1 INPUT DATA

YEAR	ENERGY GENERATED (GROSS) IN Mn. Kwh	POPULATION (MILLION)	GROSS NATIONAL PRODUCT AT FACTOR COST AT 1970-71 PRICES IN Rs. CRORES	GROSS DOMESTIC SAVING AS PER CENT OF GROSS DOMESTIC PRODUCT AT MARKET PRICES	GROSS DOMESTIC CAPITAL FORMATION AS PER CENT OF GROSS DOMESTIC PRODUCT AT MARKET PRICES
1960-61	20123	442.4	25424	13.7	16.9
1961-62	22957	452.2	26293	13.1	16.3
1962-63	26237	462.0	26834	14.5	17.1
1963-64	30212	472.1	28210	14.4	16.6
1964-65	33133	482.5	30399	13.6	16.8
1965-66	36826	493.2	28791	15.7	18.2
1966-67	40612	504.2	29081	16.3	19.7
1967-68	45215	515.4	31590	13.8	16.5
1968-69	51642	527.0	32460	14.1	15.4
1969-70	56544	538.9	34518	16.4	17.1
1970-71	61211	551.2	36452	16.8	17.8
1971-72	66384	562.5	37000	17.3	18.4
1972-73	70516	575.9	36599	16.8	16.9
1973-74	72796	588.3	38410	19.3	20.0
1974-75	76678	600.8	38857	18.2	19.1
1975-76	85926	613.2	42571	20.0	19.9
1976-77	95615	625.8	43124	22.0	20.4
1977-78	98928	638.4	46854	21.8	19.7
1978-79	110130	651.0	49343	24.4	24.6
1979-80	112820	663.6	46966	22.5	22.9
1980-81	119227	682.8	50507	22.2	24.2

4.1.2 Formulation of the process equation

The annual electrical energy consumption can be represented by a general form of process equation as

$$y(k) = f\,[\,y(k-1),\ y(k-2),\ldots,x_1(k),x_1(k-1),\ldots,$$
$$x_2(k),x_2(k-1),\ldots,x_3(k),x_3(k-1),\ldots\,] \qquad (4.1.6)$$

where k, k-1, k-2,..., refers respectively to the current day, one day preceding the current day, two days preceding the current day and so on and $y(\cdot)$, $x_1(\cdot)$, $x_2(\cdot)$, $x_3(\cdot)$ and $x_4(\cdot)$ are the rationalised data for annual electrical energy consumption, population, gross national product, gross domestic saving and gross domestic capital formation respectively.

The arguments having correlations with $y(k)$ are then selected for inclusion in the process equation of the basis of the correlation functions for the time shift λ, in years, defined as

$$\psi_{yx}(\lambda) = \frac{\displaystyle\sum_{i=1}^{N-\lambda}\left[\left(y(i)-\frac{1}{N}\sum_{k=1}^{N} y(k)\right)\left(x(i+\lambda)-\frac{1}{N}\sum_{k=1}^{N} x(k)\right)\right]}{\left[\displaystyle\sum_{i=1}^{N-\lambda}\left(y(i)-\frac{1}{N}\sum_{k=1}^{N} y(k)\right)^2 \sum_{j=1+\lambda}^{N}\left(x(j)-\frac{1}{N}\sum_{k=1}^{N} x(k)\right)^2\right]^{\frac{1}{2}}} \qquad \ldots (4.1.7)$$

where N is the number of data points.

After such selection of arguments as having correlation with annual energy consumption the process equation becomes

$$y(k) = f\left[y(k-1), x_1(k), x_1(k-1), x_1(k-2), x_2(k), x_2(k-1), x_3(k) \right] \quad \ldots \; (4.1.8)$$

or denoting these respective arguments as $y(k) = y, y(k-1) = x_1'$, $x_1(k) = x_2'$, $x_1(k-1) = x_3'$, $x_1(k-2) = x_4'$, $x_2(k) = x_5'$, $x_2(k-1) = x_6'$ and $x_3(k) = x_7'$, the process equation becomes,

$$y = f\ (x_1', x_2', x_3', x_4', x_5', x_6', x_7') \qquad (4.1.9)$$

Table 4.1.2 shows the correlation co-efficients of different exogenous variables with the annual electrical energy consumption for different lagged instances in years.

4.1.3 First layer of selection

There are $\binom{7}{2} = 21$ possible combinations of selecting two arguments at a time out of seven. For every such combination, the partial regression equation is written as

$$y_a = \alpha_{0a} + \alpha_{1a} x_b' + \alpha_{2a} x_c' + \alpha_{3a} x_b' x_c' + \alpha_{4a} x_b'^2 + \alpha_{5a} x_c'^2 \qquad (4.1.10)$$

where $a = 1, 2, \ldots, 21$, while b and c are indices for all 21 combinations. And this lead to 21 systems of normal

TABLE 4.1.2

CORRELATION COEFFICIENTS OF DIFFERENT EXOGENOUS VARIABLES WITH THE ANNUAL ELECTRICAL ENERGY CONSUMPTION FOR DIFFERENT LAGGED INSTANCES IN YEARS.

Time instant (Year)	Annual Energy Consumption - Annual Energy Consumption	Annual Energy Consumption - Population	Annual Energy Consumption - Gross National Product	Annual Energy Consumption - Gross Domestic Saving	Annual Energy Consumption - Gross Domestic Capital Formation
0	1	0.995851	0.992315	0.947459	0.850142
1	0.982801	0.979509	0.973108	0.926665	0.842097
2	0.932136	0.926809	0.925295	0.876826	0.805719
3	0.834372	0.823633	0.822906	0.801000	0.742843
4	0.699787	0.680482	0.694260	0.688842	0.640432
5	0.516921	0.496483	0.542507	0.527507	0.498223
6	0.319394	0.277528	0.347854	0.400121	0.411760
7	0.130834	0.082781	0.144170	0.288482	0.411783
8	-0.059705	-0.108712	-0.042876	0.072807	0.260831
9	-0.244062	-0.298346	-0.237582	-0.174616	0.086300
10	-0.411211	-0.462869	-0.402052	-0.338780	-0.150841

Gaussian equations with matrices of the order 6 x 6. The co-efficients α's are then estimated by solving normal equation systems constructed from the data set. For estimating the co-efficients it is assumed that the equation error is very small, being distributed with zero mean, constant variance and also uncorrelated with the inputs. The second assumption is that for the construction of the model the inputs and outputs are known exactly without any measurement error.

The accuracy of every variable y_a is calculated by using the entire data set. From all variables seven more accurate ones are chosen which give low values of integral square error criterion as

$$ISE = \frac{\sum_{i=1}^{N} \left[y_{observed}(i) - y_{modelled}(i) \right]^2}{\sum_{i=1}^{N} \left[y_{observed}(i) \right]^2} \qquad (4.1.11)$$

4.1.4 Selection of other layers

Seven intermediate variables of y_a layer chosen from the first layer give 21 combinations of two arguments of y_a layer. Again in the second layer these becomes

$$z_a = \beta_{oa} + \beta_{1a}y_b' + \beta_{2a}y_c' + \beta_{3a}y_b' y_c' + \beta_{4a}y_b'^2 + \beta_{5a}y_c'^2 \qquad \ldots \quad (4.1.12)$$

where a = 1,2,..., 21 while b and c are indices of all 21 combinations. Calculation of the co-efficients of β and estimation of the accuracy of z_a are repeated as in the case of y_a.

The seven z_a variables are then chosen for the next layer u_a.

$$u_a = \gamma_{oa} + \gamma_{1a}z_b' + \gamma_{2a}z_c' + \gamma_{3a}z_b' z_c' + \gamma_{4a}z_b'^2 + \gamma_{5a}z_c'^2 \qquad (4.1.13)$$

In this way each layer is tested for accuracy by using the entire data set and on the basis of minimum integral square error criterion explained earlier. For all layers, variables on the left hand side of the equations are kept equal to the value of the output variable.

4.2.0 Illustration

It was observed that as the layer increases the integral square errors converge asymptotically to a very small value. To make the model suitable for practical application the integral square error of 3.408346E-04 at z_5 was considered for the point of termination of formation of process equation.

The changes of integral square error for different layers are shown in Fig.4.2.1. The integral square errors for different combinations in different layers are shown in Table 4.2.1 in ascending order.

The annual energy consumption in India has been identified by the polynomial as

$$y = z_5$$

$$z_5 = 7.490082E-0.3 + 1.428809 y_1' - 0.496869 y_6'$$
$$-35.769734 y_1' y_6' + 17.285701 y_1'^2 + 12.524865 y_6'^2$$

$$y_1' = -0.031427 + 1.470818 x_2' - 0.602856 x_6' + 4.858670 x_2' x_6'$$
$$-3.220067 x_2'^2 - 2.814044 x_6'^2$$

$$y_6' = 0.074196 + 0.934731 x_4' - 0.131836 x_7'$$
$$+ 0.063213 x_4' x_7' + 0.088502 x_4'^2 + 0.156838 x_7'^2 \quad (4.2.1)$$

Fig.4.2.2(a) and 4.2.2(b) shows the observed values and errors between the observed and the modelled values of annual electrical energy consumption in India in rationalised unit.

Modelling errors are found to have a variance of 1.000014 and mean of 2.7801742E-04, and are almost found to be uncorrelated for $i \neq j$. [89] The software developed in BASIC language is given in the Appendix as A2.1.

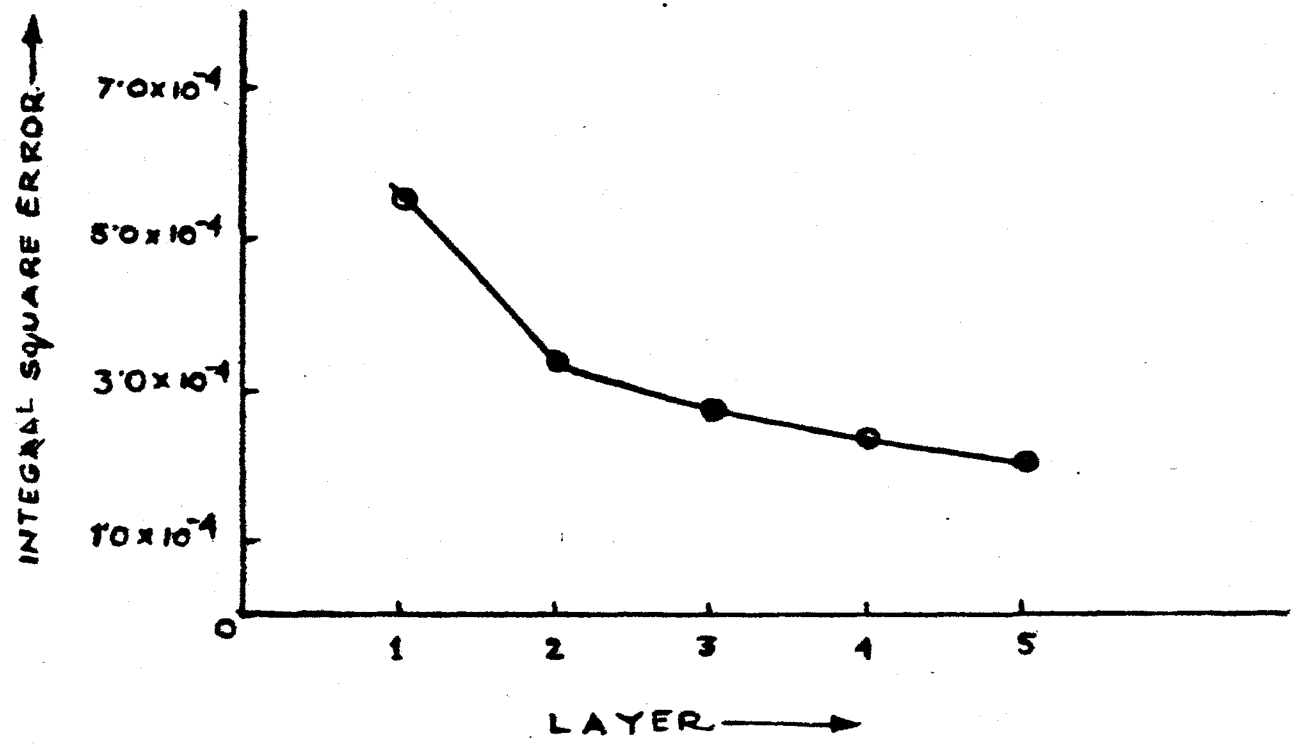


FIG. 4.2.1 INTEGRAL SQUARE ERROR IN DIFFERENT LAYERS.

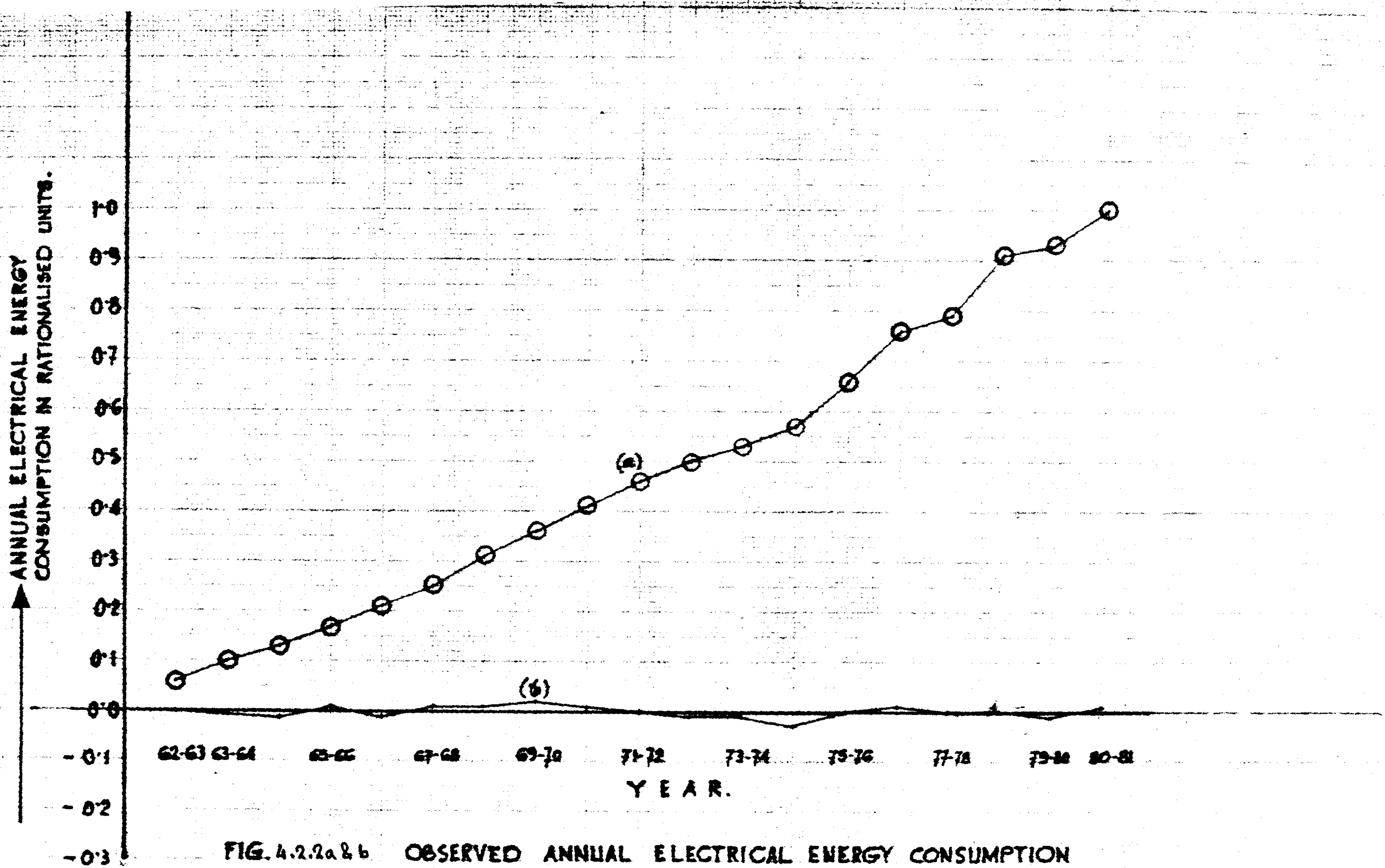


FIG. 4.2.2a & b OBSERVED ANNUAL ELECTRICAL ENERGY CONSUMPTION IN RATIONALISED UNITS & ERRORS BETWEEN THE OBSERVED & THE MODELLED VALUES.

TABLE 4.2.1

THE INTEGRAL SQUARE ERRORS FOR DIFFERENT COMBINATIONS IN DIFFERENT LAYERS.

Layer 1		Layer 2	
Combination	Integral Sq. Error	Combination	Integral Square Error
$x_2'x_6'$	5.593734 E-04	$y_1'y_6'$	3.408346 E-04
$x_3'x_6'$	5.714853 E-04	$y_1'y_5'$	3.427570 E-04
$x_4'x_6'$	5.796394 E-04	$y_1'y_4'$	3.488152 E-04
$x_2'x_7'$	6.330834 E-04	$y_2'y_4'$	3.827865 E-04
$x_3'x_7'$	7.097150 E-04	$y_3'y_6'$	3.898408 E-04
$x_4'x_7'$	7.248394 E-04	$y_3'y_4'$	3.903861 E-04
$x_3'x_5'$	7.431770 E-04	$y_2'y_5'$	3.916334 E-04

TABLE 4.2.1 (Continued)

Layer 3		Layer 4	
Combination	Integral Sq. Error	Combination	Integral Square Error
z_4z_7	2.808629 E-04	U_2U_4	2.432557 E-04
z_4z_5	2.986592 E-04	U_2U_3	2.454077 E-04
z_2z_5	3.053902 E-04	U_2U_5	2.482261 E-04
z_2z_6	3.058650 E-04	U_2U_6	2.482578 E-04
z_3z_5	3.076447 E-04	U_2U_8	2.537689 E-04
z_3z_6	3.082091 E-04	U_2U_7	2.569498 E-04
z_2z_7	3.198816 E-04	U_5U_7	2.738455 E-04

CHAPTER V

MEDIUM TERM AND LONG TERM PREDICTION MODELS OF ANNUAL INSTALLED PLANT CAPACITY AND CONSUMPTION OF ELECTRICAL ENERGY BY COMPUTER-AIDED SELF-ORGANISATION OF MATHEMATICAL MODELS

CHAPTER V

MEDIUM TERM AND LONG TERM PREDICTION MODELS OF ANNUAL INSTALLED PLANT CAPACITY AND CONSUMPTION OF ELECTRICAL ENERGY BY COMPUTER-AIDED SELF-ORGANISATION OF MATHEMATICAL MODELS

5.0 Introduction

This work relating to the short term and long term forecasting models of annual installed plant capacity and consumption of electrical energy of India has been divided into four parts. In the first part an annual model of installed capacity of electrical energy for medium term (6 to 7 years) prediction has been obtained. Different types of polynomials of increasing complexity have been tested. The polynomial which gives minimum of a selection criterion has been found. In the second part assuming an annual growth rate of installed capacity of 8 % a long term prediction model of installed plant capacity of electrical energy has been obtained.

In the third part assuming an annual growth rate of electrical energy consumption of 8 % a long term prediction model of annual energy consumption has been obtained. The annual plant load factor has been found to have a distinct preiodicity.

In the fourth part a polynomial model of annual load factor has been obtained incorporating periodic terms.

With the theory of self-organisation commonly known as group method of data handling it has been possible to formulate mathematical models for complex processes with prediction optimisation.

The concept of self-organisation can be illustrated as follows :

When the model complexity gradually increases the computer finds by shifting the different models, the minimum of a selection criterion which the computer has been asked to look for. Thus the computer indicates to the operator the model of optimum complexity.

5.1 Medium-term prediction model of annual installed plant capacity of electrical energy

We have annual installed plant capacity data from 1961 - 1981. Power density spectra versus cycle per annum characteristic [31] of the data does not show any harmonicity in the process. So it is obvious that the process does not contain any sinusoidal harmonic parts. The correlation co-efficients versus shift of instances of time (equation 5.2.2) show that the current year installed plant capacity is strongly correlated with the past three years installed plant capacity.

Thus the process is assumed to have a finite difference form structure of the following nature.

$$Y_{k+1} = f\ (Y_k, Y_{k-1}, Y_{k-2}, t_k) \qquad (5.1.1)$$

We write,

Y_{k+1} = y, installed plant capacity of electrical energy for the k+1 - th year.

Y_k = x_1, installed plant capacity of electrical energy for the k - th year.

Y_{k-1} = x_2, installed plant capacity of electrical energy for the k-1 - th year.

Y_{k-2} = x_3, installed plant capacity of electrical energy for the k-2 - th year.

$t_k = x_4$, time instant for the k-th year

$$\text{so, } y = f\ (x_1, x_2, x_3, x_4) \qquad (5.1.2)$$

The function $f(.)$ is sought in the class of quadratic polynomials on the basis of a Table of polynomial of gradually increasing complexity of four variebles as shown in Table 5.1.1, with the help of the theory of the self-organisation of combinatorial group method of data handling algorithm. The model of optimum complexity is selected on the basis of the minimum of the integral square error criterion.

TABLE 5.1.1

GRADUALLY INCREASING COMPLEXITY OF POLYNOMIALS FOR FOUR VARIABLES

General form of polynomial :

$$y = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_3 + a_5x_1x_3 + a_6x_2x_3 + a_7x_4 + a_8x_1x_4 + a_9x_2x_4 + a_{10}x_3x_4$$

x_1	x_2	x_1x_2	x_3	x_1x_3	x_3x_4
$y_1 = a_0 + a_1x_1$	$y_2 = a_0 + a_2x_2$	$y_4 = a_0 + a_3x_1x_2$	$y_8 = a_0 + a_4x_3$	$y_{16} = a_0 + a_5x_1x_3$	$y_{512} = a_0 + a_{10}x_3x_4$
	$y_3 = a_0 + a_1x_1 + a_2x_2$	$y_5 = a_0 + a_1x_1 + a_3x_1x_2$	$y_9 = a_0 + a_1x_1 + a_4x_3$	$y_{17} = a_0 + \cdots$	$y_{513} = a_0 + a_1x_1 + a_{10}x_3x_4$
		$y_6 = a_0 + a_2x_2 + a_3x_1x_2$	...	...	...
		$y_7 = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2$	...	...	...
			$y_{15} = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_3$	...	$y_{1023} = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_3 + a_5x_1x_3 + \cdots + a_{10}x_3x_4$ *

* General rule of total number of combination is $2^n - 1$.
Where n is the number of terms containing the variables in the general form of polynomial.

The integral square error is defined as

$$\text{I.S.E.} = \frac{\sum_{i=1}^{N} \left(X_{tab}(i) - X_{d.m}(i) \right)^2}{\sum_{i=1}^{N} \left(X_{tab}(i) \right)^2} \qquad (5.1.3)$$

where $X_{tab}(i)$, $i = 1, 2, \ldots, N$ years are the tabulated values of the variables in the interpolation region and $X_{d.m}(i)$ are the values of the variable obtained from the model. The time instances t_k is taken as $1, 2, \ldots,$ $k = 1961, 1962, \ldots$ and so on. The models from the Table 5.1.1 comprising of four variables are tested for all the data points.

The model of annual installed plant capacity of electrical energy in MW for India is obtained as

$$\begin{aligned} y = \; & 5584.157096 + 0.599659\, x_1 \\ & -0.330972\, x_2 - 0.185849\, x_3 \\ & +770.911284\, x_4 + 1.05645165\text{ E-05 } x_1 x_2 \\ & -8.57186948\text{ E-09 } x_1 x_3 \end{aligned} \qquad (5.1.4)$$

In finite difference form,

$$\begin{aligned} Y_{k+1} = \; & 5584.157096 + 0.59965\, Y_k \\ & - 0.330972\, Y_{k-1} - 0.185849\, Y_{k-2} \\ & + 770.911284\, t_k + 1.05645165\text{ E- 05 } Y_k Y_{k-1} \\ & - 8.57186948\text{ E - 09 } Y_k Y_{k-2} \end{aligned} \qquad (5.1.5)$$

The corresponding minimum value for the integral square error and the mean error are 9.366641 E - 04 and 8.6946 E - 06 respectively.

The errors between the observed and the modelled values are found to be almost uncorrelated for k ≠ j. Fig. 5.1.1a and 5.1.1b shows the observed and errors between the observed and the modelled values respectively. Fig. 5.1.1a has been extrapolated for seven years of prediction of the installed plant capacity of electrical energy i.e., upto 1988.

5.2 Long term prediction model of installed plant capacity

During the period from 1970 - 80 the installed plant capacity of electrical energy has grown at an average annual rate of 7.2 %. Considering the deleterious impact of power shortage on the productive sectors, both industry and agriculture, of the economy the Planning Commission of the Govt. of India has suggested an average annual growth rate of 11.3 per cent during the Sixth Five Year Plan period (1980 - 85). The growth rate has been suggested on the assumption that a distinct improvement in the working of power plants and strict adherence to the working schedules of power projects. Over the years the trend has been a

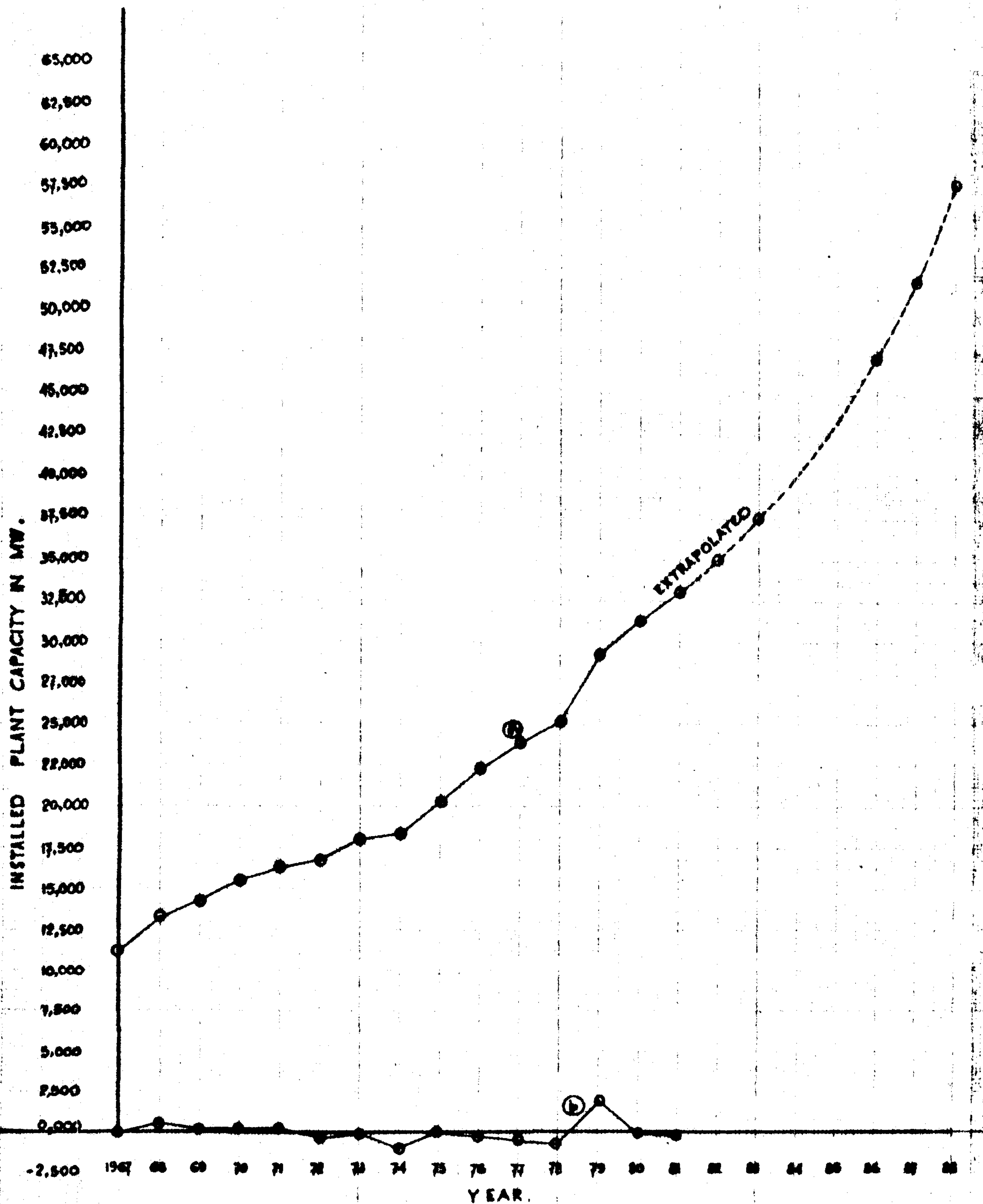


FIG 5.1.1a & b ANNUAL INSTALLED PLANT CAPACITY EXTRAPOLATED FROM 1982-88 IN MW AND ERRORS BETWEEN THE OBSERVED AND MODELLED VALUES RESPECTIVELY.

mounting negligence and an appalling backlags. The plant load factor has been moving in the vicinity of 45 per cent. It is now suggested that an annual growth rate of installed plant capacity of 8 per cent would be a realistic estimate. Because of the interactions of different deep lying feedback paths it may be recognised that this growth rate is a complicated process and is likely to change slowly rather than quickly.

On the basis of an annual growth rate of 8 per cent the installed plant capacity of electrical energy has been extrapolated upto 4 times the 1981 figures.

A mathematical model in the form of a finite difference equation has been postulated as

$$Y_{k+1} = f\,(Y_k, Y_{k-1}, Y_{k-2}, Y_{k-3}, t_k) \qquad (5.2.1)$$

The four arguments Y_k, Y_{k-1}, Y_{k-2}, Y_{k-3} are selected because of their strong correlation on Y_{k+1}. The correlation co-efficient at a shift instance λ is defined as

$$\phi_{YY}(\lambda) = \frac{\sum_{i=1}^{N-\lambda} \left[(Y_i - \frac{1}{N} \sum_{k=1}^{N} Y_k)(Y_{i+\lambda} - \frac{1}{N} \sum_{k=1}^{N} Y_k) \right]}{\left[\sum_{i=1}^{N-\lambda} (Y_i - \frac{1}{N} \sum_{k=1}^{N} Y_k)^2 \sum_{j=1+\lambda}^{N} (Y_j - \frac{1}{N} \sum_{k=1}^{N} Y_k)^2 \right]^{\frac{1}{2}}}$$

... (5.2.2)

Polynomials of gradually increasing complexity of five variables are shown in Table 5.2.1.

The model of optimum complexity which gives minimum of integral square criterion is found to be

$$
\begin{aligned}
Y_{k+1} = & - 232.764083 + 0.992565\ Y_k \\
& + 0.167739\ Y_{k-1} - 0.020582\ Y_{k-2} \\
& + 0.086204\ Y_{k-3} - 171.471615\ t_k \\
& - 6.719787\ E-07\ Y_k\ Y_{k-1} \\
& + 3.512649\ E-11\ Y_k\ Y_{k-2}
\end{aligned}
\qquad (5.2.3)
$$

The corresponding values of the integral square error and the mean error are 1.284107 E-04 and 1.812884 E-05.

The errors are found to be almost uncorrelated for $k \neq j$.

Fig. 5.2.1a and 5.2.1b shows the extrapolated annual installed plant capacity upto 1998 and errors of modelling respectively.

5.3 Long term prediction model of annual electrical energy consumption

On the basis of an annual growth rate of 8 per cent the electrical energy consumption has been extrapolated to 5 times its 1981 consumption figures. It has been observed

TABLE 5.8.1

GRADUALLY INCREASING COMPLEXITY OF POLYNOMIALS OF FIVE VARIABLES

General form of polynomial :

$y = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_3 + a_5x_1x_3 + a_6x_2x_3 + a_7x_4 + a_8x_1x_4 + a_9x_2x_4 + a_{10}x_3x_4 + a_{11}x_5 + a_{12}x_1x_5 + a_{13}x_2x_5 + a_{14}x_3x_5 + a_{15}x_4x_5$

x_1	x_2	x_1x_2	x_3	x_1x_3	...	x_3x_4	...	x_4x_5
$y_1 = a_0 + a_1x_1$	$y_2 = a_0 + a_2x_2$	$y_4 = a_0 + a_3x_1x_2$	$y_8 = a_0 + a_4x_3$	$y_{16} = a_0 + a_5x_1x_3$	...	$y_{512} = a_0 + a_{10}x_3x_4$	...	....
	$y_3 = a_0 + a_1x_1 + a_2x_2$	$y_5 = a_0 + a_2x_2 + a_3x_1x_2$	$y_9 = a_0 + a_1x_1 + a_4x_3$	$y_{17} = a_0 +$		$y_{513} = a_0 + a_1x_1 + a_{10}x_3x_4$		⋮
		$y_6 = a_0 + a_1x_1 + a_3x_1x_2$	⋮			⋮		
		$y_7 = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2$	$y_{15} = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_3$			$y_{1023} = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_3 + a_5x_1x_3 + \ldots + a_{10}x_3x_4$		$y_{32767} = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + \ldots + a_{15}x_4x_5$

The General rule of total number of combination is $2^n - 1$ where n is the total number of terms containing the variables in the general form of the polynomial.

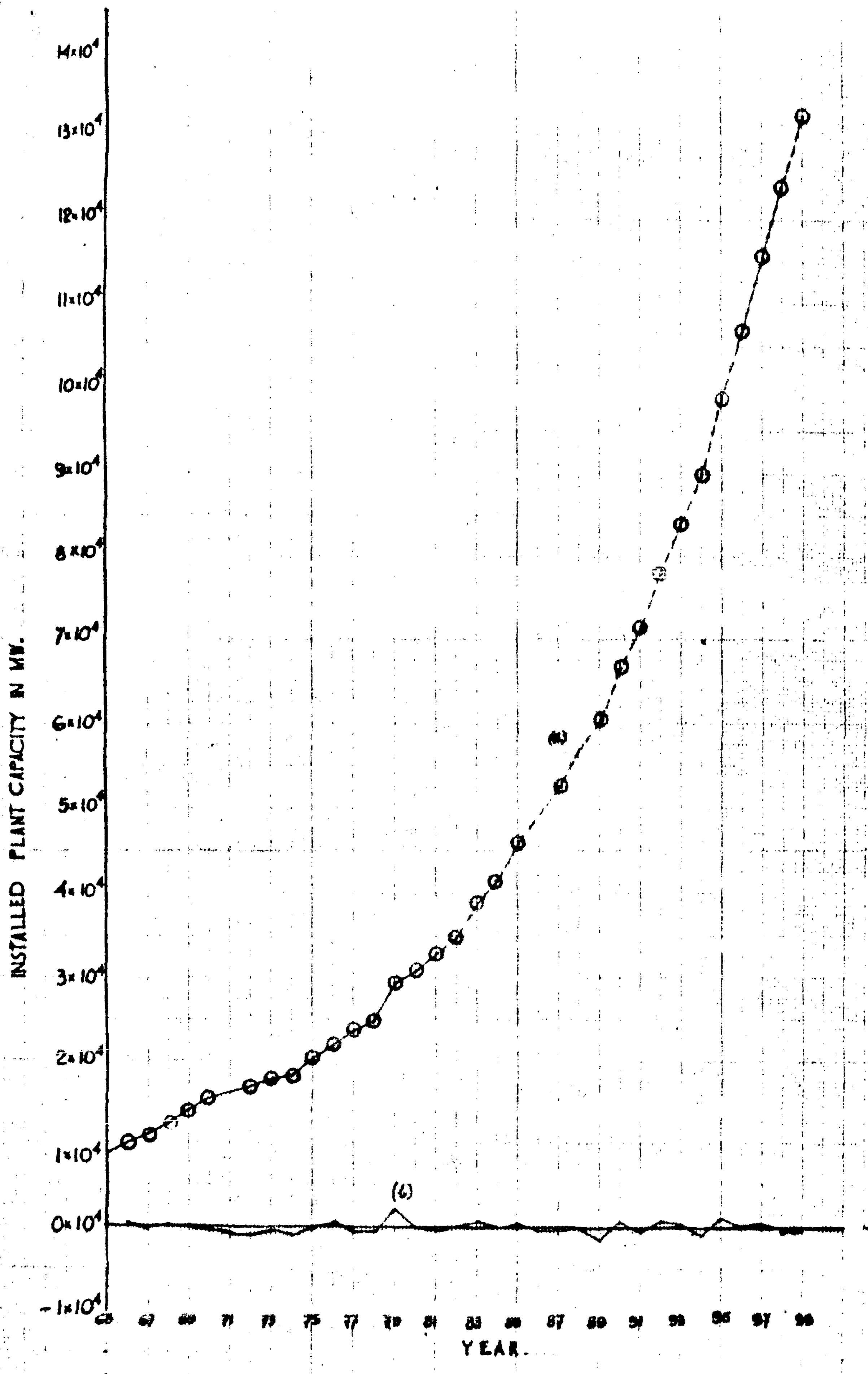


FIG. 5.2.1a & b INSTALLED PLANT CAPACITY AND ERRORS BETWEEN THE OBSERVED AND THE MODELLED VALUES.

there will be a significant improvement of the quality of life of the Indian people if the energy consumption is increased to five times its 1981 figures. All reasonable solutions to alleviate the multitudes of complex problems involving population growth, economies, energy, development, transport and communication involve sharp increases both in the amount of energy consumed and in the efficiency of their use.

A long term energy consumption model has been postulated in the form of a finite difference equation as stated below :

$$Y_{k+1} = f\ (Y_k,\ Y_{k-1},\ Y_{k-2},\ Y_{k-3},\ t_k\ ) \qquad (5.3.1)$$

The arguments are selected on the basis of the correlation with the output Y_{k+1}. The polynomial model of optimum complex has been obtained as

$$\begin{aligned} Y_{k+1} = & - 62.374577 + 1.115110\ Y_k \\ & + 0.068099\ Y_{k-1} - 0.114979\ Y_{k-2} \\ & + 0.028869\ Y_{k-3} - 128.655721\ t_k \\ & - 3.701969\ E-08\ Y_k\ Y_{k-1} \\ & + 4.507208\ E-12\ Y_k\ Y_{k-2} \end{aligned} \qquad (5.3.2)$$

The corresponding integral square error and mean error are 7.162363 E- 06 and 3.409563 E- 02.

The errors are found to be uncorrelated for $k \neq j$. Fig. 5.3.1a and 5.3.1b show the extrapolated annual electrical energy consumption and the errors of modelling.

5.4 Polynomial model with periodicity terms for annual plant load factor

The purpose of this part of the work is to obtain a polynomial model for annual plant load factor. The observed data are processed according to the method stated below.

The annual measured data for load factor are

$P(i), i = 1,2, \ldots, N_1$

i being the instant in years.

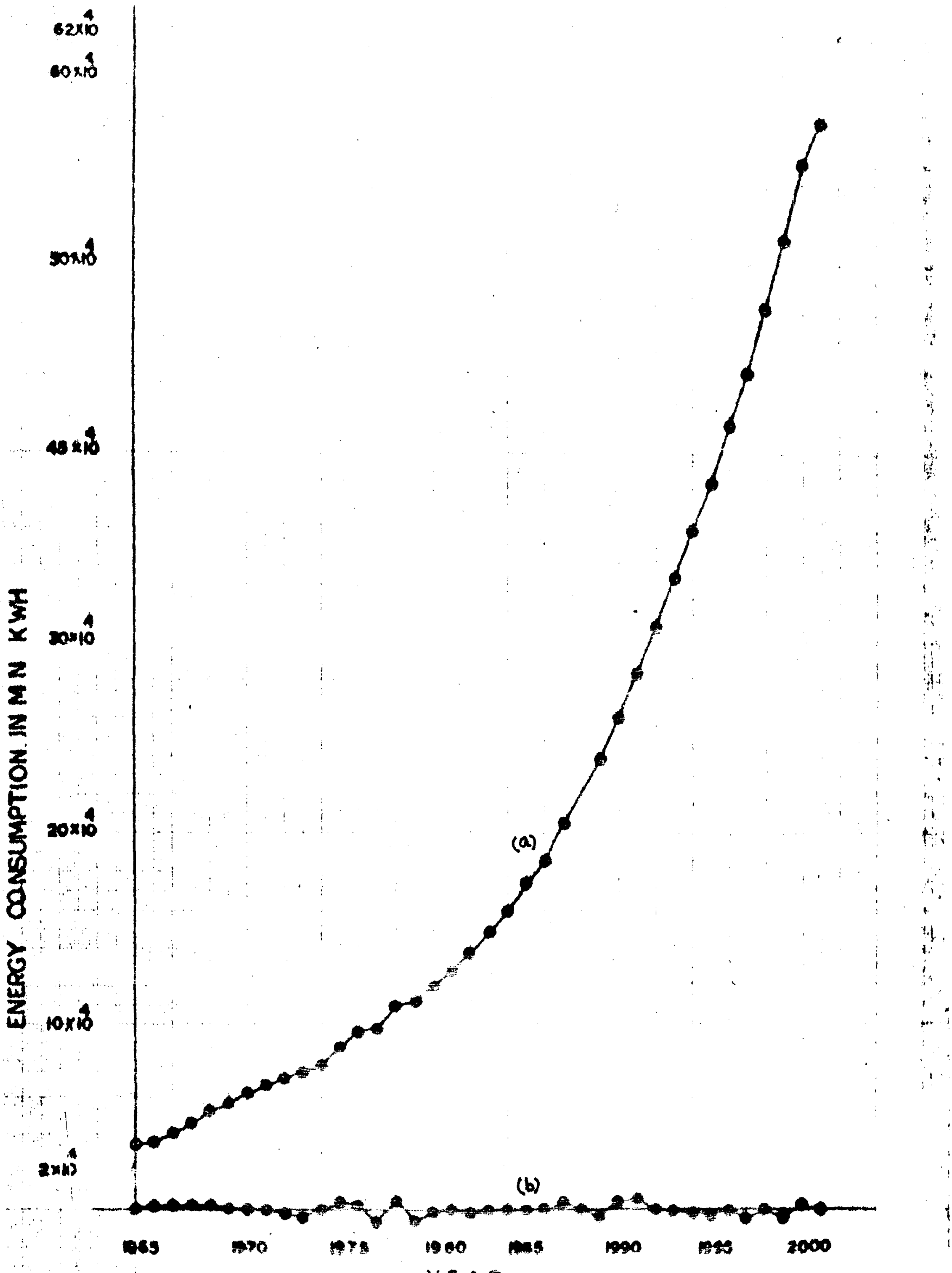
The mean value is given by

$$\bar{P} (N_1) = \frac{1}{N_1} \sum_{i=1}^{N_1} P(i) \qquad (5.4.1)$$

Autocovariance of the data at lag instant k is given by

$$CA(k) = \frac{1}{N_1-k} \sum_{i=1}^{N_1-k} \left[P(i) - \bar{P}(N_1) \right] \left[P(i+k) - \bar{P}(N_1) \right] \qquad \ldots (5.4.2)$$

where $k = 0, 1,2, \ldots, M$ and $M < \frac{1}{4} N_1$.



ANNUAL ENERGY CONSUMPITION EXTRA POLATED FROM 1981-2001 AND ERRORS, BETWEEN THE OBSERVED AND MODELLED VALUES RESPECTIVELY. FIG 5.3.1a & 5.3.1b

The normalised co-efficients of covariance are given by

$$RA(k) = \frac{GA(k)}{GA(o)} \qquad (5.4.3)$$

The estimate of the normalised power density spectra for the data are given by

$$PS(W_h) = \frac{2}{\pi} \sum_{k=0}^{M} E_k \, RA(k) \cos W_h k \qquad (5.4.4)$$

where $W_h = 2\pi f_h$, $f_h = \frac{h}{2M}$; $0 \leqslant f_h \leqslant 0.5$

$$h = 0, 1, 2, \ldots, M$$

E_k has been defined as the weight for window correction and may be taken as

$$E_k = 1.0 \quad \text{for} \quad 0 < k < M$$
$$ 0.5 \quad \text{for} \quad k = 0, M \qquad (5.4.5)$$

These raw estimate of power spectral density are smoothed by using Hamming Window to obtain the final estimates of the power spectrum. The smoothed estimates of the ordinates of the power spectrum are

$S(W_h) = 0.54\ PS(W_h) + 0.46\ PS(W_1)$; for $h = 0$

$S(W_h) = 0.23\ PS(W_{h-1}) + 0.54\ PS(W_h)$

$+ 0.23\ PS(W_{h+1})$; for $0 < h < M$

$S(W_h) = 0.54\ PS(W_h) + 0.46\ PS(W_{h-1})$; for $h = M$

... (5.4.6)

The periodicity in terms of fundamental and its harmonics can be estimated from the power spectral density - frequency characteristics as shown in Fig. 5.4.1. The process has been found to have a 10 yearly cycle (i.e., 0.1 cycle per annum). Only one lag instant of the annual plant load factor has been found to be strongly correlated with current instant of the annual plant load factor.

Consequently the functional model of the annual plant load factor in functional form is given by

$$P(k) = f(P(k-1),\ \sin(2\pi f_o k),\ \cos(2\pi f_o k),$$

$$\sin(2\ f_o(k-1)),\ \cos(2\pi f_o(k-1))) + \eta(k)$$

... (5.4.7)

Let $P(k) = y, P(k-1) = x_1,\ \sin(2\pi f_o k) = x_2,$

$\cos 2\pi f_o k = x_3,\ \sin 2\pi f_o(k-1) = x_4,$

$\cos 2\pi f_o(k-1) = x_5$ and $\eta(k) = \xi$

Equation (5.4.7) becomes

$$y = f(x_1, x_2, x_3, x_4, x_5) + \xi \quad (5.4.8)$$

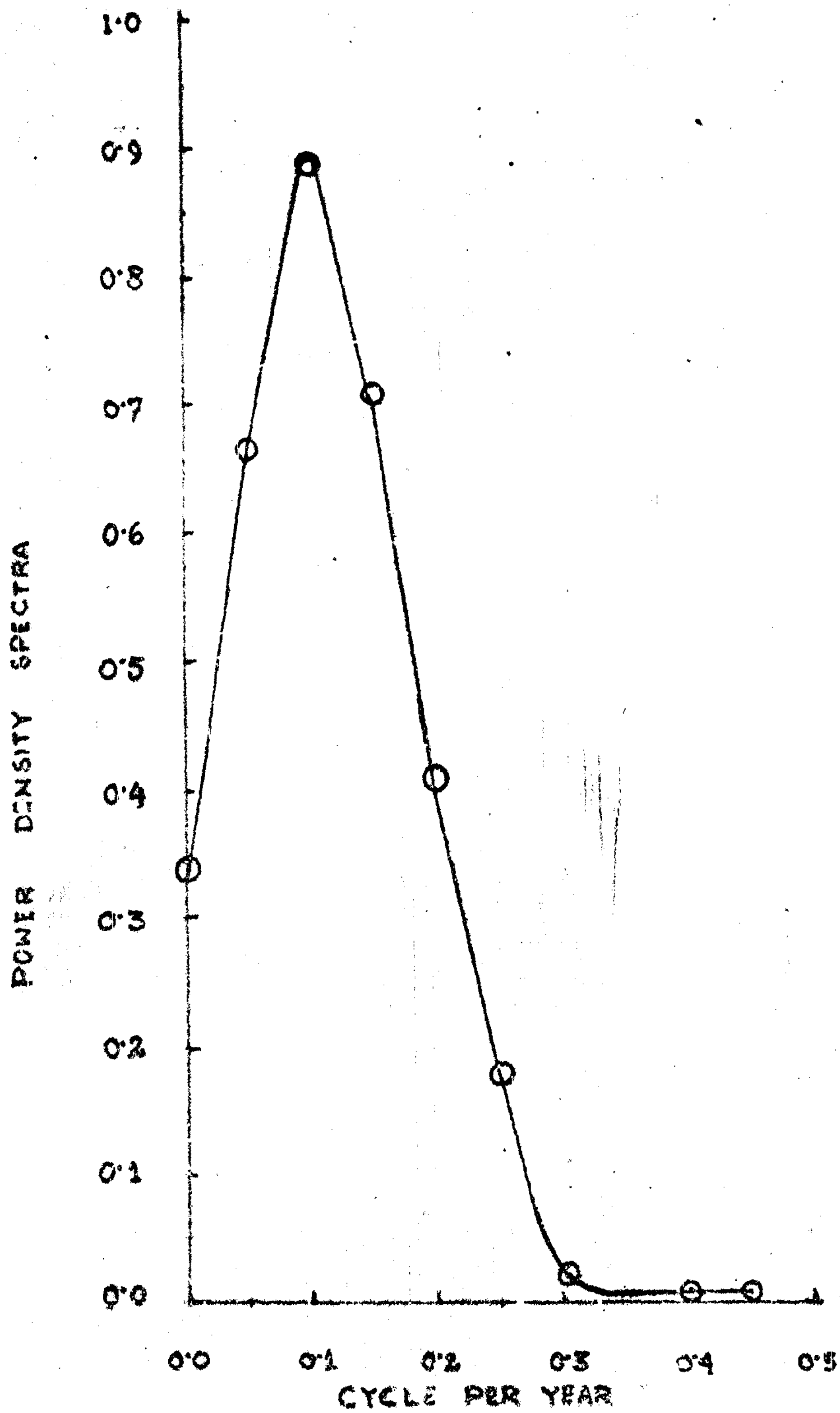


FIG-5.4.1 POWER DENSITY SPECTRA VS CYCLE PER ANNUM

The estimate of y as $\hat{y}$ is defined as

$$\hat{y} = f\ (x_1, x_2, x_3, x_4, x_5) \qquad (5.4.9)$$

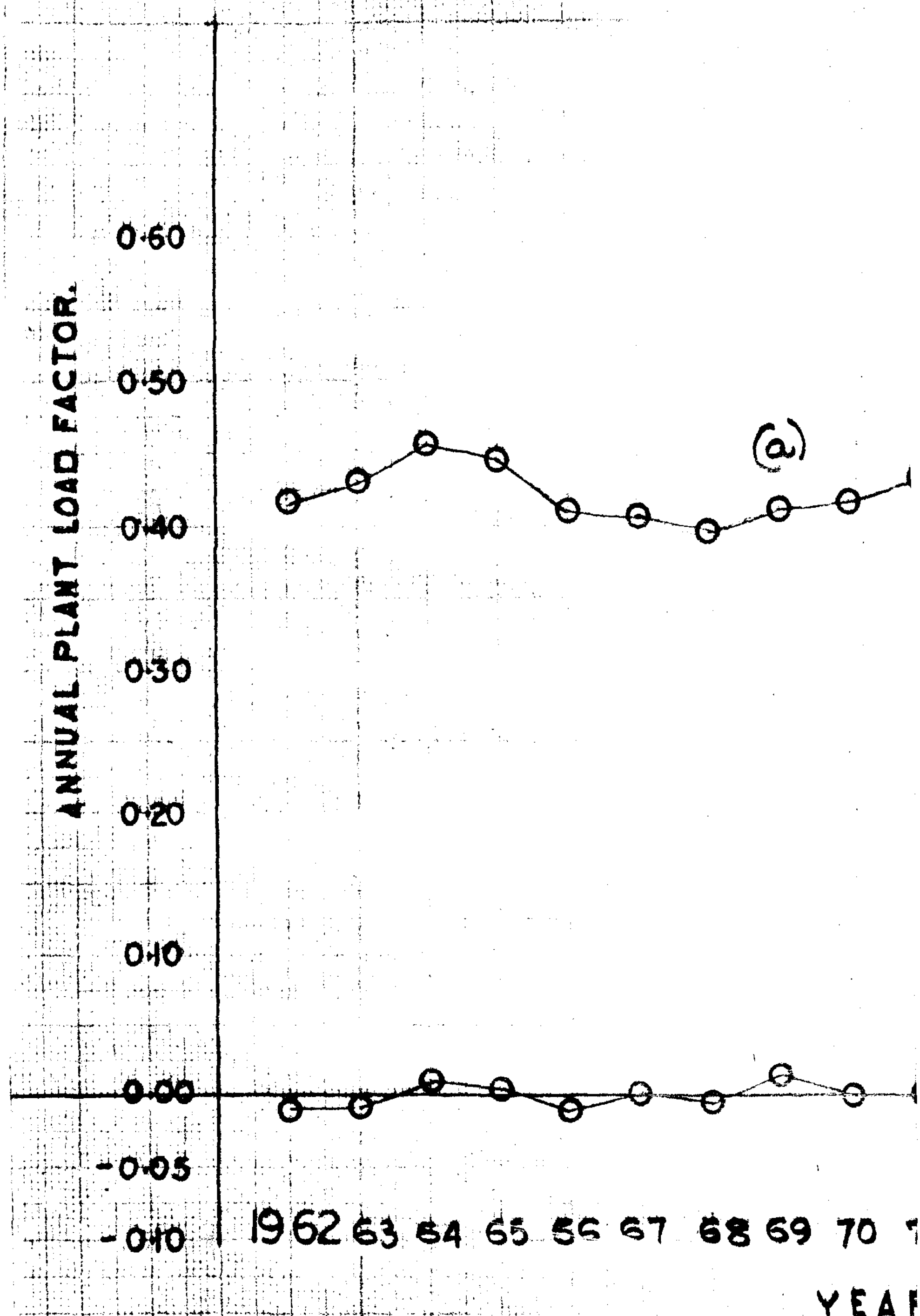
The function $f(\cdot)$ in equation (5.4.9) is sought in the class of quadratic polynomials on the basis of a Table of polynomials of gradually increasing complexity of five variables as shown in Table 5.2.1 with the theory of self-organisation of mathematical models. The model of optimum complexity is selected on the basis of minimum integral square error criterion as defined in equation (5.1.3).

The model of optimum complexity is obtained as

$$\begin{aligned} P(k) = {} & 0.215817 + 0.502090\, P(k-1) \\ & + 17.065098 \sin 0.2\pi k \\ & - 16.103889 \cos 0.2\pi k \\ & - 23.161123 \sin 0.2\pi (k-1) \\ & + 2.964121 \cos 0.2\pi (k-1) \\ & - 17.073073\, P(k-1) \sin 0.2\pi k \\ & + 12.529761\, P(k-1) \cos 0.2\pi k \\ & + 20.968455\, P(k-1) \sin 0.2\pi (k-1) \\ & - 4.75\,E{-12}\, P(k-1) \cos 0.2\pi (k-1) \end{aligned} \qquad \ldots (5.4.10)$$

The corresponding integral square error and mean error are 9.242190 E-04 and 1.808587 E-03. $\eta(\cdot)$ is found to be almost uncorrelated for $k \neq j$ and with variance at 0.999892.

Fig. 5.4.2a and 5.4.2b show the observed values of the plant load factor and the errors between the observed and the modelled values respectively [90]. The software developed in BASIC language is given the Appendix as A3.1 and A3.2.



ANNUAL PLANT LOAD FACTOR

AND MODELLED VALUES RESPEC

FIG 5.4.2a & 5.4.2

CHAPTER VI

STATES ESTIMATION OF ELECTRICAL POWER SYSTEM BY A TRACKING ALGORITHM

CHAPTER VI

STATE ESTIMATION OF ELECTRICAL POWER SYSTEM BY A TRACKING ALGORITHM

6.0 Introduction

The reliable operation of a power network depends on a real time data base for monitoring, security and control of power system. State estimation programmes can enhance the data base for on-line real time operation of power networks. The basic function of an estimation programme is to convert the telemetered raw measurement data into a reliable information base. The information base contains all complex bus voltages, power and current flows as well as injections along with network status and parameters errors. For a successful operation of a state estimator it is essential that the measurement system must have a degree af redundancy greater than unity (i.e., (number of measurements)/(number of state variables)>1). Because the redundancy in the measurement set improves state estimation accuracy.

6.1 State Estimation Tracking Algorithm

The System Equation is

$Z = f(x) + \eta$ for the k-th iteration

$$Z(k) = f(x(k)) + \eta_k \tag{6.1.1}$$

Let

$$\Theta = E\{\eta\eta^T\} \tag{6.1.2}$$

f(x) is a non-linear function of x, and S is defined as a positive, definite weighting matrix as

$$S = E\{(x - \bar{x})(x - \bar{x})^T\} \tag{6.1.3}$$

The objective function is formulated as for the k-th iteration

$$J_k(x(k)) = \sum_{j=1}^{k} [Z(k) - f(x_k)] \Theta^{-1}(k) [Z(k) - f(x_k)]$$

$$+ [x - x(o)]^T S^{-1}(o)(x-x(o)] \tag{6.1.4}$$

x(o) is a priori estimate x and S(o) is the initialised positive definite weighting matrix.

By Taylor's series expansion, neglecting higher order terms

$$Z(k) - f(x_k) = \Delta Z(k) - \Delta x^T F(x_{k-1}) \tag{6.1.5}$$

$$\therefore J_k(x_k) = \sum_{j=1}^{k} [\Delta Z(j) - \Delta x^T F(x_{j-1})]^T \theta^{-1}$$

$$[\Delta Z(j) - \Delta x^T F(x_{j-1})] + \Delta x^T S^{-1}(o)\, \Delta x \qquad (6.1.6)$$

For minimisation

$$\frac{\delta J_k(x_k)}{\delta(\Delta x)} = 0 = -2 \sum_{j=1}^{k} F(x_{j-1})\, \theta^{-1} [\Delta Z(j) - \Delta x^T F(x_{j-1})]$$

$$+ 2\, S^{-1}(o)\, \Delta x \qquad (6.1.7)$$

or,
$$\sum_{j=1}^{k} F(x_{j-1})\, \theta^{-1} \Delta Z(j) + S^{-1}(o)\Delta x = \sum_{j=1}^{k} F(x_{j-1})\, \theta^{-1} F^T(x_{j-1})\, \Delta x$$

$$\ldots (6.1.8)$$

Let
$$S^{-1}(k) = \sum_{j=1}^{k} F(j-1)\, \theta^{-1} F^T(j-1) \qquad (6.1.9)$$

and
$$d(k) = \sum_{j=1}^{k} F(j-1)\, \theta^{-1} \Delta Z(j) + S^{-1}(o)\, \Delta x \qquad (6.1.10)$$

So,
$$d(k+1) = d(k) + F(k)\, \theta^{-1}\, \Delta Z(k+1) \qquad (6.1.11)$$

Let the estimate of Δx at the k-th iteration is $\Delta x(k)$

$$\therefore \quad S^{-1}(k)\,\Delta x(k) = d(k) \tag{6.1.12}$$

From equation (6.1.9)

$$S^{-1}(k+1) = S^{-1}(k) + F(k)\,e^{-1}\,F^{T}(k) \tag{6.1.13}$$

By Matrix inversion lemma,

Let

$$M = S(k) - \frac{S(k)\,F(k)\,e^{-1}\,F^{T}(k)\,S(k)}{1+F^{T}(k)S(k)F(k)\,e^{-1}} \tag{6.1.14}$$

Multiplying equation (6.1.14) by equation (6.1.13)

$$M\,S^{-1}(k+1) = I+S(k)F(k)\,e^{-1}\,F^{T}(k) - \frac{S(k)F(k)\,e^{-1}F^{T}(k)+S(k)F(k)\,e^{-1}F^{T}(k)S(k)F(k)\,e^{-1}F^{T}(k)}{1+F^{T}(k)S(k)F(k)\,e^{-1}} \quad \ldots (6.1.15)$$

Since e^{-1} and $F^{T}(k)S(k)F(k)$ are scalar

$$S(k)F(k)\,e^{-1}F^{T}(k)+S(k)F(k)\,e^{-1}\,F^{T}(k)S(k)F(k)\,e^{-1}\,F^{T}(k)$$

$$M\,S^{-1}(k+1) - I = \frac{-S(k)F(k)e^{-1}F^{T}(k)-S(k)F(k)e^{-1}F^{T}(k)S(k)F(k)e^{-1}F^{T}(k)}{1+F^{T}(k)S(k)F(k)\,e^{-1}} \quad \ldots (6.1.16)$$

So,

$$M\, S^{-1}(k+1) = I \qquad (6.1.17)$$

$$\therefore\; M = S(k+1) \qquad (6.1.18)$$

$$\therefore\; S(k+1) = S(k) - \frac{S(k)\,F(k)\,\theta^{-1}F^{T}(k)S(k)}{1+F^{T}(k)S(k)F(k)\,\theta^{-1}} \qquad (6.1.19)$$

$$\Delta x(k+1) = S(k+1)\, d(k+1) \qquad (6.1.20)$$

From equation (6.1.19) and (6.1.11),

$$\Delta x(k+1) = \left[S(k) - \frac{S(k)F(k)\,\theta^{-1}F^{T}(k)\,S(k)}{1+F^{T}(k)S(k)F(k)\,\theta^{-1}} \right] \left[d(k) + F(k)\,\Delta Z(k+1)\,\theta^{-1} \right] \qquad (6.1.21)$$

$$\Delta x(k+1) = \Delta x(k) + S(k)F(k)\Delta Z(k+1)\,\theta^{-1} - \frac{S(k)F(k)\,\theta^{-1}F^{T}(k)\Delta x(k)+S(k)F(k)\,\theta^{-1}F^{T}(k)S(k)F(k)\,\theta^{-1}\Delta Z(k+1)}{1+F^{T}(k)S(k)F(k)\,\theta^{-1}}$$

$$\Delta x(k+1) = \Delta x(k) + S(k)F(k)\,\theta^{-1}\left[\Delta Z(k+1) - \Delta x^{T}(k)F(k)\right] \times \left[1+F^{T}(k)S(k)F(k)\,\theta^{-1}\right]^{-1} \qquad (6.1.22)$$

So the algorithms for the recursive estimation of states of the power system are

$$\Delta x(k+1) = \Delta x(k) + S(k)F(k)\,\theta^{-1}[\Delta Z(k+1) - \Delta x^T(k)F(k)] \times [1 + F^T(k)S(k)F(k)\,\theta^{-1}]^{-1} \quad (6.1.23a)$$

$$S(k+1) = S(k) - S(k)F(k)\,\theta^{-1}F^T(k)S(k) \times [1 + F^T(k)S(k)F(k)\,\theta^{-1}]^{-1} \quad (6.1.23b)$$

6.2 Implementation of the Algorithm

Step 1

Individual measurements are considered one after another as a scalar quantity, e.g. z_1, z_2, z_3, ... and so on. θ^{-1} is the variance of measurements of individual quantity and it is assumed to be constant for all the measured variables and its value is taken as 2500. S - is initialized as a positive definite weighting matrix of diagonals as 0.0004 and all non-diagonals as zero. Initial values of the state vectors are taken as those values obtained from an off-line load flow analysis with reactive component of the voltage of one bus as zero. $F(.)$ is the corresponding column of the Jacobian.

Step 2

Deviations of P and Q of the measured from the calculated ones are obtained for all the buses such as

$\Delta P_1(k) = P_1 - P_1(k)$ calculated, $\Delta Q_1(k) = Q_1 - Q_1(k)$ calculated.

For a 5 bus system we have $\Delta Z(k)$ as

$$[\Delta P_1(k), \Delta P_2(k), \Delta P_3(k), \Delta P_4(k), \Delta P_5(k), \Delta Q_1(k), \Delta Q_2(k), \Delta Q_3(k), \Delta Q_5(k)]^T$$

where k- is the iteration count.

The state vectors at the k-th iteration are denoted as

$\Delta e_1(k), \Delta e_2(k), \ldots, \Delta e_5(k), \Delta f_2(k), \ldots, \Delta f_5(k) \ldots;$ since

$\Delta f_1(k) = 0$ for all k.

Elements of the Jacobian matrix (9 x 9) are computed for the k-th iteration as $F(k)$.

To introduce recursiveness in the algorithm and for possible on-line application P and Q are initialised as the measured quantities and as the iteration proceeds P and Q become one step back af the calculated values of P and Q as stated as

$$\Delta P_1(k) = P_1(k-1) - P_1(k) \text{ and } \Delta Q_1(k) = Q_1(k-1) - Q(k).$$

Step 3

Iteration proceeds with the algorithms for the state vectors as

$$\begin{bmatrix} S_{11}(k+1), S_{12}(k+1), \ldots, S_{19}(k+1) \\ \vdots \\ S_{91}(k+1), S_{92}(k+1), \ldots, S_{99}(k+1) \end{bmatrix}$$

$$= \begin{bmatrix} S_{11}(k), S_{12}(k), \ldots, S_{19}(k) \\ \vdots \\ S_{91}(k), S_{92}(k), \ldots, S_{99}(k) \end{bmatrix} - \begin{bmatrix} S_{11}(k), S_{12}(k), \ldots, S_{19}(k) \\ \vdots \\ S_{91}(k), S_{92}(k), \ldots, S_{99}(k) \end{bmatrix} \times$$

$$\times \begin{bmatrix} F_{11}(k) \\ F_{21}(k) \\ \vdots \\ F_{91}(k) \end{bmatrix} e^{-1} \begin{bmatrix} F_{11}(k), F_{21}(k), \ldots, F_{91}(k) \end{bmatrix} \begin{bmatrix} S_{11}(k), S_{12}(k), \ldots, S_{19}(k) \\ \vdots \qquad\qquad \vdots \\ S_{91}(k), S_{92}(k), \ldots, S_{99}(k) \end{bmatrix} \times$$

$$\left\{ 1 + \left[F_{11}(k), F_{21}(k), \ldots, F_{91}(k) \right] \begin{bmatrix} S_{11}(k), S_{12}(k), \ldots, S_{19}(k) \\ \vdots \\ S_{91}(k), S_{92}(k), \ldots, S_{99}(k) \end{bmatrix} \begin{bmatrix} F_{11}(k) \\ F_{21}(k) \\ \vdots \\ F_{91}(k) \end{bmatrix} e^{-1} \right\}$$

... (6.2.1a)

$$
\begin{bmatrix} \Delta e_1(k+1) \\ \Delta e_2(k+1) \\ \Delta e_3(k+1) \\ \Delta e_4(k+1) \\ \Delta e_5(k+1) \\ \Delta f_2(k+1) \\ \Delta f_3(k+1) \\ \Delta f_4(k+1) \\ \Delta f_5(k+1) \end{bmatrix}
=
\begin{bmatrix} \Delta e_1(k) \\ \cdot \\ \cdot \\ \cdot \\ \Delta e_5(k) \\ \Delta f_2(k) \\ \\ \\ \Delta f_5(k) \end{bmatrix}
+
\begin{bmatrix} S_{11}(k), & \ldots, & S_{19}(k) \\ S_{21}(k), & \ldots, & S_{29}(k) \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \\ \\ \\ S_{91}(k), & \ldots, & S_{99}(k) \end{bmatrix}
\begin{bmatrix} F_{11}(k) \\ F_{21}(k) \\ \cdot \\ \cdot \\ \cdot \\ \\ \\ \\ F_{91}(k) \end{bmatrix} \theta^{-1} \; x
$$

$$
\Big[P_1(k+1) - \left[\Delta e_1(k), \ldots, \Delta e_5(k), \Delta f_2(k), \ldots, \Delta f_5(k) \right]
$$

$$
\times \left[F_{11}(k), F_{21}(k), \ldots, F_{91}(k) \right]^T \Big] \; x
$$

$$
\left\{ 1 + \left[F_{11}(k), F_{21}(k), \ldots, F_{91}(k) \right]
\begin{bmatrix} S_{11}(k), \ldots, S_{19}(k) \\ \cdot \\ \cdot \\ \cdot \\ S_{91}(k), \ldots, S_{99}(k) \end{bmatrix}
\begin{bmatrix} F_{11}(k) \\ \cdot \\ \cdot \\ \cdot \\ F_{91}(k) \end{bmatrix} \theta^{-1} \right\}
$$

... (6.2.1b)

Now, $e_1(k+1) = e_1(k) + \Delta e_1(k+1)$ (6.2.2)

$\cdot$

$\cdot$

$\cdot$

$e_5(k+1) = e_5(k) + \Delta e_5(k+1)$

$f_2(k+1) = f_2(k) + \Delta f_2(k+1)$

$\cdot$

$\cdot$

$\cdot$

$f_5(k+1) = f_5(k) + \Delta f_5(k+1)$

Step 4

Go to step 2, if max $\Delta P_i(k+1)$ <= tolerance and if max $\Delta Q_i(k+1)$ <= tolerance, then go to step 5 Else GO TO Step 2.

Step 5

Print results.

A 5-Bus 7 lines network [67] was considered. For tolerance of 0.01 state vectors were found to converge after 3 iterations [91]. The power network under consideration has been shown in Fig.6.2.1. Line parameters, line Admittance, Bus Admittance Matrix, Initial values of the States Vectors, Active and Reactive Bus Power Measurements are shown in Tables 6.2.1, 6.2.2, 6.2.3a and 6.2.3b,

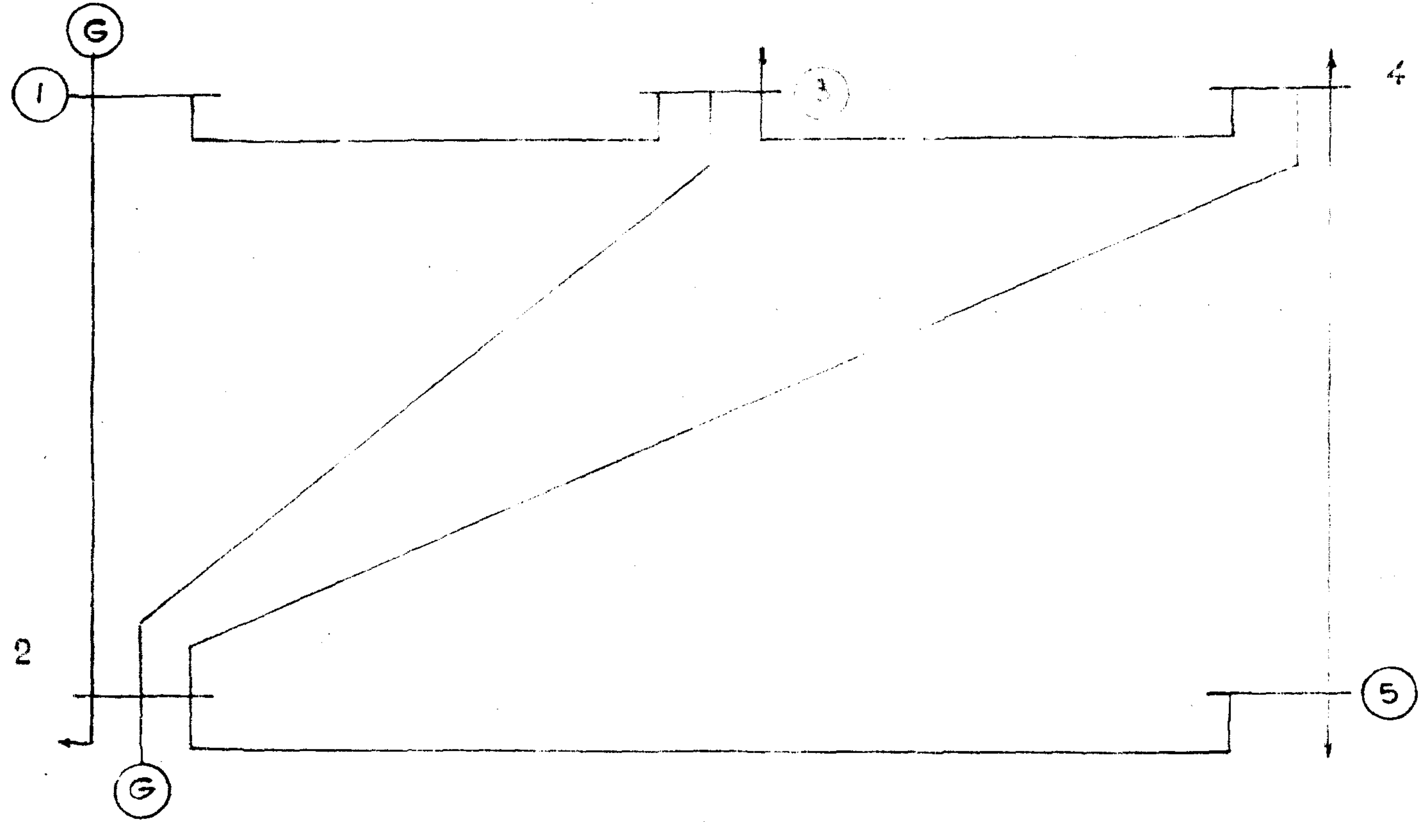


FIG. 6.2.1 : SAMPLE POWER NETWORK.

6.2.4. For iteration 1 elements of the Jacobian, the weighting Matrix S(I,J), the States Vectors e(.) and f(.) are shown in Table 6.2.5, 6.2.6 and 6.2.7 respectively.

For iteration 2, the elements of the Jacobian, the weighting matrix S (I,J), the States Vectors are shown in Tables 6.2.8, 6.2.9, and 6.2.10 respectively.

For iteration 3 , the weighting matrix S (I, J) and the States Vectors are shown in Tables 6.2.11 and 6.2.12 respectively. The software developed in BASIC is given in the Appendix as A4.1.

TABLE 6.2.1

LINE PARAMETERS

Line No.	Between Buses	Line impedance	Half line charging admittance
1	1 - 2	0.02 + j 0.06	0.0 + j 0.03
2	1 - 3	0.08 + j 0.24	0.0 + j 0.025
3	2 - 3	0.06 + j 0.18	0.0 + j 0.02
4	2 - 4	0.06 + j 0.18	0.0 + j 0.02
5	2 - 5	0.04 + j 0.12	0.0 + j 0.015
6	3 - 4	0.01 + j 0.03	0.0 + j 0.01
7	4 - 5	0.08 + j 0.24	0.0 + j 0.025

TABLE 6.2.2

LINE ADMITTANCE AND LINE CHARGING

Line No.	Between Buses	Line admittance	Line Half charging
1	1 - 2	5.00 - j 15.00	0.0 + j 0.03
2	1 - 3	1.25 - j 3.75	0.0 + j 0.025
3	2 - 3	1.66 - j 5.00	0.0 + j 0.02
4	2 - 4	1.66 - j 5.00	0.0 + j 0.02
5	2 - 5	2.50 - j 7.50	0.0 + j 0.015
6	3 - 4	10.00 - j 30.00	0.0 + j 0.01
7	4 - 5	1.25 - j 3.75	0.0 + j 0.025

TABLE 6.2.3a

BUS ADMITTANCE

Column / Row	1	2	3	4	5
1	6.25-j18.69	-5.00+j15.00	-1.25+j3.75	0	0
2	-5.00+j15.00	10.83-j32.41	-1.66+j5.00	-1.66+j5.00	-2.50+j7.50
3	-1.25+j3.75	-1.66+j5.00	12.91-j38.69	-10.00+j30.00	0
4	0	-1.66+j5.00	-10.00+j30.00	12.91-j38.69	-1.25+j3.75
5	0	-2.50+j7.50	0	-1.25+j3.75	3.75-j11.21

TABLE 6.2.3b

INITIAL VALUES OF THE STATE VECTOR

Bus No.	Active Voltage	Reactive Voltage
1	1.060	-0.000
2	1.0463	-0.0513
3	1.0204	-0.0828
4	1.0193	-0.0951
5	1.0123	-0.1091

TABLE 6.2.4

ACTIVE AND REACTIVE BUS POWER MEASUREMENT

Bus No.	Generation		Load	
	Megawatts	Megavars	Megawatts	Megavars
1	129.5	0	0	7.5
2	40	30	20	10
3	0	0	45	15
4	0	0	40	5
5	0	0	60	10

TABLE 6.2.3

ELEMENTS OF THE JACOBIAN MATRIX FOR FIRST ITERATION

Column / Row	1	2	3	4	5
1	7.847	- 5.3	- 1.325	0	0
2	- 6.001	13.1793	- 2.0003	- 2.0003	- 3.0005
3	- 1.61	- 2.1466	16.808	-12.88	0
4	0	- 2.1743	-13.046	16.4603	- 1.6307
5	0	- 3.349	0	- 1.6745	4.4437
6	19.7444	-15.9	- 3.975	0	0
7	-15.438	33.5609	- 5.146	- 5.146	- 7.719
8	- 3.715	- 4.9533	38.1467	-29.72	0
9	0	- 4.938	-29.628	38.1301	- 3.7035
10	0	- 7.3195	0	- 3.6597	10.7790

TABLE 6.2.3 (Continued)

Column / Row	6	7	8	9	10
1	19.889	-15.5	- 3.975	0	0
2	-15.438	32.6034	- 5.146	- 5.146	7.719
3	- 3.175	- 4.9533	37.3656	-39.72	0
4	0	- 4.938	-29.628	37.0683	-3.7035
5	0	- 7.3195	0	- 3.6597	10.6893
6	- 5.403	5.30	1.325	0	0
7	6.001	-12.8162	2.0003	2.0003	3.0005
8	1.61	5.1466	-17.0555	12.88	0
9	0	2.1743	13.046	-17.2313	1.6307
10	0	3.349	0	1.6745	-5.5945

TABLE 6.2.6

WEIGHTING MATRIX S(I,J) FOR FIRST ITERATION

Column / Row	1	2	3	4	5
1	7.3842E-05	7.3721E-05	7.5213E-05	7.5217E-05	7.5479E-05
2	7.3721E-05	7.4868E-05	7.6450E-05	7.6530E-05	7.7119E-05
3	7.5213E-05	7.6450E-05	7.9358E-05	7.9366E-05	7.8686E-05
4	7.5217E-05	7.6530E-05	7.9366E-05	7.9536E-05	7.8942E-05
5	7.5479E-05	7.7119E-05	7.8686E-05	7.8942E-05	8.1667E-05
6	3.4676E-06	3.5096E-06	3.6074E-06	3.6125E-06	3.6374E-06
7	6.1504E-06	6.2260E-06	6.4681E-06	6.4726E-06	6.3929E-06
8	6.5610E-06	6.6442E-06	6.8964E-06	6.9036E-06	6.8360E-06
9	7.1706E-06	7.2881E-06	7.4581E-06	7.4726E-06	7.6729E-06

TABLE 6.2.6 (Continued)

Column / Row	6	7	8	9
1	3.4676E-06	6.1504E-06	6.5610E-06	7.1706E-06
2	3.5096E-06	6.2260E-06	6.6442E-06	7.2881E-06
3	3.6074E-06	6.4681E-06	6.8964E-06	7.4581E-06
4	3.6125E-06	6.4726E-06	6.9036E-06	7.4726E-06
5	3.6374E-06	6.3929E-06	6.8360E-06	7.6729E-06
6	8.4216E-07	1.0593E-06	1.1560E-06	1.4291E-06
7	1.0593E-06	2.7378E-06	2.7828E-06	1.7990E-06
8	1.1560E-06	2.7828E-06	2.9945E-06	2.0210E-06
9	1.4291E-06	1.7990E-06	2.0210E-06	4.3106E-06

TABLE 6.2.7

STATES VECTORS e(.), f(.) FOR THIRD ITERATION

Bus No.	Active voltage	Reactive voltage	Voltage Magnitude	Angle
1	1.0606	0	1.0606	0
2	1.0389	-0.0527	1.0403	-2.9062
3	1.0104	-0.0901	1.0144	-5.0963
4	1.0101	-0.0964	1.0147	-5.4516
5	1.0160	-0.1160	1.0226	-6.5127

TABLE 6.2.8

ELEMENTS OF THE JACOBIAN MATRIX FOR THE SECOND ITERATION

Column / Row	1	2	3	4	5
1	7.8049	- 5.2532	- 1.3133	0	0
2	- 5.9867	13.1530	- 1.9955	- 1.9955	- 2.9923
3	- 1.6010	- 2.1347	16.1144	-12.8067	0
4	0	- 2.1655	-12.9935	16.3909	- 1.6241
5	0	- 3.4103	0	- 1.7061	4.5180
6	19.5330	-15.7597	- 3.9399	0	0
7	-15.3208	33.2731	- 5.1069	- 5.1068	- 7.6604
8	- 3.6766	- 4.9022	37.7110	-29.4132	0
9	0	- 4.8898	-29.3398	37.7185	- 3.6674
10	0	- 7.3306	0	- 3.6653	10.9478

TABLE 6.2.8 (Continued)

Column / Row	6	7	8	9	10
1	19.7507	-15.7597	- 3.9399	0	0
2	-15.3208	32.3686	- 5.1069	- 5.1069	- 7.6604
3	- 3.6766	- 4.9022	36.9977	-29.4132	0
4	0	- 4.8898	-29.3398	36.7180	- 3.6674
5	0	- 7.3306	0	- 3.6653	10.5876
6	- 5.3281	5.2532	1.3133	0	0
7	5.9867	-12.7806	1.9955	1.9955	2.9923
8	1.6010	2.1347	-16.9649	12.8067	0
9	0	2.1655	12.9935	-17.1651	1.6241
10	0	3.4103	0	1.7061	- 5.7036

TABLE 6.2.9

WEIGHTING MATRIX S(I,J) FOR SECOND ITERATION

Column / Row	1	2	3	4	5
1	7.0319E-05	7.1027E-05	7.2805E-05	7.2832E-05	7.2774E-05
2	7.1027E-05	7.2079E-05	7.3918E-05	7.3984E-05	7.4084E-05
3	7.2805E-05	7.3918E-05	7.6458E-05	7.6489E-05	7.5946E-05
4	7.2832E-05	7.3984E-05	7.6489E-05	7.6602E-05	7.6100E-05
5	7.2774E-05	7.4084E-05	7.5946E-05	7.6100E-05	7.7253E-05
6	3.4389E-06	3.4839E-06	3.5844E-06	3.5884E-06	3.5945E-06
7	6.0264E-06	6.1056E-06	6.3167E-06	6.3212E-06	6.2674E-06
8	6.4453E-06	6.5313E-06	6.7534E-06	6.7601E-06	6.7113E-06
9	7.2996E-06	7.4121E-06	7.6053E-06	7.6151E-06	7.7107E-06

TABLE 6.2.9 (Continued)

Column / Row	6	7	8	9
1	3.4388E-06	6.0264E-06	6.4453E-06	7.2996E-06
2	3.4839E-06	6.1056E-06	6.5313E-06	7.4121E-06
3	3.5844E-06	6.3167E-06	6.7534E-06	7.6052E-06
4	3.5884E-06	6.3213E-06	6.7601E-06	7.6151E-06
5	3.5945E-06	6.2674E-06	6.7113E-06	7.7107E-06
6	5.1246E-07	6.2572E-07	7.4565E-07	9.1282E-07
7	6.2527E-07	1.6427E-06	1.6845E-06	1.2067E-06
8	7.4565E-07	1.6845E-06	1.8118E-06	1.3754E-06
9	9.1282E-07	1.2067E-06	1.3754E-06	2.5959E-06

TABLE 6.2.10

STATES VECTORS e(.), f(.) FOR SECOND ITERATION

Bus No.	Active Voltage	Reactive Voltage	Voltage Magnitude	Angle
1	1.0545	0	1.0545	0
2	1.0420	-0.0521	1.0433	-2.8653
3	1.0146	-0.0897	1.0186	-5.0538
4	1.0139	-0.0968	1.0185	-5.3996
5	1.0146	-0.1121	1.0209	-6.2614

TABLE 6.2.11

WEIGHTING MATRIX S(I,J) FOR THIRD ITERATION

Column / Row	1	2	3	4	5
1	7.0219E-06	7.1087E-06	7.2806E-06	7.2832E-06	7.2774E-06
2	7.1087E-06	7.2079E-06	7.3918E-06	7.3984E-06	7.4084E-06
3	7.2806E-06	7.3918E-06	7.6488E-06	7.6489E-06	7.5946E-06
4	7.2832E-06	7.3984E-06	7.6489E-06	7.6608E-06	7.6100E-06
5	7.2774E-06	7.4084E-06	7.5946E-06	7.6100E-06	7.7853E-06
6	3.4388E-06	3.4839E-06	3.5844E-06	3.5884E-06	3.5945E-06
7	6.0864E-06	6.1056E-06	6.3167E-06	6.3213E-06	6.2674E-06
8	6.4453E-06	6.5313E-06	6.7534E-06	6.7601E-06	6.7112E-06
9	7.2996E-06	7.4121E-06	7.6055E-06	7.6191E-06	7.7107E-06

TABLE 6.2.11 (Continued)

Column / Row	6	7	8	9
1	3.4388E-06	6.0864E-06	6.4453E-06	7.2996E-06
2	3.4839E-06	6.1056E-06	6.5313E-06	7.4121E-06
3	3.5844E-06	6.3167E-06	6.7534E-06	7.6055E-06
4	3.5884E-06	6.3213E-06	6.7601E-06	7.6191E-06
5	3.5945E-06	6.2674E-06	6.7113E-06	7.7107E-06
6	5.1246E-07	6.8572E-07	7.4565E-07	9.1852E-07
7	6.8572E-07	1.6427E-06	1.6845E-06	1.2087E-06
8	7.4565E-07	1.6845E-06	1.8118E-06	1.3754E-06
9	9.1852E-07	1.2087E-06	1.3754E-06	2.5959E-06

TABLE 6.2.12

STATES VECTORS e(.), f(.) FOR THIRD ITERATION

Bus No	Active Voltage	Reactive Voltage	Voltage Magnitude	Angle
1	1.0545	0	1.0545	0
2	1.0420	-0.0521	1.0433	-2.8653
3	1.0146	-0.0897	1.0186	-5.0528
4	1.0139	-0.0958	1.0185	-5.3996
5	1.0146	-0.1131	1.0209	-6.3614

CHAPTER VII

GAUSS-SEIDEL LOAD FLOW WITH OPTIMALLY ORDERED NODES BY DYNAMIC PROGRAMMING ALGORITHM

CHAPTER VII

GAUSS-SEIDEL LOAD FLOW WITH OPTIMALLY ORDERED NODES BY DYNAMIC PROGRAMMING ALGORITHM

7.0 Introduction

It is desired that transmission system should be able to transmit electric energy economically and reliably from generation centres to all load centres at a generally acceptable voltage level. This necessiates the study of the load flow in a power system to determine steady operating states. Results of the load flow analysis are used for stability analysis and for power system planning, operation and control. A large number of numerical algorithms have been developed over the last 25 years. The most of the algorithms are variations of two numerical technique such as (i) Gauss-Seidel method and (ii) Newton Raphson method. The present effort is an exposure of the Gauss-Seidel method under different bus conditions with optimal ordering of buses by Dynamic programming algorithm. The algorithms are developed in easily understandable manner and illustrated on IEEE 14 bus system.

7.1 Bus Type

Power System Buses are characterised as follows.

(i) P - Q Bus : A P - Q Bus is one where total bus power in complex form is specified. At such a p-th bus the complex power is

$$E_p^* I_p = P_p - jQ_p$$

$$= (P_{Gp} - P_{Lp}) - j(Q_{Gp} - Q_{Lp}) \qquad (7.1.1)$$

where $E_p = e_p + j f_p$ and the subscripts G_p and L_p refer to the generation and load respectively at the p-th bus.

(ii) P - V Bus : A P - V Bus is one where real power P_p is specified and the voltage magnitude is maintained at a constant value. At such a bus the characteristics are

$$\text{Re} \left[E_p^* I_p \right] = P_{Gp} - P_{Lp} \qquad (7.1.2)$$

with

$$\left| E_p \right| = (e_p^2 + f_p^2)^{\frac{1}{2}} \qquad (7.1.3)$$

(iii) Swing Bus or Slack Bus : Swing Bus or Slack Bus is a Bus where the complex voltage is specified. The concept of a swing bus is necessary because in the system the losses are not known in advance, and hence it is not possible to fix injected real power at all the buses. It is the standard practice to designate one of the voltage controlled buses having the largest generation as the swing bus. At this bus the complex power is not specified and is calculated at the end when the load flow calculations are converged. The phase angle at the swing bus is specified and is taken as zero. Hence the swing bus is considered as the reference bus.

7.2 Power System Equation

Bus frame of reference of the line parameters in the admittance form has gained widespread application because of the simplicity of data preparation and the ease with which the bus admittance matrix can be formulated and modified for any subsequent network changes. The method using the bus admittance matrix remains the most economical from the point of view of computer time and memory requirements. The solutions of the aljebraic equations describing the load flow process are based on iterative technique because of the non-linearity in the power equations. The present investigation deals with the Gauss-Seidel iterative technique using Y bus and optimally ordered nodes by Dynamic programming algorithm.

The network equation is written as

$$I_{BUS} = Y_{BUS} \; E_{BUS} \qquad (7.2.1)$$

Y BUS includes line admittance and the effects of shunt elements to ground such as static capacitors and reactors, line charging, shunt elements of transformer equivalents.

Algorithms for calculation of Y_{BUS} is stated below :

(1) Read bus code p - q , impedance z_{pq} , line charging $y'_{pq/2}$ transformer tap 'a' ;

(2) Obtain reciprocal of transmission line impedance z_{pq} to get admittance y_{pq} ;

(3) Obtain total line charging and shunt capacitor at each bus ;

(4) Obtain self admittance at bus p as

$$Y_{pp} = \sum_{q=1}^{n} y_{pq}$$

$$= y_{p1} + y_{p2} + \dots + y_{pn} \qquad (7.2.2)$$

and mutual admittance from p to q as

$$Y_{pq} = - y_{pq}$$

(5) When the off-nominal turns ratio 'a' is represented at bus p for a transformer connecting p and q , the self admittance at bus p is

$$Y_{pp} = y_{p1} + y_{p2} + \ldots + y_{pq/a} + \ldots + y_{pn} + \frac{1}{a}\left(\frac{1}{a} - 1\right) y_{pq} \qquad (7.2.3)$$

The mutual admittance from p to q is

$$Y_{pq} = - \frac{y_{pq}}{a} \qquad (7.2.4)$$

and self admittance at bus q is

$$Y_{qq} = y_{q1} + y_{q2} + \ldots + y_{qp/a} + \ldots + y_{qn} + \left(1 - \frac{1}{a}\right) y_{pq}$$

$$Y_{qq} = y_{q1} + y_{q2} + \ldots + y_{qp} + \ldots + y_{qn} \qquad (7.2.5)$$

7.3 Algorithms for Optimal Ordering

Non zero pattern of the bus admittance matrix is prepared as

IF Y_{ij} YES THEN $Y_{ij} = 1$

OTHERWISE $Y_{ij} = 0$

A simple illustration for preparation of non zero pattern of bus admittance matrix is given as below.

Consider a 14 nodes network of IEEE 14 BUS system. Computational efficiency of load flow analysis depends on the order in which the Gaussian elimination is performed on sparse matrices and total number of new non zero elements are generated in course of elimination. It is observed that the computational efficiency is greatly improved if the nodes are ordered in an optimal way.

The principle of solution of sparsity oriented node ordering problem can be stated as follows.

An initial segment of an optimal ordering is a group of nodes of a network which has the property that their optimally ordered elimination of the remaining nodes in a network constitutes an optimally ordered elimination of all the nodes in a network.

The principle of optimality as stated above is applied to the problem of optimal ordering of sparsity oriented nodes in power system network. This optimisation problem is solved in an iterative procedure by Dynamic Programming algorithm following an optimum decision policy. The objective of the sparsity oriented optimum ordering of nodes is to determine the best possible way of performing Gaussian elimination, so that the amount of fill in or the valency of the elimination is minimum ; the valency of a node is the number of new paths added among the remaining set of nodes as a result of elimination of the node and the valency of an ordering is the total number of new paths generated in the process of performing the node elimination in the order specified.

The objective function for optimal ordering is stated as

$$\text{Minimise } J\left[u(k)\right] = \sum_{k=1}^{n} f\left[x(k), u(k)\right] \qquad (7.3.1)$$

where

k : stage variable. Indicates the step of the elimination algorithm.

x(k) : State variable. Node to be eliminated at stage k.

u(k) ; Decision variable. Pointer indicating which node x(k) is to be eliminated at stage k.

$f\left[\ ,\ \right]$: Valency of node x(k)

n is the total number of nodes to be eliminated. The state and decision variables are related by the state equation,

$$x(k+1) = x(k) + u(k) \qquad (7.3.2)$$

with

$$x(k) \in X(k) = \left\{ i \,/\, i = 1,2,\ldots, n \right\} \qquad (7.3.3)$$

$$x(k) \neq x(i) \quad \text{for} \quad i = 1,2,\ldots,(k-1) \qquad (7.3.4)$$

and

$$u(k) \in U(k) = \left\{ i/i = \pm 1, \pm 2, \ldots, + (n-1) \right\} \qquad (7.3.5)$$

The problem of optimisation is formulated as follows :

Given the state equation (7.3.2) and the constraints (7.3.3), (7.3.4) and (7.3.5), find the control sequence u(1), u(2),..., u(n) that minimises the objective function (7.3.1). The optimisation problem is solved by dynamic programming algorithm.

The computational procedure is implemented as follows :

Initially at stage 1, the valency of each node is calculated and stored in the first column of a cost matrix ; zeros are stored in the first column of a corresponding decision or control matrix. Then starting with node 1 of stage 2, the valency af every ordering sequence consting of a node at stage 1 and node 1 of stage 2 is evaluated. The ordering sequence with minimum valency is kept by storing the valency in combined elimination of row 1 column 2 of the cost matrix.

At the corresponding location of the decision matrix a control value is stored which traces backwards from node 1 to the selected node at stage 1. This process is repeated for the all the nodes in stage 2. Similarly the process described just above is repeated for 3,4,..., (n - 1), and n stages. In the n - th stage the smallest value in the cost matrix is obtained at the j - th node, the j - th node is the last node in optimal ordering sequence. The other nodes in the optimal ordering can be obtained recursively with the help of decision matrix.

7.4 An Illustrative Example

Consider the example as stated above. The non zero pattern of the matrix is shown in Table 7.4.1. At stage 1, the valency is stored in the first column of the cost matrix, Table 7.4.2 and the control u = o is stored in column 1 of the decision matrix, Table 7.4.3 . After completion of all the stages the cost matrix, the decision matrix and the sequence of nodes eliminated are shown in Tables 7.4.2, 7.4.3, 7.4.4. Non zero pattern of the unordered and ordered buses are shown in Table 7.4.5 and 7.4.6 respectively.

Reordering of buses

After the optimal ordering of the buses is known the bus admittance matrix is reordered and the nodes are also reordered.

TABLE 7.4.1

1 - 0 - PATTERN OF BUS ADMITTANCE MATRIX OF IEEE 14 BUS SYSTEM

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	1	0	0	1	0	0	0	0	0	0	0	0	0
2	1	1	1	1	1	0	0	0	0	0	0	0	0	0
3	0	1	1	1	0	0	0	0	0	0	0	0	0	0
4	0	1	1	1	1	0	1	0	1	0	0	0	0	0
5	1	1	0	1	1	1	0	0	0	0	0	0	0	0
6	0	0	0	0	1	1	0	0	0	0	1	1	1	0
7	0	0	0	1	0	0	1	1	1	0	0	0	0	0
8	0	0	0	0	0	0	1	1	0	0	0	0	0	0
9	0	0	0	1	0	0	1	0	1	1	0	0	0	1
10	0	0	0	0	0	0	0	0	1	1	1	0	0	0
11	0	0	0	0	0	1	0	0	0	1	1	0	0	0
12	0	0	0	0	0	1	0	0	0	0	0	1	1	0
13	0	0	0	0	0	1	0	0	0	0	0	1	1	1
14		0	0	0	0	0	0	0	1	0	0	0	1	1

TABLE 7.4.2

DIFFERENT STAGES OF COST MATRIX

Stages/ Nodes	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	0	0	0	x	x	x	x	x	x	x	x	x	x
2	3	1	0	0	0	0	1	x	x	x	x	x	x	x
3	0	0	0	0	0	1	x	x	x	x	x	x	x	x
4	7	4	4	2	2	1	1	2	3	x	x	x	x	x
5	4	2	2	1	1	1	1	2	3	3	x	x	x	x
6	5	3	3	3	3	3	3	4	5	5	4	4	x	x
7	2	0	0	0	0	1	x	x	x	x	x	x	x	x
8	0	0	0	0	1	x	x	x	x	x	x	x	x	x
9	5	5	3	3	3	3	2	4	5	5	4	4	4	x
10	1	1	1	1	1	1	1	2	3	4	4	4	4	4
11	1	1	1	1	1	1	1	2	3	4	4	4	4	4
12	0	0	0	0	0	0	1	x	x	x	x	x	x	x
13	2	1	1	1	1	1	1	2	x	x	x	x	x	x
14	1	1	1	1	1	1	1	2	3	2	4	x	x	x

TABLE 7.4.2

DIFFERENT STAGES OF DECISION MATRIX

Stages/ Nodes	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	-2	-6	-2	x	x	x	x	x	x	x	x	x	x
2	0	1	1	-6	1	-1	-1	x	x	x	x	x	x	x
3	0	2	-4	2	-9	-10	x	x	x	x	x	x	x	x
4	0	1	3	2	3	2	2	2	-1	x	x	x	x	x
5	0	4	4	3	3	3	3	3	1	-9	x	x	x	x
6	0	-6	-6	-6	-6	3	4	4	2	2	1	-3	x	x
7	0	-1	-1	-1	5	2	x	x	x	x	x	x	x	x
8	0	7	7	6	3	x	x	x	x	x	x	x	x	x
9	0	8	8	8	8	7	7	7	5	5	4	3	-1	x
10	0	9	9	9	9	8	8	8	6	6	5	4	1	x
11	0	10	10	10	10	9	9	9	7	7	6	6	2	-1
12	0	11	11	11	11	10	8	x	x	x	x	x	x	1
13	0	1	1	1	1	10	11	9	x	x	x	x	x	x
14	0	13	13	13	13	12	12	12	10	10	4	x	x	x

TABLE 7.4.4

DIFFERENT STAGES OF NODE ELIMINATION

1	2	3	4	5
1	3 - 1	8-7-1	7-7-3-1	x
2	1 - 2	3-1-2	3-1-8-2	8-7-3-1-2
3	1 - 3	8-7-3	8-7-1-3	8-7-1-12-3
4	3 - 4	3-1-4	3-1-2-4	8-7-3-1-4
5	1 - 5	3-1-5	3-1-2-6	3-1-8-2-5
6	12- 6	1-12-6	3-1-12-6	8-7-1-12-6
7	8 - 7	1-8-7	3-1-8-7	3-1-8-2-7
8	1 - 8	3-1-8	3-1-2-8	3-1-2-5-8
9	1 - 9	8-7-9	8-7-1-9	8-7-3-1-9
10	1 - 10	3-1-10	8-7-1-10	8-7-3-1-10
11	1 - 11	3-1-11	8-7-1-11	8-7-3-1-11
12	1 - 12	3-1-12	8-7-1-12	8-3-3-1-12
13	12-13	1-12-13	3-1-12-13	8-7-1-12-13
14	1 - 14	3-1-14	8-7-1-14	8-7-3-1-14

TABLE 7.4.4 (Continued)

	6	7	8
1	x	x	x
2	8-7-1-12-3-2	8-7-1-12-13-3-2	x
3	8-7-1-12-13-3	x	x
4	8-7-3-1-2-4	8-7-1-12-3-2-4	8-7-1-12-13-3-2-4
5	8-7-3-1-2-5	8-7-1-12-3-2-5	8-7-1-12-13-3-2-5
6	8-7-1-12-3-6	8-7-1-12-3-2-6	8-7-1-12-13-3-2-6
7	3-1-8-2-5-7	x	x
8	x	x	x
9	8-7-3-1-2-9	8-7-1-12-3-2-9	8-7-1-12-13-3-2-9
10	8-7-3-1-2-10	8-7-1-12-3-2-10	8-7-1-12-13-3-2-10
11	8-7-3-1-2-11	8-7-1-12-3-2-11	8-7-1-12-13-3-2-11
12	8-7-3-1-2-12	8-7-3-1-2-4-12	x
13	8-7-1-12-3-13	8-7-1-12-3-2-13	8-7-1-12-3-2-4-13
14	8-7-3-1-2-14	8-7-1-12-3-2-14	8-7-1-12-13-3-2-14

TABLE 7.4.4 (Continued)

	9	10
1	x	x
2	x	x
3	x	x
4	8-7-1-12-13-3-2-5-4	x
5	8-7-1-12-13-3-2-4-5	8-7-1-12-13-3-2-4-14-5
6	8-7-1-12-13-3-2-4-6	8-7-1-12-13-3-2-5-4-6
7	x	x
8	x	x
9	8-7-1-12-13-3-2-4-9	8-7-1-12-13-3-2-5-4-9
10	8-7-1-12-13-3-2-4-10	8-7-1-12-13-3-2-5-4-10
11	8-7-1-12-13-3-2-4-11	8-7-1-12-13-3-2-5-4-11
12	x	x
13	x	x
14	8-7-1-12-13-3-2-4-14	8-7-1-12-13-3-2-5-4-14

TABLE 7.4.4 (Continued)

	11	12
1	x	x
2	x	x
3	x	x
4	x	x
5	x	x
6	8-7-1-12-13-3-2-4-14-5-6	8-7-1-12-13-3-2-4-14-5-2-6
7	x	x
8	x	x
9	8-7-1-12-13-3-2-4-14-5-9	8-7-1-12-13-3-2-4-14-5-6-9
10	8-7-1-12-13-3-2-4-14-5-10	8-7-1-12-13-3-2-4-14-5-6-10
11	8-7-1-12-13-3-2-4-14-5-11	8-7-1-12-13-3-2-4-14-5-6-11
12	x	x
13	x	x
14	8-7-1-12-13-3-2-5-4-10-14	x

TABLE 7.4.4 (Continued)

	13	14
1	x	x
2	x	x
3	x	x
4	x	x
5	x	x
6	x	x
7	x	x
8	x	x
9	8-7-1-12-13-3-2-4-14-5-6-10-9	x
10	8-7-1-12-13-3-2-4-14-5-6-9-10	8-7-1-12-13-3-2-4-14-5-6-9-11-10
11	8-7-1-12-13-3-2-4-14-5-6-9-11	8-7-1-12-13-3-2-4-14-5-6-9-10-11
12	x	x
13	x	x
14	x	x

TABLE 7.4.5

UNORDERED BUSES

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	x	x			x									
2	x	x		x	x									
3		x	x	x										
4		x	x	x	x		x		x					
5	x	x		x	x	x								
6					x	x					x	x	x	
7				x			x	x	x					
8							x	x						
9				x			x		x	x				x
10									x	x	x			
11						x				x	x			
12						x						x	x	
13						x						x	x	x
14									x				x	x

TABLE 7.4.6

OPTIMALLY ORDERED BUSES

	8	7	1	12	13	3	2	4	14	5	6	9	10	11
8	x	x												
7	x	x						x				x		
1			x			x	x			x				
12				x	x						x			
13				x	x				x		x			
3						x	x	x						
2						x	x	x		x				
4		x				x	x	x		x		x		
14					x				x			x		
5			x					x		x	x			
6				x	x					x	x			x
9								x	x			x	x	
10												x	x	x
11													x	x

7.5 Solution Techniques

A bus with largest generation is assumed as a slack bus or any other bus as specified. The solution of the load flow problem is initialised assuming voltages for all buses except the slack bus. At the slack bus voltage is specified. The currents are calculated for all buses except the slack bus s from the bus loading equation

$$I_p = \frac{(P_{Gp} - P_{Lp}) - j(Q_{Gp} - Q_{Lp})}{E_p^*}$$

$$p = 1, 2, \ldots, n \qquad p \neq s \tag{7.5.1}$$

and

$$\begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_p \\ I_n \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1n} \\ Y_{21} & Y_{22} & \cdots & Y_{2n} \\ \vdots & & & \\ Y_{p1} & Y_{p2} & \cdots & Y_{pn} \end{bmatrix} \begin{bmatrix} E_1 \\ E_2 \\ E_p \\ \vdots \\ E_n \end{bmatrix} \tag{7.5.2}$$

It follows from equation (7.5.2)

$$Y_{pp} E_p + \sum_{\substack{q=1 \\ q \neq p}}^{n} Y_{pq} E_q = I_p \tag{7.5.3}$$

Hence

$$E_p = \frac{1}{Y_{pp}} \left[I_p - \sum_{\substack{q=1 \\ q \neq p}}^{n} Y_{pq} E_q \right] \quad p = 1, 2, \ldots, n \quad p \neq s \qquad (7.5.4)$$

The equation (7.5.4) involves only the bus voltages as variables. The corresponding voltage equations are non-linear in form and require iterative techniques for their solution.

Let $L_p = \frac{1}{Y_{pp}}$, the equation (7.5.4) can be written as

$$E_p = L_p \left[\frac{(P_{Gp} - P_{Lp}) - j\,(Q_{Gp} - Q_{Lp})}{E_p^*} - \sum_{\substack{q=1 \\ q \neq p}}^{n} Y_{pq} E_q \right]$$

... (7.5.5)

$$E_p = \frac{KL_p}{E_p^*} - \sum_{\substack{q=1 \\ q \neq p}}^{n} YL_{pq} E_q \quad p = 1, 2, \ldots, n \quad p \neq s \qquad (7.5.6)$$

where

$$KL_p = \left[(P_{Gp} - P_{Lp}) - j\,(Q_{Gp} - Q_{Lp}) \right] L_p$$

is known as the bus parameters

and

$$KL_{pq} = L_p \sum_{\substack{q=1 \\ q \neq p}}^{n} Y_{pq} \text{ is the line parameters.}$$

If the bus p is a voltage controlled bus KL_p is to be recomputed for every iteration.

7.6 Voltage Controlled Buses

In Gauss-Seidel method with Y_{BUS} the reactive power at a voltage controlled bus p must be calculated before calculating the voltage at that bus. Separating the real and imaginary parts of the bus power equation

$$(P_{Gp} - P_{Lp}) - j(Q_{Gp} - Q_{Lp}) = E_p^* \sum_{q=1}^{n} Y_{pq} E_q \text{, the reactive}$$

bus power

$$Q_{GP} - Q_{LP} = e_p^2 B_{pp} + f_p^2 B_{pp} + \sum_{\substack{q=1 \\ p \neq q}}^{n} \Big\{ f_p (e_q G_{pq} + f_q B_{pq})$$

$$- e_p (f_q G_{pq} - e_q B_{pq}) \Big\} \qquad (7.6.1)$$

where

$Y_{pq} = G_{pq} - jB_{pq}$ and

$$e_{pp}^2 + f_p^2 = \left\{ \left| E_p \text{ (scheduled)} \right| \right\}^2 \tag{7.6.2}$$

In order to calculate the reactive bus power needed to give the scheduled bus voltage, the equation (7.6.1) must be satisfied. The present estimate of e_p^k and f_p^k must be adjusted accordingly. The phase angle of the estimated bus voltage is

$\delta_p^k = \arctan \dfrac{f_p}{e_p^k}$. The adjusted estimate of e_p^k and f_p

are e_p^k (new) = e_p scheduled cos δ_p^k

f_p^k (new) = E_p scheduled sin δ_p^k

where superfix k is the iteration count in Gauss-Seidel method. Substituting e_p^k (new) and f_p^k (new) in equation (7.6.1) the reactive power Q_p^k is obtained and is used with E_p^k (new) for calculating the new voltage estimate E_p^{k+1}.

If the calculated Q_p^k exceeds the $Q_{p\,(max)}$ then $Q_{p\,(max)}$ is considered as the reactive power of the bus ; if Q_p^k is less than $Q_{p\,(min)}$ then $Q_{p\,(min)}$ is considered as the reactive power of the bus, and the bus is considered as P - Q. The bus parameter KL_p is recomputed.

Then the equation (7.5.6) is solved by Gauss-Seidel iterative method. In this method the new calculated E_p^{k+1} immediately replaces E_p^k and is used in solution of the subsequent equations.

7.7 Line Flow Equations

After the voltages of the buses are converged to a solution iteratively, the line flows are determined as

$$P_{pq} - jQ_{pq} = E_p^*(E_p - E_q)\, y_{pq} + E_p^* E_p\, y'_{pq}/2 \qquad (7.7.1)$$

where y_{pq} is the line admittance

y_{pq}' is the total line charging admittance.

The reversed power flow is

$$P_{qp} - jQ_{qp} = E_q^*(E_q - E_p)\, y_{pq} + E_q^* E_q\, y'_{pq}/2$$

The slack bus power can be determined by summing the flows on the lines terminating at the slack bus.

Tolerance test was made to achieve convergence as

$$E_p^{k+1} - E_p^k = \Delta E_p^{k+1}$$

The calculation will be terminated when ΔE_p^{k+1} is a predetermined small value ζ.

To achieve quicker convergence the voltage is accelerated as

$$E_p^{k+1} = E_p^{k} + \alpha \Delta E_p^{k+1}$$

where α is a predetermined value empirically obtained in the neighbourhood of 1.6.

A complete flow chart for Gauss-Seidel Load Flow Analysis with optimally ordered nodes is shown in Fig.7.7.1.

Another method of sub-optimal ordering is also included in the illustration. This technique states that

> "This scheme partly simulates the Gauss elimination process and requires that at each step of row - column elimination, the node with least number of off-diagonal terms be eliminated next. If more than one row - column meets this criterion, select any one."

The above scheme of sub-optimal ordering is known as Tinney's second scheme. Computation time for this scheme is less than that of optimatimal ordering by dynamic programming.

7.8 Illustration

IEEE 14 bus system is considered for the load flow analysis. Sub-optimal bus ordering according to Tinney's Second Scheme was obtained as 1,3,2,8,7,12,4,5,10,11,6,9, 13,14 and the valency of ordering was 5.

Optimal bus ordering according to Dynamic Programming algorithm was obtained as 8,7,1,12,13,3,2,4,14,5, 6,9,11,10 and the valency of ordering was 4. For a tolerance limit of 0.01 the Gauss-Seidel load flow calculation converged after 14 iteration [92]. Fig. 7.8.1 shows the IEEE 14 BUS system and the Table 7.8.1 shows the description of the IEEE 14 BUS system. The software developed in BASIC language is given the Appendix as A5.1, A5.2 and A5.3.

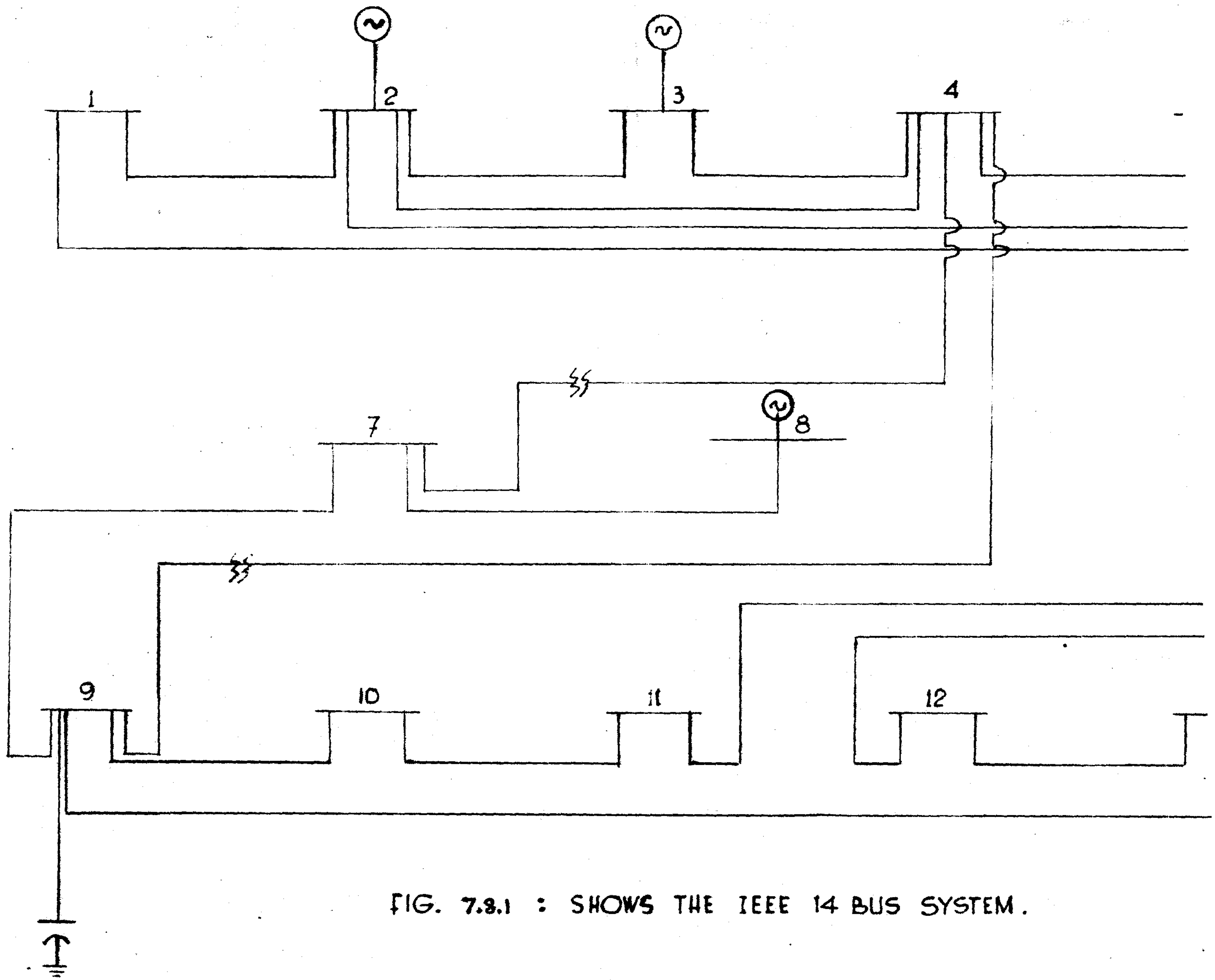


FIG. 7.8.1 : SHOWS THE IEEE 14 BUS SYSTEM.

TABLE 7.8.1

DESCRIPTION OF THE IEEE 14 BUS SYSTEM

BUS DATA

Bus No.	Generation		Load	
	Real MW	Reactive MVAR	Real MW	Reactive MVAR
1	232.4	-16.9	0.0	0.0
2	40.0	42.4	21.7	12.7
3	0.0	23.4	94.2	19.0
4	0.0	0.0	47.8	3.9
5	0.0	0.0	7.6	1.6
6	0.0	12.2	11.2	7.5
7	0.0	0.0	0.0	0.0
8	0.0	17.4	0.0	0.0
9	0.0	0.0	29.5	16.6
10	0.0	0.0	9.0	5.8
11	0.0	0.0	3.5	1.8
12	0.0	0.0	6.1	1.6
13	0.0	0.0	13.5	5.8
14	0.0	0.0	14.9	5.0

LINE DATA

Line No.	Between Buses	Line impedance		Half line charging susceptance per Unit
		R per Unit	X per Unit	
1	1-2	0.01938	0.05917	0.02640
2	2-3	0.04699	0.19797	0.02190
3	2-4	0.05811	0.17632	0.01870
4	1-5	0.05403	0.22304	0.02460
5	2-5	0.05695	0.17388	0.01700
6	3-4	0.06701	0.17103	0.01730
7	4-5	0.01335	0.04211	0.0064
8	5-6	0.0	0.25202	0.0
9	4-7	0.0	0.20912	0.0
10	7-8	0.0	0.17615	0.0
11	4-9	0.0	0.55618	0.0
12	7-9	0.0	0.11001	0.0
13	9-10	0.03181	0.08450	0.0
14	6-11	0.09498	0.19890	0.0
15	6-12	0.12291	0.25581	0.0
16	6-13	0.06625	0.13027	0.0
17	9-14	0.12711	0.27038	0.0
18	10-11	0.08205	0.19207	0.0
19	12-13	0.22092	0.19988	0.0
20	13-14	0.17093	0.34802	0.0

TRANSFORMER DATA

Transformer	Between Buses	Tap Setting
1	4 - 7	0.978
2	4 - 9	0.969
3	5 - 6	0.932

SHUNT CAPACITOR DATA

Bus Number	Susceptance per Unit
9	0.190

REGULATED BUS DATA

Bus Number	Voltage magnitude per Unit	Reactive power limits	
		Minimum MVAR	Maximum MVAR
2	1.045	- 40.0	50.0
3	1.010	0.0	40.0
6	1.070	- 6.0	24.0
8	1.090	- 6.0	24.0

CHAPTER VIII

CONCLUSION AND SCOPE OF FURTHER WORK

CHAPTER VIII

CONCLUSION AND SCOPE OF FURTHER WORK

The present work is a report of our experiments with the application of different methods of applied cybernetics in electrical power industry.

In chapter III we have developed the hourly flow simulation technique with the cybernetical method of recursive least square instrument variable algorithm with on-line parameter tracking adaptiveness. The effectiveness of the developed technique has been demonstrated with the field data observed at the different gauging stations of the hilly river Teesta in North Bengal.

The seemingly complex algorithms and high performance computational requirements have in the past limited the application of estimation techniques to aerospace technology and missile guidance system where there are high cost performance tradeoffs. Recently, LSI microprocessors have had a dramatic impact in the control of data processing area. LSI and VLSI technology have advanced so rapidly that the current high performance microprocessors have the computational capabilities to implement estimation and control algorithms. In this part sufficient details of the recursive least square instrument variable algorithms are given with associated parameter tracking adaptiveness in the instrument variables including real time experiments to allow one to assess the requirements and capabilities of microprocessor based estimation algorithms for real time monitoring of hourly river flows and for on-line controlling of the hydraulic structures of the large size run-of-the river hydroelectric plant. The ability of the algorithm has not yet been tested on real world problem.

In chapter IV a mathematical description of annual electrical energy consumption in India has been developed with population, gross national product, gross domestic saving and gross domestic capital formation as exogenous variables in the form of a polynomial of optimum complexity with the help of a learning identification technique known as the multilayer group method of data handling algorithm. The model is found to simulate adequately the effects of interactions of different techno-economic parameters on annual electrical energy consumption.

The modelled data are found to be in close agreement with the observed ones. From the results of the illustration it is observed that multilayer group method of data handling technique is capable of ascribing structure on processed data by providing missing information to systems with uncertainties operating in a complex behavioural environment. Other techno-economic and socio-economic parameters e.g. national productivity index and indices showing the quality of life of the people can be incorporated in the developed model.

In chapter V the model of annual installed plant capacity of electrical energy of India has been obtained in the form of polynomial of optimum complexity by computer aided self-organisation of mathematical models. Desired rate of growth of annual installed plant capacity and annual energy consumption have been assumed. On the basis of the growth rates the polynomial models of optimum complexity have been obtained for annual installed plant capacity and energy consumption. A model for plant annual load factor has also been obtained. The models can be used as handy tools for planners of power industry.

The polynomial models presented in this work are found to simulate adequately the variations in the observed data. The models are handy tools for planners of power industry. A close look into the model for plant annual load factor will bring out many important features of Indian power industry. This discussion has been kept outside the purview of our present research interest.

In chapter VI we have developed State Estimation technique which provides a powerful tool to obtain a data base for on-line supervision and control of power system. In this work recursive type least square technique with parameter tracking algorithm has been used to obtain the estimation of the power system state variables. Incorporation of the parameter tracking algorithm makes the states estimator amenable to on-line operation. The estimates of the states will help in selecting on-line contingency plan. An illustration is given to show the application of methods and the software developed.

The tracking state estimator developed above has been illustrated using only one snapshot of measurements. It can handle the slowly varying characteristics of the power system and thus is capable of functioning properly under discretetive on-line operating conditions.

This part presented a new type of state estimation algorithm. This algorithm is suitable for both estimating on real time basis the static states and quasi-dynamic states for on-line operation. The algorithm has fast converging characteristic since it does not involve matrix inversion. The approximations used in this method were shown not to have any negative impact on the discriminatory ability of the data detection and identification process. Since the sequence of operations, during iterations for static states as well as for recursive estimation of quasi-dynamic states for on-line operation, is identical for both $e(.)$ and $f(.)$ systems, the modularization of the algorithm and the use of structured programming are possible. However, it is believed that much effort is still needed for real life application of the developed algorithm.

In chapter VII Gauss-Seidel load flow with optimally ordered nodes by dynamic programming algorithm of applied cybernetics has been described. Algorithm has been illustrated on IEEE 14 Bus System. It has been observed that the computational efficiency is improved with optimal ordering by dynamic programming algorithm.

The Gauss-Seidel load flow with dynamic programming algorithms have improved the computational efficiency of the load flow analysis. The convergence of load flow calculation is quick. SPECTRUM/ -3 microprocessor of DCM data products was used for calculation. Because of the memory limitations higher order bus system could not be tested. It is hoped that number of iterations will be drastically reduced if other fast converging load flow methods are used.

Investigation, carried out in this work, has helped in developing necessary softwares for off line planning and on-line control of electrical power industry. It also shows that cybernetical methods are powerful tools for analysis the different aspects of electrical power systems.

A P P E N D I X

A P P E N D I X

A1.1 RATIONALISATION OF INPUT DATA

```
COMPILE NRDATA $R
NYBASIC COMPILER V2.0
    REM RATIONALISATION OF DATA
    LPRINTER WIDTH 80
    INPUT "N=" ; N
    INPUT "K ="; K
    DIM X(N,K)
    INPUT "DATA FILE =" ; D$
    OPEN D$ AS 1
    FOR J = 1 TO K
    FOR I = 1 TO N
    READ #1 ; X( I,J )
    NEXT  I
    NEXT  J
    CLOSE 1
    FOR  J = 1  TO  K
    MIN = X( 1,J )
    FOR  I = 1  TO  N
    IF MIN < X( I,J ) THEN 10
    MIN = X( I,J )
    10 NEXT  I
    MAX = X( 1,J )
    FOR I = 1  TO  N
    IF MAX > X( I,J )  THEN 20
    MAX = X( I,J )
    20 NEXT  I
    FOR I = 1  TO  N
    X( I,J ) = ( X( I,J ) - MIN )/( MAX - MIN )
    NEXT  I
    NEXT  J
```

```
29 : INPUT "FILE NAME =" ; A$
30 : CREATE A$ AS 2
31 : FOR J = 1 TO K
32 : FOR I = 1 TO N
33 : PRINT # 2 ; X( I,J )
34 : NEXT I
35 : NEXT J
36 : CLOSE 2
37 : PRINT
38 : STOP
39 : END
NO ERRORS DETECTED
CONSTANT AREA :          8
CODE SIZE :            603
DATA STMT  AREA :        0
VARIABLE AREA  :        78
A
```

A1.2 DETERMINATION OF CORRELATION CO-EFFICIENTS

```
COMPILE THV2 $H
HIBASIC COMPILER V2.0
   1: LPRINTER WIDTH 80
   2: REM CORRELATION
   3: INPUT "DATA, VARIABLES, SHIFT = "; N,K,M
   4: DIM Y( N,K ), YM( K ), COR ( M+1, K )
   5: INPUT "DATA FILE ="; D$
   6: OPEN D$ AS 1
   7: FOR J = 1 TO K
   8: FOR I = 1 TO N
   9: READ # 1 ; Y ( I,J )
  10: NEXT I
  11: NEXT J
  12: CLOSE 1
  13: FOR J = 1 TO K
  14: SUM = 0
  15: FOR I = 1 TO N
  16: SUM = SUM + Y ( I,J )
  17: NEXT I
  18: YM( J) = SUM/N
  19: NEXT J
  20: M1 = M + 1
  21: FOR K1 = 1 TO M1
  22: SUMA = 0
  23: NI = N - K1 + 1
  24: FOR I = 1 TO NI
  25: SUMA = SUMA+ (Y( I,1) - YM(1))*(Y(I,1) - YM(1))
  26: NEXT I
  27: FOR J = 1 TO K
  28: SUMC = 0
  29: SUMD = 0
  30: FOR I = 1 TO NI
```

```
31: II = I + K1 - 1
32: SUMC =SUMC +(Y(I,1) -YM(1))*(Y(II,J) - YM(J))
33: SUMD = SUMD + (Y ( II,J ) - YM(J))*(Y(II,J) -YM(J))
34: NEXT  I
35: COR ( K1,J ) = SUMC/SQR ( SUMA*SUMD )
36: NEXT  J
37: NEXT  K1
38: PRINT TAB ( 20 ); "CORRELATION"; PRINT
39: FOR  K1 = 1  TO M1
40: K11 = K1 - 1
41: PRINT TAB ( 10 );"TIME INSTANT ("; K11 ;")" ; PRINT
42: FOR  J = 1  TO  K
43: PRINT COR ( K1, J ),
44: NEXT  J
45: PRINT
46: NEXT  K1
47: PRINT
48: STOP
49: END
NO ERRORS DETECTED
CONSTANT AREA :            8
CODE SIZE  :             907
DATA STMT AREA  :          0
VARIABLE AREA    :       1AA
A
```

A1.3 DETERMINATION OF INPUT ARGUMENTS

```
COMPILE HISTORY $E
HIBASIC COMPILER V2.0
1: REM CALCULATION OF PREHISTORY
2: LPRINTER WIDTH 80
3: INPUT "N,K,M1, K1 ="; N,K,M1,K1
4: DIM X(N,K), Y(M1,K1)
5: INPUT "FILE NAME =" ; B$
6: OPEN B$ AS 1
7: FOR J = 1 TO K
8: FOR I = 1 TO N
9: READ # 1 ; X( I,J )
10: NEXT I.
11: NEXT J
12: CLOSE 1
13: FOR I = 1 TO M1
14: Y( I,1 ) = X( I + 4, 1 )
15: Y( I,2 ) = X(I + 3,1) : Y( I,3 ) = X( I + 2, 1 )
16: Y( I,4 ) = X(I + 1,1) : Y( I,5 ) = X( I,1 ) : Y( I,6 ) = I
17: NEXT I
18: INPUT "FILE NAME =" ; A$
19: CREATE A$ AS 2
20: FOR J = 1 TO K1
21: FOR I = 1 TO M1
22: PRINT # 2 ; Y ( I, J )
23: NEXT I
24: NEXT J
25: CLOSE 2
26: PRINT
27: STOP
28: END
NO ERRORS DETECTED
CONSTANT AREA :          8
CODE SIZE  :           543
DATA STMT AREA :         0
VARIABLE AREA  :        80
A
```

A1.4 ON LINE RECURSIVE INSTRUMENT VARIABLES ALGORITHM

```
COMPILE EASIV $E
MBASIC COMPILER V5.0
  1: PRINT TAB (10); "ON LINE HOURLY ESTIMATED FLOWS"
  2: PRINT TAB (10); "AT CORONATION BRIDGE POINT OF THE TEESTA RIVER"
  3: PRINT
  4: LPRINTER WIDTH 80
  5: REM LEAST SQUARE, LSQ. RECURSIVE & LSQ.SVR.IV.ALGORITHMS
  6: INPUT "NO. OF DATA, NO. OF PREHISTORY, VARIABLE + 1 ="; NL,NP,N
  7: MP = ( N - 1 ) * ( NP + 1 ) + NP
  8: DIM YD (NL,N),SV1(MP), P(MP,MP), G1(MP), SV(MP)
  9: DIM PE ( NL + 2), YA( NL,4 ), P1(MP), P2 (MP )
 10: DIM YE( NL ), GE( MP ), SV2 ( MP )
 11: INPUT "FILE NAME : CORON. INP"; B$
 12: OPEN B$ AS 1
 13: FOR J = 1 TO N : FOR I = 1 TO NL
 14: READ # 1 ; YD( I,J ): NEXT I, J : CLOSE 1
 15: INPUT " INITIAL PREHISTORY, NN = " ; NN
 16: FOR N = NN TO NP : MS = N + ( N - 1 )*( N + 1 )
 17: INPUT "KASHYAP ALGM. REQD. ? YES = 1 NO = 0 ="; KAS
 18: IF KAS = 0 THEN 500
 19: INPUT "NO. OF ERROR TERMS NV =" ; NV
 20: NT = N + NV +( N - 1 )*( N + 1 )
 21: DIM S(NT,NT),A(NT), Z(NT),DA(NT),ED(NT),EG(NT)
 22: DIM EK ( NT )
 23: FOR I = 1 TO NT : A(I) = 0: Z(I)= 0: FOR J = 1 TO NT
 24: S(I,J) = 0:S( I,I ) = 1 : NEXT J, I
 25: K11 = 1 : K22 = NL
 26: PRINT "PARAMETERS TRACKING ALGMS." : PRINT
 27: PRINT "SERIAL NO., OBSERVED, MODELLED & ERROR" : PRINT
 28: FOR KO = K11 TO K22: SUMC = 1 : FOR I = 1 TO NT
 29: SUMA = 0: SUMB = 0 : FOR KD = 1 TO NT
 30: SUMA = SUMA+ S(I,KD)*Z(KD): SUMB =SUMB+ Z(KD)*S(KD,I)
```

```
NEXT KD: DA(I) = SUMA: DB(I)= SUMB: NEXT I: FOR I=1 TO NT
SUMC= SUMC + DB(I)*Z(I): NEXT I
FOR I = 1 TO NT: DC(I) = DA(I)/SUMC: FOR J = 1 TO NT
S( I,J ) = S( I,J ) - DC( I )* DB( J ): NEXT J,I
SUMA= YD(KO,1): FOR I = 1 TO NT: SUMA = SUMA -A(I)* Z(I)
NEXT I: FE( KO ) = SUMA: YE( KO )= YD( KO,1) - FE( KO )
PRINT KO, YD( KO,1 ), YE( KO ), FE( KO ) : PRINT
IF KO = KE2 THEN 600
FOR I=1 TO NT: SUMD = 0: FOR J= 1 TO NT
SUMD = SUMD + S( I,J )* Z( J )
DK( I ) = SUMD : NEXT J, I
FOR I = 1 TO NT: A( I ) = A( I )+ FE( KO )* DK( I ): NEXT I
REM GENERATION OF Z - VECTOR
FOR I = 1 TO M: KK1 = KO - I + 1
IF KK1 <= 0 THEN 5
Z(I) = YD( KK1,1 ) : GO TO 4
5 Z( I ) = 0
4 NEXT I
FOR I = 1 TO NV: KK1 = KO - I + 1 : II = M + I
IF KK1 <= 0 THEN 6
Z( II ) = FE( KK1 )
GO TO 7
6 Z( II ) = 0
7 NEXT I
FOR J = 2 TO N: FOR I = 1 TO N + 1: KK1 = KO - I
IF KK1 = 0 THEN 8
II=M+NV+( J-2)*( N+1 )+ I: Z( II ) = YD( KK1,J )
GO TO 9
8 Z( II ) = 0
9 NEXT I
NEXT J
NEXT KO
600 INPUT "PARAMETRIC ALGM. ERROR FILE ="; L2$
CREATE L2$ AS 4: FOR KO = 1 TO K1 : PRINT # 4; FE( KO )
```

```
65: NEXT K0: CLOSE 1
66: SUMA=0:SUMB=0:SUMC=0:FOR I=1 TO N1:SUMA = SUMA +FE(I)*FE(I)
67: SUMB = SUMB + YD(I,1)* YD(I,1) : SUMC = SUMC + FE (I)
68: NEXT I : SUMB = SUMA/SUMB : SUMA = SUMA / ( N1 - N )
69: PRINT " INTEGRAL SQUARE ERROR & MEAN ERROR =" ; SUMB, SUMA
70: 500 INPUT " LSQ SKIP = 1 OTHERWISE = 0 =" ; SKIP
71: IF SKIP = 1 THEN 105
72: PRINT "LEAST SQUARE METHOD" : PRINT
73: FOR I = 1 TO N5 : SV(I) = 0 : SV1(I) = 0
74: NEXT I : FOR I = 1 TO N5 : FOR J = 1 TO N5
75: P(I,J) = 0 : NEXT J, I : N2 = N1 - N
76: INPUT " HOW MANY DATA SKIPPED FOR LSQ COMP =" ; LSQ
77: FOR K = 1 TO N2 - LSQ : KK = K
78: GOSUB 100
79: GO TO 11
80: 100 REM GENERATION OF S - VECTOR
81: FOR I = 1 TO N : II = N + KK - I
82: SV(I) = YD(II,1) : SV2(I) = UD(II) : NEXT I
83: FOR J = 2 TO N : FOR I = 1 TO N [illegible]
84: SV( IJ ) = YD(II, J): SV2(IJ) = YD(II,J): NEXT I, J: RETURN
85: 11 FOR I = 1 TO N5 : FOR J = 1 TO N5
86: P( I,J ) = P ( I,J ) + SV( I ) * SV( J ) : NEXT J,I
87: FOR I = 1 TO N5 : KK = N + K
88: SVB(I) = SV1(I) + YD( KK,1 )* SV ( I )
89: NEXT I, K : GOSUB 200
90: INPUT " MATRIX PIJ = "; X$
91: CREATE X$ AS 0
92: FOR I = 1 TO N5 : FOR J = 1 TO N5
93: PRINT # 0 ; P( I,J ) : NEXT J, I : CLOSE 0
94: GO TO 21
95: 200 REM MATRIX INVERSION
96: FOR L = 1 TO N5 : X = 1 / P( L,L ) : P( L,L ) = 1
97: FOR I = 1 TO N5: P(I,L) = P(I,L) * X
98: NEXT I : FOR J = 1 TO N5
```

```
 99: IF J = L THEN 20
100: X = P ( L,J ) : P ( L,J ) = 0 : FOR I = 1 TO N3
101: P( I,J ) = P( I,J ) - P( I,L )* X : NEXT I
102: 20 NEXT J: NEXT L : RETURN
103: 21 FOR I =1 TO N3 : SUM = 0: FOR J = 1 TO N3
104: SUM = SUM + P ( I,J )* SVR( J ): NEXT J
105: C1(I) = SUM: NEXT I : PRINT "LSQ. COEFF."
106: FOR I = 1 TO N3 : PRINT C1 (I), : NEXT I : PRINT
107: INPUT "LSQ COEF FILE NAME =" ; CF $
108: CREATE CF$ AS 3
109: FOR I = 1 TO N3 : PRINT # 3 ; C1(I) : NEXT I : CLOSE 3
110: PRINT " LEAST SQUARE ESTIMATION" : PRINT
111: PRINT " SERIAL NO., OBSERVED, MODELLED & ERROR ": PRINT
112: FOR K = 1 TO N2
113: K1 = K : GOSUB 100
114: SUM = 0 : FOR I = 1 TO N3
115: SUM = SUM + EV (I)* C1(I): NEXT I
116: YE = SUM: NK = N + K: ERR = YD( NK, 1 ) - YE
117: YA(K,1) = NK: YA( K,2 ) = YD( NK,1 ): YA( K,3 ) = YE
118: YA( K,4 )= ERR:PRINT YA(K,1), YA(K,2),YA( K,3), YA(K,4): NEXT K
119: INPUT " LSQ ERROR FILE NAME "; L1 $
120: CREATE L1 $ AS 2
121: FOR K = 1 TO N2: PRINT # 2 ; YA(K,4) : NEXT K : CLOSE 2
122: SUMC = 0: SUMB = 0 : SUMA = 0
123: FOR K = 1 TO N 2 : SUMA = SUMA + YA( K,4 )
124: SUMB = SUMB + YA( K,4 )* YA( K,4 )
125: SUMC = SUMC + YA( K,2 )*YA( K,2 ): NEXT K
126: SUMB = (SUMB / SUMC ): SUMA = SUMA / N 2
127: PRINT "INTEGRAL SQR.ERROR & MEAN ERROR ="; SUMB, SUMA
128: PRINT : PRINT : PRINT : PRINT
129: GO TO 33
130: 205 INPUT "LSQ COEF FILE NAME ="; CFF $
131: OPEN CFF $ AS 3 : FOR I = 1 TO N3
132: READ # 3 ; C1(I) : NEXT I: CLOSE 3
```

```
133: INPUT "  PIJ FILE =" ; X$
134: OPEN X$ AS 9
135: FOR I = 1  TO N3 : FOR J = 1 TO  N3
136: READ # 9 ; P(I,J) : NEXT J,I : CLOSE  9
137: INPUT "LOQE SKIP = 1 OTHERWISE  0 = "; SKIP
138: IF  SKIP = 1  THEN 205
139: 35 PRINT " LEAST SQUARE RECURSIVE ALGORITHMS"
140: PRINT: PRINT " SERIAL NO., OBSERVED, MODELLED & ERROR": PRINT
141: FOR  I=1 TO N 3: SV(I) = 0:SVR(I) = 0:GI(I) =G1(I): NEXT  I
142: NA = NE _ N - 2
143: FOR  K = 1  TO  NA
144: K1 = K: GOSUB  100
145: FOR I=1 TO  N 3 : SUM = 0 : FOR  J = 1  TO  N 3
146: SUM = SUM + P(I, J)* SV(I) : NEXT  J
147: P1(I)= SUM : NEXT I : D = 1.0
148: FOR I = 1  TO N 3 : D = D + SV(I)* P1(I)
149: NEXT I : FOR J = 1  TO N 3 : SUM = 0
150: FOR I = 1  TO  N 3 : SUM = SUM + SV(I)* P(I,J)
151: NEXT I : P 2(J) = SUM : NEXT  J
152: D = 1.0 / D: FOR I= 1  TO  N 3 : FOR  J = 1  TO  N 3
153: P(I,J) = P(I,J) - P1( I)* P2(J)*D
154: NEXT J,I : K1 =  K + 1 : GOSUB  100
155: FOR I=1  TO  N 3: SUM = 0: FOR  J = 1 TO  N 3
156: SUM = SUM + P(I,J)* SV(J) : NEXT  J
157: P1(I) = SUM : NEXT I : D = 1.0 : FOR  I = 1  TO  N 3
158:  D = D + SV(I) * P1(I) :  NEXT  I
159: D = 1 / D : E1 = 0.0 : FOR  I = 1 TO  N 3
160: E1 = E1+ SV(I)* GI(I) : NEXT I : NK 2 = N + K + 1
161: Y = YD (NK 2,1) - E1 : Y = Y*D : FOR I = 1  TO  N  3
162: GI(I) = GI(I) + P1(I) * Y : Next  I
163: NK 2 = NK 2 + 1 :  K1 = K1 +  1 : GOSUB 100
164: YE = 0: FOR I = 1 TO  N3  : YE = YE + SV(I)* GI(I)
165: NEXT I : ERR = YD( NK 2, 1 ) - YE
166: YA(K,1) = NK 2: YA(K,2) = YD(NK 2,1): YA( K,3 ) = YE
167: YA(K,4) = ERR: PRINT YA(K,1),YA(K,2),YA(K,3),YA(K,4) : PRINT
```

```
168: NEXT K
169: INPUT "LSQR ERROR FILE NAME =" ; L $
170: CREATE L $ AS 4 : FOR K = 1 TO N4:PRINT #4; YA(K,4):NEXT K:CLOSE 4
171: SUMA = 0 :   SUMB = 0 : SUMC = 0
172: FOR K = 1 TO  N4 : SUMA = SUMA + YA( K,4 )
173: SUMB = SUMB + YA( K,4 )* YA( K,4 )
174: SUMC = SUMC + YA( K,2 )* YA( K,2 ) : NEXT K
175: SUMA = SUMA / N4 : SUMB = ( SUMB / SUMC )
176: PRINT "LSQR INTGL. SQR ERROR & MEAN ERROR.="; SUMB, SUMA
177: INPUT "VIJ FILE NAME ="; XXX$
178: OPEN XXX$ AS 10 : FOR I = 1  TO  N3 : FOR J = 1  TO  N3
179: READ # 10 ; P(I,J) : NEXT  J, I : CLOSE  10
180: GO TO  99
181: 805 INPUT " PAR. FILE NAME KAL. I = " ; A$
182: OPEN A$ AS  1  FOR  I = 1  TO  N1
183: READ # 1 ; PR(I) : NEXT I : CLOSE  1
184: FOR  I = 1 TO N1: YR(I) = YR(I,1) - PR(I) : NEXT  I
185: 99 PRINT : PRINT : PRINT : PRINT
186: PRINT "LSQ. ESTG. INST. ALGM."
187: PRINT : PRINT " SERIAL NO., OBSERVED, MODELLED & ERROR"; PRINT
188: N4 = N1 - 2*N - 2
189: FOR I = 1  TO  N3 : CX( I ) = C1( I )
190: SV(I) = 0 : SV1(I) = 0: SV2(I) = 0 : NEXT  I
191: FOR K = 1  TO  N4
192: K1 = K :K2=N+K1:GOSUB 100:FOR I=1 TO N3 : SV1(I) =SV(I)
193: NEXT  I : K1 = K
194: FOR  I = 1  TO  N3 : SUM = 0
195: FOR J = 1  TO  N3 : SUM = SUM + P(I,J)* SV1(J)
196: NEXT J :  P1(I) = SUM : NEXT  I
197: D = 1 : FOR I = 1  TO  N3: D = D +SV1(I)* P1(I)
198: NEXT I : FOR J = 1  TO  N3 : SUM = 0
199: FOR I = 1  TO  N3 : SUM = SUM + SV1(I)* P(I,J)
200: NEXT I :  P 2(J) = SUM : NEXT  J
201: D = 1 / D : FOR  I = 1  TO  N3
202: FOR J = 1 TO N3: P(I,J)= P(I,J) - P1(I)* P 2 (J)* D
```

```
203: NEXT J,I : K 2 = K1 + 1
204: K1 = N + K 2 :  GOSUB 100
205: FOR I = 1  TO N 3  : SV1(I) = SV(I) : NEXT I : K1   = K + 2
206: FOR I = 1 TO N 3 : SUM = 0 :  FOR J = 1  to  N 3
207: SUM = SUM + P( I,J )* SV2(J) : NEXT  J
208: P1( I ) =  SUM : NEXT I : D = 1.0
209: FOR I = 1 TO  N 3 : D = D + SV(I)* P1(I)
210: NEXT I : D = 1.0 / D : E1  = 0.0
211: FOR I = 1 TO  N 3 : E1 = E1  + SV1(I)* GX(I)
212: NEXT I : NK 2 = 2 * N + K 2
213: Y = YD( NK 2 , 1 ) - E 1  : Y = Y* D
214: FOR I = 1 TO N 3 : GX(I) = GX(I) + P1(I) * Y : NEXT  I
215: NK 2 = NK 2 +1 : K1 = N + K2 + 1 : GOSUB 100 : K1 = K + 1
216: YB = 0 : FOR  I = 1  TO  N 3
217: YB = YB + SV1 ( I )* GX( I ) :  NEXT  I
218: ERR = YD ( NK 2,1 ) - YB
219: YA( K,1) = NK 2 : YA( K,2 ) = YD( NK2,1 )
220: YA ( K,3 ) = YB : YA( K,4 ) = ERR
221: PRINT  YA(K,1),YA(K,2),YA(K,3),YA(K,4) : PRINT : NEXT  K
222: INPUT " LSQRIV ERROR FILE =" ; Q$
223: CREATE  Q$ AS 7
224: FOR K = 1  TO  N 4
225: PRINT  # 7 : YA( K,4 ) : NEXT  K
226: CLOSE  7
227: SUMA = 0 : SUMB = 0 :  SUMC = 0
228: FOR  K = 1 TO N 4 : SUMA = SUMA + YA ( K,4 )
```

```
229 : SUMB = SUMB + YA ( K,4 ) * YA ( K,4 )
230 : SUMS = SUMS + YA ( K,2 ) * YA ( K,2 ) : NEXT K
231 : SUMA = SUMA / N4 : SUMB = ( SUMB / SUMS )
232 : PRINT " INTEGRAL SQR ERROR & MEAN ERROR"
233 : PRINT SUMB, SUMA : PRINT
234 : PRINT : PRINT : PRINT
235 : NEXT N
236 : STOP
237 : END
NO ERRORS DETECTED
CONSTANT AREA :         24
CODE SIZE :           2469
DATA STMT AREA :         0
VARIABLE AREA :        960
A
```

A1.5 DETERMINATION OF CONSTANT VARIANCE OF MODELLED ERRORS

```
COMPILE BEHGA $N
MIBASIC COMPILER V 2.0
   1: PRINT TAB(20);"TESTING FOR CONSTANT VARIANCE & MEAN ERROR":PRINT
   2: LPRINTER WIDTH  80
   3: REM NON STATIONARY HOURLY FLOW MODEL
   4: INPUT "DATA =": N
   5: INPUT "L 1, L 2 =" ; L 1, L 2
   6: DIM Y ( N )
   7: INPUT "DATA FILE=": D $
   8: OPEN D $  AS 1
   9: FOR I = 1  TO  N
  10: READ  # 1  ; Y ( I )
  11: NEXT  I
  12: CLOSE  1
  13: FOR N 1 = 2  TO  6
  14: NT = N 1
  15: DIM S(NT,NT),A(NT),B(NT),BA(NT),BB(NT),BC(NT)
  16: DIM BE(NT), PE( N+2 )
  17: FOR I = 1  TO  NT
  18: A(I) = 0
  19: B(I) = 0
  20: FOR J = 1  TO  NT
  21: S(I,J) = 0
  22: S(I,I) = 1
  23: NEXT  J
  24: NEXT  I
  25: FOR K = 1  TO  N
  26: SUMB = 1
  27: FOR  I = 1  TO  NT
  28: SUMA = 0 : SUMB = 0
  29: FOR  ID = 1  TO  NT
  30: SUMA = SUMA + S ( I,ID ) * E ( ID )
```

```
31: SUMB = SUMB + Z ( KD ) * S ( KD, I )
32: NEXT KD
33: BA ( I ) = SUMA : BB ( I ) = SUMB
34: NEXT I
35: FOR I = 1 TO NT
36: SUMC = SUMC + BB ( I ) * Z ( I )
37: NEXT I
38: FOR I = 1 TO NT
39: BC ( I ) = BA ( I ) / SUMC
40: FOR J = 1 TO NT
41: S ( I,J ) = S ( I,J ) - BC ( I ) * BB ( J )
42: NEXT J : NEXT I
43: SUMA = Y ( K )
44: FOR I = 1 TO NT
45: SUMA = SUMA - A ( I ) * Z ( I )
46: NEXT I
47: PE ( K ) = SUMA
48: FOR I = 1 TO NT
49: SUMD = 0
50: FOR J = 1 TO NT
51: SUMD = SUMD + S ( I,J ) * Z ( J )
52: BK ( I ) = SUMD : NEXT J : NEXT I
53: FOR I = 1 TO NT
54: A ( I ) = A ( I ) + PE ( K ) * BK ( I )
55: NEXT I
56: REM GENERATION OF Z VECTOR
57: FOR I = 1 TO N1
58: KK1 = K - I + 1
59: IF KK1<=0 THEN 5
60: Z ( I ) = Y ( KK1 )
61: GO TO 6
62: 5 Z ( I ) = 0
63 s6 NEXT I
64: IF K<L1 THEN 3
65: IF K>L2 THEN 3
```

```
66: PRINT "K = "; K
67: PRINT " A ( I ) "
68: FOR  I = 1  TO  NT
69: PRINT  A ( I ) ,
70: NEXT  I
71: PRINT " S ( I ) "
72: FOR  I = 1  TO  NT
73: PRINT  S ( I ),
74: NEXT  I
75: 5 NEXT  K
76: SUMR = 0 : SUMP = 0 : SUMS = 0
77: KK = N1 + 1
78: FOR  K = KK  TO  N
79: SUMR = SUMR + PE ( K ) * PE ( K )
80: SUMP = SUMP + Y ( K ) * Y ( K )
81: SUMS = SUMS + PE ( K )
82: NEXT  K
83: PE ( N + 1 ) = SUMR / SUMP
84: PE ( N + 2 ) = SUMS / N
85: PRINT "INT.SQR. & MEAN ERROR ="; PE(N+1), PE(N+2): PRINT
86: PRINT "N1+ 1 =";  KK : PRINT
87: PRINT "PREHISTORY INTERVAL ="; N1
88: INPUT "FILE NAME ="; C$
89: CREATE  C$ AS  2
90: FOR  I = KK  TO  N
91: PRINT  # 2 ; PE ( I )
92: NEXT  I : CLOSE  2
93: PRINT : PRINT : PRINT
94: NEXT  N1
95: STOP
96: END
NO ERRORS DETECTED
CONSTANT AREA :          8
CODE SIZE  :          1965
DATA STMT AREA :         0
VARIABLE AREA  :       232
A
```

A2.1 MULTILAYER GROUP METHOD DATA HANDLING ALGORITHM

```
COMPILE GMDH [illegible]
MBASIC COMPILER V2.0
  1: REM CONSOLE
  2: LPRINTER WIDTH 80
  3: PRINT TAB( 10 ); "GMDH MULTILAYER" : PRINT
  4: INPUT "DATA, VARIABLES ="; N,K
  5: P = ( K - 1 ) * ( K - 2 ) / 2
  6: DIM X ( N,K ), X1 ( N )
  7: INPUT "DATA FILE ="; B $
  8: OPEN B $ AS 1
  9: FOR J = 1 TO K
 10: FOR I = 1 TO N
 11: READ #1 ; X ( I,J )
 12: NEXT I
 13: NEXT J
 14: CLOSE 1
 15: NN = N + 6 : N 1 = NN + 3
 16: DIM Y (N), X2 (N), A1 (6), B1 (6)
 17: DIM B2 (6), B3(6), B4(6), B5(6), B6(6), B(6)
 18: DIM D3(6), D5(6), D4(6), [illegible](6), D6(6), E2(6)
 19: DIM E4(6), E5(6), E6(6), F4(6), F5(6), F6(6)
 20: DIM C5(6), C6(6), C(6), YN( P,N1 )
 21: FOR I = 1 TO N
 22: Y(I) = X( I,1 )
 23: NEXT I
 24: K L = 0
 25: FOR J = 2 TO K - 1
 26: J1 = J - 1
 27: FOR I = 1 TO N
 28: X 1 (I) = X (I,J)
 29: NEXT I
 30: JJ = J + 1
 31: FOR L = JJ TO K
 32: K3 = KL + L - J
```

```
33: JJ1 = L - 1
34: FOR I = 1 TO N
35: X 2 (I) = X (I, L)
36: NEXT I
37: FOR I = 1 TO 6
38: A 1 (I) = 0
39: NEXT I
40: FOR I = 1 TO N
41: A 1 ( 1 ) = A 1 ( 1 ) + Y ( I )
42: A 1 (2) = A 1 (2) + Y (I) * X 1 (I)
43: A 1 (3) = A 1 (3) + Y (I) * X 2 (I)
44: A 1 (4) = A 1 (4) + Y (I) * ( X 1 (I) * X 2 (I))
45: A 1 (5) = A 1 (5) + Y (I) * ( X 1 (I) * X 1 (I))
46: A 1 (6) = A 1 (6) + Y (I) * ( X 2 (I) * X 2 (I))
47: NEXT I
48: B 1 ( 1 ) = N
49: FOR I = 2 TO 6
50: B 1 ( I ) = 0
51: NEXT I
52: FOR I = 1 TO N
53: B 1 (2) = B 1 (2) + X 1 (I)
54: B 1 (3) = B 1 (3) + X 2 (I)
55: B 1 (4) = B 1 (4) + X 1 (I) * X 2 (I)
56: B 1 (5) = B 1 (5) + X 1 (I) * X 1 (I)
57: B 1 (6) = B 1 (6) + X 2 (I) * X 2 (I)
58: NEXT I
59: B 2 ( 1)= B 1 (2) : B 2 ( 2 ) = B 1 (3) : B 2 (3) = B 1 (4)
60: FOR I = 4 TO 6
61: B 2 (I) = 0
62: NEXT I
63: FOR I = 1 TO N
64: B 2 (4) = B 2 (4) + X 1 (I) * ( X 1 (I) * X 2 (I))
65: B 2 (5) = B 2 (5) + X 1 (I) * X 1 (I) * X 1 (I)
66: B 2 (6) = B 2 (6) + X 1 (I) *(X 2 (I) * X 2 (I))
```

```
67: NEXT I
68: B 3 (1) = B 1 (3) : B 3 (2) = B 2 (3) : B 3 (3) = B 1 (6)
69: B 3 (4) = B 2 (6) : B 3 (5) = B 3 (4)
70: B 3 (6) = 0
71: FOR I = 1 TO N
72: B 3 (6)= B 3 (6) + X 2 (I) * ( X 2 (I) * X 2 (I))
73: NEXT I
74: B 4 (1) = B 1 (4) : B 4 (2) = B 2 (4) : B 4 (3) = B 3 (4)
75: FOR I = 4 TO 6
76: B 4 (I) = 0
77: NEXT I
78: FOR I = 1 TO N
79: B 4 (4) = B 4 (4) + (X 1(I)*X 1(I))*(X 2(I)* X 2 (I))
80: B 4 (5) = B 4 (5) + (X 1(I)*X 1(I))*(X 1(I)* X 2 (I))
81: B 4 (6) = B 4 (6) + (X 2(I)*X 2(I))*(X 1(I)* X 2 (I))
82: NEXT I
83: B 5 (1) = B 1 (5) : B 5 (2) = B 2 (5) : B 5 (3) = B 3 (5)
84: B 5 (4) = B 4 (5) : B 5 (5) = 0 : B 5 (6) = B 4 (6)
85: FOR I = 1 TO N
86: B 5 (5)= B 5 (5)+ (X 1 (I)* X 1 (I)) * (X 1 (I)* X 1 (I))
87: NEXT I
88: B 6 (1) = B 1 (6) : B 6 (2) = B 2 (6) : B 6 (3) = B 3 (6)
89: B 6 (4) = B 4 (6) : B 6 (5) = B 5 (6) : B 6 (6) = 0
90: FOR I = 1 TO N
91: B 6 (6) = B 6 (6)+ (X 2 (I)* X 2 (I))* ( X 2 (I) * X 2 (I))
92: NEXT I
93: Z (1) = B 1 (1) / B 2 (1)
94: FOR I = 2 TO 6
95: B 2 (I) = B 2 (I) * Z (1) - B 1 (I)
96: NEXT I
97: Z (2) = B 1 (1) / B 3 (1)
98: FOR I = 2 TO 6
99: B 3 (I) = B 3 (I) * Z (2) - B 1 (I)
100: NEXT I
101: Z (3) = B 1 (1) / B 4 (1)
```

```
102: FOR  I = 2  TO  6
103: D 4 (I) = D 4 (I) * Z (3) - D 1 (I)
104: NEXT  I
105: Z (4) = D 1 (1) / D 5 (1)
106: FOR  I = 2  TO  6
107: D 5 (I) = D 5 (I) * Z (4) - D 1 (I)
108: NEXT  I
109: Z (5) = D 1 (1) / D 6 (1)
110: FOR I = 2  TO  6
111: D 6 (I) = D 6 (I) * Z (5) - D 1 (I)
112: NEXT  I
113: FOR  I = 1  TO  5
114: II = I + 1
115: A 1 (II) = A 1 (II) * Z (I) - A 1 (1)
116: NEXT  I
117: Z (1) = D 2 (2) / D 3 (2)
118: FOR  I = 3  TO  6
119: D 3 (I) = D 3 (I) * Z (1) - D 2 (I)
120: NEXT  I
121: Z (2) = D 2 (2) / D 4 (2)
122: FOR  I = 3  TO  6
123: D 4 (I) = D 4 (I) * Z (2) - D 2 (I)
124: NEXT  I
125: Z (3) = D 2 (2)/ D 5 (2) : Z (4) = D 2 (2)/ D 6 (2)
126: FOR  I = 3  TO  6
127: D 5 (I) = D 5 (I) * Z (3) - D 2 (I)
128: D 6 (I) = D 6 (I) * Z (4) - D 2 (I)
129: NEXT  I
130: FOR  I = 1  TO  4
131: II = I + 2
132: A 1 (II) = A 1 (II) * Z (I) - A 1 (2)
133: NEXT  I
134: Z (1) = D 3 (3)/ D 4 (3) : Z (2) = D 3 (3)/ D 5 (3)
135: Z (3) = D 3 (3)/ D 6 (3)
136: FOR  I = 4  TO  6
137: D 4 (I) = D 4 (I) * Z (1) - D 3 (I)
```

```
138: F 5(I) = E 5 (I) * Z(2)  - E3 (I)
139: F 6(I) = E6 (I) * Z(3)  - E3 (I)
140: NEXT  I
141: FOR  I = 1  TO  3
142: II = I + 3
143: A 1(II) = A 1 (II) * Z(I) - A 1 (3)
144: NEXT  I
145: Z(1) = F 4(4)/F 5(4) : Z(2) = F 4(4)/ F 6(4)
146: FOR  I = 5  TO  6
147: G 5(I) = F 5 (I) * Z(1) - F 4 (I)
148: G 6(I) = F 6 (I) * Z(2) - F 4 (I)
149: NEXT  I
150: FOR  I = 1  TO  2
151: II = I + 4
152: A 1 (II) = A 1 (II) * Z(I) - A 1 (4)
153: NEXT  I
154: Z (1) = G 5 (5)/ G 6 (5)
155: H 6 = G 6 (6) * Z(1) - G 5 (6)
156: A 1 (6) = A 1 (6) * Z (1) - A 1 (5)
157: C (6) = A 1 (6)/ H 6
158: C (5) = ( A 1 (5) - G 5 (6) * C (6))/ G 5(5)
159: C (4) = ( A 1 (4) - F 4 (5) * C (5) - F 4(6) * C(6))/ F 4 (4)
160: A 2 = E 3(4)* C(4) :  A 3 = E 3(5) * C(5)
161: A 4 = E 3(6) * C(6)
162: C(3) = ( A 1 (3) - A 2 - A 3 - A 4 )/ E 3 (3)
163: A 2 = D 2 (3) * C (3) : A 3 = D 2 (4) * C (4)
164: A 4 = D 2 (5) * C (5) : A 5 = D 2 (6) * C (6)
165: A 2 = A 2 + A 3 + A 4 + A 5
166: C (2) = ( A 1(2) - A 2 )/ D 2(2)
167: A 2 = B1(2) * C(2)  : A 3 = B 1 (3) *  C(3)
168: A 4 = B 1 (4) * C(4) : A 5 = B 1(5) * C(5)
169: A 6 = B 1 (6) * C(6) : A Z = A 2 + A 3 + A 4  + A 5 + A 6
170: C (1) =  ( A 1 (1) - A 2 )/ B 1 (1)
```

```
171: FOR I = 1  TO  6
172: YB ( K 3, N + I ) = C (I)
173: NEXT  I
174: FOR  I = 1  TO  N
175: A = C (1) + C(2) * X 1 (I) + C(3) * X 2 (I)
176: B = C (4) * X 1(I) * X 2(I)
177: C 1 = C(5) * X 1 (I) * X 1 (I)
178: D = C(6) * X 2 (I) * X 2 (I)
179: YB ( K 3, I ) = A + B + C 1 + D
180: NEXT  I
181: SUMM =  0 : SUMY = 0
182: FOR  I = 1  TO  N
183: SUMM = SUMM +(Y(I) - YB(K 3,I))*(Y(I) - YB (K 3,I))
184: SUMY = SUMY + Y(I) * Y(I)
185: NEXT  I
186: DEL 1 = SUMM / SUMY
187: PRINT TAB (10);"DEL 1 (";K 3 ;") =";DEL 1
188: PRINT
189: YB ( K 3, NN + 1 ) = J1
190: YB ( K 3, NN + 2 ) = JJ1
191: YB ( K 3, NN + 3 ) = DEL 1
192: NEXT  L
193: K L = K 3
194: NEXT  J
195: FOR  I = 1  TO  P - 1
196: FOR  J = I + 1  TO  P
197: IF YB ( J,M 1 ) > Y B ( I,M 1)  THEN  10
198: FOR  L = 1  TO  M 1
199: Z 1 = Y B ( I,L )
200: YB (I,L) = YB ( J,L )
201: YB ( J,L ) = Z 1
202: NEXT  L
203: 10  NEXT  J
204: NEXT  I
205: REM FOR I = 1  TO  K - 1
```

```
206: REM PRINT " ROW = "; I
207: REM FOR J = N + 1 TO N 1
208: REM PRINT YB ( I, J ),
209: REM NEXT J
210: REM NEXT I
211: INPUT "COEFF. & ERR FILE COEFF. I ="; C $
212: CREATE C $ AS 1 ; FOR I = 1 TO K - 1 ; FOR J = N + 1 TO N 1
213: PRINT # 1 ; YB ( I,J ) ; NEXT J ; NEXT I: CLOSE 1
214: FOR J = 2 TO K
215: FOR I = 1 TO N
216: X ( I,J ) = YB ( J - 1 , I )
217: NEXT I
218: NEXT J
219: INPUT "DATA FILE ="; A $
220: CREATE A $ AS 2
221: FOR J = 1 TO K
222: FOR I = 1 TO N
223: PRINT # 2 ; X ( I, J )
224: NEXT I
225: NEXT J
226: CLOSE 2
227: PRINT
228: STOP
229: END
NO ERRORS DETECTED
CONSTANT AREA :          8
CODE SIZE :           6116
DATA STMT AREA :         0
VARIABLE AREA :        472
A
```

A5.1 DETERMINATION OF PERIODICITY IN THE INPUT DATA

```
COMPILE SPECTRA $ B
HIBASIC COMPILER V 2.0
   1: REM POWER DENSITY
   2: LPRINTER WIDTH 80
   3: INPUT "DATA, VARS. = "; N,K
   4: M = K : M 1 = M + 1
   5: DIM P(N), PS(M 1 ), HP(M 1), S( M 1 ), G A ( M 1 ), RA ( M 1 )
   6: INPUT "FILE NAME = "; B $
   7: OPEN B $ AS 1
   8: FOR I = 1 TO N
   9: READ # 1 ; P (I)
  10: NEXT I
  11: CLOSE 1
  12: SUM = 0
  13: FOR I = 1 TO N
  14: SUM = SUM + P (I)
  15: NEXT I
  16: PMN = SUM / N
  17: FOR K = 1 TO M 1
  18: NK = N - K : SUM = 0
  19: FOR I = 1 TO NK
  20: II = I + K - 1
  21: SUM = SUM + (P(I) - PMN) * (P(II) - PMN)
  22: NEXT I
  23: GA(K) = SUM/(N - K + 1) : RA (K) = GA (K)/ GA (1)
  24: NEXT K
  25: NK =.5
  26: FOR IH =1 TO M 1
  27: G = IH - 1 : A = G * 3.142/ M : SUM = 0
  28: HP ( IH ) = .5 * A/ 3.142
  29: FOR K = 1 TO M 1
  30: B = A * ( K - 1 )
```

```
31: IF  K = 2  THEN  HK  = 1.
32: IF  K = N 1  THEN  HK = .5
33: SUM = SUM + HK * RA (K) * COS (B)
34: NEXT  K
35: PS ( IH ) =  SUM *  2 / 3.142
36: NEXT  IH
37: S(1) = .54 * PS(1) + .46 * PS(2)
38: S(N 1) = .54 * PS(N 1) + .46 + PS(N)
39: FOR  I = 2  TO  N
40: S(I) = .25 * PS(I - 1) + .54 * PS(I) + .25 * PS (I + 1)
41: NEXT  I
42: PRINT "SPECTRA"
43: FOR  I = 1  TO  N 1
44: PRINT  S(I),
45: NEXT  I
46: PRINT "FREQUENCY"
47: FOR  I = 1  TO  N 1
48: PRINT  HF(I),
49: NEXT  I
50: PRINT
51: STOP
52: END
NO ERRORS DETECTED
CONSTANT AREA  :        48
CODE SIZE  :          1035
DATA STMT AREA  :        0
VARIABLE AREA   :      160
A
```

A3.2 COMBINATORIAL GROUP METHOD OF DATA HANDLING ALGORITHM

```
COMPILE GMDCOMB #E
HIBASIC COMPILER V 2.0
    1: LPRINTER WIDTH 80
    2: PRINT "COMBINATORIAL GMDH ALGORITHM POLINOMIAL"
    3: PRINT "WITHOUT SQUARE TERMS"
    4: PRINT
    5: INPUT "DATA & VARIABLES" ; N,K
    6: KK = ( K - 1 )*K / 2 + 2
    7: DIM Y(KK,KK),X(N,K+1),B(KK),XM(N),C(KK),B1(KK),
           YO(KK,KK), CR(15)
    8: DIM XE(N)
    9: INPUT "INPUT DATA FILE NAME" ; D $
   10: OPEN D$ AS 1: FOR I=1 TO N:READ #1;X(I,1):NEXT I
   11: FOR J = 3 TO K+1:FOR I = 1 TO N:READ #1; X(I,J):NEXT I:
           NEXT J:CLOSE 1
   12: FOR I = 1 TO N:X(I,2)=1: NEXT I
   13: PRINT "TEST DATA":FOR I = 1 TO K+1:PRINT X(1,I),:NEXT I
   14: PRINT
   15: INPUT "MATRIX FORMATION TO BE SKIPPED ? YES = 1 NO = 0";A
   16: IF A = 1  THEN 20
   17: FOR I= 1 TO KK - 1: FOR J = 1 TO KK:YO(I,J)=0:NEXT J,I
   18: FOR I= 1 TO N:B(1)=X(I,1):B(2)=1:M=2:FOR L=2 TO K:L1=L+1
   19: FOR J=L1 TO K+1:M=M+1:B(M)=X(I,L)* X(I,J):NEXT J,L
   20: FOR L=1 TO M-1:FOR J=1 TO M:YO(L,J)= YO(L,J)+B(L+1)*B(J):
           NEXT J: NEXT L
```

```
NEXT I:INPUT"COMBINATORIAL MATRIX FILE NAME COMBN.I"; A$
CREATE A$ AS 1: FOR J=1 TO KK:FOR I=1 TO KK - 1
PRINT #1; YO(I,J): NEXT I,J: CLOSE 1
GO TO 30
20 INPUT "COMBINATORIAL MATRIX FILE COMBN.I"; A$
OPEN A$ AS 1:FOR J=1 TO KK:FOR I=1 TO KK - 1
READ #1; YO(I,J): NEXT I, J : CLOSE 1
30 INPUT "HOW MANY COMBN WANT TO TEST ?"; EL
FOR LOOP = 1 TO EL : FOR J = 1 TO KK: FOR I=1 TO KK - 1
Y( I,J ) = YO( I,J ): NEXT  I,J
INPUT "HOW MANY TERMS OF POLMN. TO BE SKIPPED ?"; TN
IF TN = 0 THEN 40
IF TN = 100 THEN 100
FOR V = 1 TO TN: INPUT "WHICH ONE SERIALLY ?"; CR( V )
IF CR ( V ) = KK  THEN  60
FOR J=CR(V) - V+1 TO KK- V: FOR I = CR(V)- V TO KK - V - 1
Y(I,J)=Y(I+1,J+1):NEXT I,J,V
40 FOR J = 1 TO KK- TN:B1(J)= Y(J,1): NEXT J
FOR J=1 TO KK- TN- 1:FOR I=1 TO KK- TN- 1:Y(I,J)=Y(I,J+1):NEXT I,J
NN=KK- TN- 1:PRINT"NO. OF TERMS OF THE TESTED POLMN.";NN:PRINT
PRINT"Y(1,J)":PRINT:FOR J=1 TO NN:PRINT Y(1,J); :NEXT J:PRINT
REM MATRIX INVERSION ALGORITH
60 FOR L = 1 TO NN
Z=1/Y(L,L):PRINT"Z=";Z:PRINT:Y(L,L)=1:FOR I=1 TO NN
Y(I,L) = Y(I,L) * Z:NEXT I: FOR J=1  TO  NN
```

```
IF J = L THEN 50
Z = Y(L,J): Y(L,J) = 0: FOR I = 1 TO NN
Y( I,J ) = Y ( I,J ) - Y( I,L )* Z : NEXT I
50 NEXT J
NEXT L
PRINT "MATRIX INVERSED"
PRINT TAB (10);"COEFFICIENTS OF THE POLYNOMIAL": PRINT
FOR I=1 TO NN:SUM=0:FOR J=1 TO NN:SUM=SUM+Y(I,J) * B1(J)
NEXT J:C(I)=SUM:PRINT C(I),: NEXT I
FOR I=1 TO N:SUM=0:B(1)=X(I,1): B(2)= X(I,2):M=2
FOR L=2 TO K:L2=L+1:FOR J=L2 TO K+1:M=M+1
B(M)= X(I,L)* X(I,J): NEXT J: NEXT L
FOR J=2 TO M-TH:SUM=SUM+C(J -1) * B(J): NEXT J
XM(I) = SUM : NEXT I
FOR I=1 TO N:XE(I)=X(I,1) -XM(I): NEXT I
PRINT TAB(1);"SERIAL";TAB(15);"OBSERVED";TAB(35);
     "MODELLED VALUES"; TAB
FOR I=1 TO N:PRINT TAB(1);I;TAB(15);X(I,1);TAB(35);XM(I);
     TAB(55); XE(I)
PRINT : NEXT I
SUMC = 0: SUMA= 0: SUMB= 0: FOR I = 1 TO N
SUMA=SUMA+XE(I)* XE(I): SUMB=SUMB+X(I,1) * X(I,1)
SUMC = SUMC + XE ( I )
NEXT I: ERR = SUMA/SUMB: SUMC = SUMC/N
PRINT "INTEGRAL SQUARE ERROR ="; ERR : PRINT
PRINT "MEAN ERROR ="; SUMC : PRINT
```

```
INPUT "ERROR FILE NEEDED ? Y/N = 1/ 0"; A
IF A = 0  THEN 11
INPUT "ERROR FILE XERR. I"; A$
CREATE A$ AS 1: FOR I = 1 TO N:PRINT  #1 ; XE( I )
NEXT I: CLOSE  1
INPUT "MATRIX PRINT FILE REQD. ? YES = 1 NO = 0"; B
IF B = 0  THEN  11
INPUT "INVERSED MATRIX FILE COMBMAT.I"; A$
CREATE A$  AS  1
FOR I=1 TO NN:FOR J=1 TO NN:PRINT #1;Y(I,J):NEXT J,I:CLOSE 1
11  NEXT  LOOP
100  STOP
END
NO ERRORS DETECTED
CONSTANT AREA :          8
CODE SIZE    :        3668
DATA STMT AREA :         0
VARIABLE AREA  :       264
A
```

A4.1 POWER SYSTEM STATES ESTIMATION

```
COMPILE NRSE $E
HIBASIC COMPILER V2.0
    LPRINTER WIDTH 80
    PRINT TAB(5);"POWER SYSTEM STATE ESTIMATION"
    PRINT
    INPUT "NO. OF BUSES & NO. OF LINES ="; NB, LPQ
    DIM Y(LPQ,6), Z(LPQ),G(NB,NB), B(NB,NB)
    INPUT "P - Q; R,X; YPQ1/2 (R & I) FILE YZ ="; Y$
    OPEN Y$ AS 1: FOR I=1 TO LPQ : FOR J = 1 TO 6
    READ #1 ; Y(I,J): NEXT J,I
    CLOSE 1
    INPUT "PQRX PRINT NEEDED ? YES = 1; NO=0="; PQR
    IF PQR = 0 THEN 501
    PRINT "P Q R X YPZ1/2 - R YPQ1/2-X"
    FOR I=1 TO LPQ: PRINT" I=";I: FOR J = 1 TO 6
    PRINT Y(I,J),: NEXT J: PRINT: NEXT I
    501 FOR I = 1 TO LPQ
    Z=Y(I,3)*Y(I,3)+Y(I,4)*Y(I,4):Y(I,3)=Y(I,3)/Z:
        Y(I,4)= -Y(I,4)/Z: NEXT I
    INPUT "PQ ADMITTANCE PRINT NEEDED ? YES = 1 NO=0="; L
    IF L = 0 THEN 502
    PRINT "P-Q LINE ADMITTANCE & LINE CHARGING"
    FOR I=1 TO LPQ:PRINT"I=";I:FOR J=1 TO 6:PRINT Y(I,J),:
        NEXT J: PRINT: NEXT I
    502 FOR J=1 TO NB:FOR I=1 TO NB:G(I,J)=0:B(I,J)=0:NEXT I,J
```

```
FOR I=1 TO NB:SUMA=0:SUMB=0:FOR J=1 TO LPQ
IF (X( J,1 ) -I)<>0 THEN 10
SUMA=SUMA+Y(J,3)+Y(J,5):SUMB=SUMB+Y(J,4)+Y(J,6)
10 NEXT J
FOR J=1 TO LPQ
IF (X(J,2) - I)<>0 THEN 11
SUMA=SUMA+Y(J,3)+Y(J,5):SUMB=SUMB+Y(J,4)+Y(J,6)
11 NEXT J
G(I,I)= SUMA: B(I,I)= SUMB
IF I=NB THEN 9
FOR J=1 TO LPQ
IF (X(J,1) - I)<> 0 THEN 21
II=I+1: FOR K=II TO NB
IF (X(J,2)-K)<>0 THEN 41
G(I,K)= -Y(J,3):G(K,I)=G(I,K ):B(I,K)= -Y(J,4):B(K,I)=B(I,K)
41 NEXT  K
21 NEXT  J
9 NEXT  I
INPUT "BUS ADMITTANCE PRINT NEEDED ? YES=1 NO= 0="; BUS
IF BUS = 0 THEN 503
PRINT "BUS ADMITTANCE MATRIX": PRINT
PRINT "ACTIVE ADMITTANCE COMPONENTS": PRINT
FOR I=1 TO NB:PRINT"ROW=";I:FOR J=1 TO NB:PRINT G(I,J),:
        NEXT J: PRINT
NEXT I:PRINT:PRINT"REACTIVE ADMITTANCE COMPONENTS"
```

```
FOR I=1 TO NB:PRINT"ROW=";I:FOR J=1 TO NB:PRINT B(I,J),:
        NEXT J: PRINT
NEXT I:PRINT:PRINT"ASSUMPTION OF BUS VOLTAGES"
503 INPUT"REFERENCE BUS J1=";J1
DIM E(NB,1),F(NB,1),PL(NB,4),P(NB,2),Q(NB,2),S(2*NB,2* NB)
FOR I=1 TO NB:FOR J=1 TO NB:B(I,J)=-B(I,J):NEXT J,I:FOR I=1 TO N
INPUT "E(I,1) & F( I,1) ="; E( I,1),F( I,1):NEXT I
INPUT"GENERATION & LOAD POWER FILE =GL="; B$
OPEN B$ AS 2:FOR I=1 TO NB:FOR J=1 TO 4:READ #2;PL(I,J):
        NEXT J,I: CLOSE 2
PRINT "GENERATION ACTIVE & EACTIVE LOAD ACTIVE & REACTIVE"
INPUT "MVA BASE ="; BASE
INPUT "LOAD POWER PRINT NEEDED ? YES=1 NO= 0="; LP
IF LP = 0 THEN 504
FOR I=1 TO NB: FOR J=1 TO 4:PRINT PL(I,J),: NEXT J,I
504 FOR I = 1 TO NB
PL(I,1)=(PL(I,1)-PL(I,3))/BASE:PL(I,2)= - (PL(I,2)-PL(I,4))/BASE
P(I,1) = PL(I,1): Q(I,1)= PL( I,2 )
NEXT I
INPUT "TOLLERANCE=" ; TL
INPUT "SIGMA="; SIGMA
INPUT "THETA="; THETA
ITN=1:NB2=2 * NB
FOR I=1 TO NB2 - 1: FOR J=1 TO NB2 - 1: S(I,J) = 0
S( I,I ) = SIGMA: NEXT J: NEXT I
100 FOR I=1 TO NB
```

```
70: SUMA = 0 : SUMB = 0
71: FOR J = 1 TO NB
72: P1 =E( I,1)* (E(J,1)*G(I,J)+ F( J,1 )* B( I,J ))
73: P2 =F( I,1)* (F(J,1)*G(I,J)- E( J,1 )* B( I,J ))
74: Q1 =F( I,1)* (E(J,1)*G(I,J)+ F( J,1 )* B( I,J ))
75: Q2 =- E(I,1)*(F(J,1)*G(I,J)- E(J,1 ) * B( I,J ))
76: SUMA = SUMA+ P1+P2: SUMB = SUMB + Q1 + Q 2
77: NEXT J:P(I,2)=SUMA:Q(I,2)= SUMB:NEXT I: KITN= ITN - 1
78: PRINT TAB(10);"REAL AND REACTIVE BUS POWER "
79: PRINT:PRINT "BUS NO.,P(I,K+1),P(I,K),Q(I,K+1),Q(I,K)":PRINT
80: FOR I=1 TO NB:PRINT TAB(10);"BUS(";I;")": PRINT
81: PRINT P(I,2),P(I,1),Q(I,2),Q(I,1):PRINT: NEXT I: PRINT
82: FOR I=1 TO NB:PRINT TAB(10); Q(I,2),Q(I,1):PRINT:NEXT I:PRINT
83: DIM DELP(NB), DELQ(NB), C(NB), D(NB),JN(NB2,NB2)
84: DIM JN1(NB,NB),JN2(NB,NB),JN3(NB,NB),JN4(NB,NB)
85: DIM DELE(NB),DELF(NB)
86: PRINT TAB(10);"DELP(I) DELQ(I)":PRINT
87: FOR I = 1 TO NB
88: DELP(I)= P(I,1) -P(I,2):DELQ(I)= Q(I,1) - Q( I,2 )
89: PRINT I, DELP(I), DELQ(I): PRINT
90: NEXT I
91: DIM YD(NB,2)
92: FOR I= 1 TO NB
93: YD( I,1 ) = DELP(I): YD( I,2 ) = DELQ(I)
```

```
NEXT I
FOR I = 1  TO NB
IF YD( I,1 )< 0  THEN 58
GO TO 57
58 YD( I,1 ) = - YD( I,1 )
57 IF YD( I,2 )< 0  THEN 59
GO TO 60
59 YD( I,2 ) = - YD( I,2 )
60 NEXT  I
DX1 = 0: DX2 = 0
FOR I = 1  TO NB
IF ( DX1 - YD ( I,1 ))> 0  THEN  62
DX1 = YD( I,1 )
62 IF ( DX 2 - YD ( I,2 ))> 0  THEN  61
DX 2 = YD ( I,2 )
61 NEXT  I
IF ( DX1 - DX 2 )> 0  THEN  63
DX1 = DX2
63 IF ( DX1 - TL )< = 0  THEN  64
PRINT "ITERATION COUNT = "; KITN: PRINT
FOR I = 1 TO NB: C(I) = 0:D(I)= 0:NEXT I
FOR I = 1 TO NB
D1 = E( I,1 ) * E( I,1 ) + F( I,1 ) * F( I,1 )
C(I)= ( P( I,2 )* E( I,1 )+ Q( I,2 )* F( I,1 ))/D1
D(I)= ( P( I,2 )* F( I,1 )- Q( I,2 )* E( I,1 ))/D1
```

```
NEXT I
PRINT: PRINT: PRINT
INPUT "JACOBIAN REQRD ? YES = 1  NO = 0"; BB
IF BB = 0  THEN 556
FOR I = 1  TO NB
JN1( I,I ) = E(I,1)* G(I,I)-F(I,1)* B(I,I)+ C(I)
NEXT I
FOR I = 1  TO NB
FOR J = 1  TO NB
IF I = J  THEN 167
JN1(I,J) = E(I,1)* G(I,J)-F(I,1)* B(I,J)
167 NEXT J
NEXT I
FOR I = 1 TO NB : FOR J = 1  TO NB
JN2 (I,J) = 0: NEXT J,I
FOR I = 1  TO NB
JN2(I,I) =E(I,1)* B(I,I)+2* F(I,1)* G(I,I)+D(I)
NEXT I
FOR I = 1  TO NB
FOR J = 1  TO NB
IF I = J THEN 72
JN2(I,J) = E(I,1)* B(I,J)+ F(I,1)* G(I,J)
72 NEXT J
NEXT I
FOR I = 1  TO NB
FOR J = 1  TO NB
```

```
IF I = J  THEN  74
JN3(I,J) = E(I,1)* B(I,J)+ F(I,1)* G(I,J)
74 NEXT  J
NEXT  I
FOR I = 1  TO NB
JN3(I,I) = E(I,1)* B(I,I)+ F(I,1)* G(I,I) - D(I)
NEXT  I
FOR  I = 1  TO NB
FOR  J = 1  TO NB
IF I = J  THEN 76
JN4(I,J) =- E(I,1)* G(I,J)+ F(I,1)* B(I,J)
76 NEXT  J
NEXT  I
FOR  I = 1  TO NB
JN4(I,I) =-E(I,1)* G(I,I)+ F(I,1)* B(I,I)+ G(I)
NEXT  I
FOR I=1 TO NB:FOR J=1 TO NB:JN(I,J)= JN1(I,J): NEXT J,I
FOR I=1 TO NB:FOR J=NB+1 TO NB2:JN(I,J)=JN2(I,J-NB):NEXT J,I
FOR I= NB+ 1 TO NB2 : FOR J = 1  TO NB
JN( I,J ) = JN3( I - NB,J) : NEXT  J,I
FOR I=NB+1 TO NB2: FOR J= NB+1  TO NB2
JN( I,J ) = JN4( L-NB,J-NB) : NEXT  J,I
PRINT
INPUT "PRINT COPY OF JACOBIAN NEEDED YES = 1 NO=0="; LL
IF LL = 0 THEN 555
PRINT "ELEMENTS OF JACOBIAN"
```

```
PRINT: FOR I = 1 TO NB2: PRINT"ROW=";I
FOR J=1 TO NB2:PRINT JN(I,J),:NEXT J:PRINT:NEXT I:PRINT
555 PRINT
DIM DELE(NB), DELF(NB)
FOR I=1 TO NB:DELE(I)= 0:DELF(I)=0:NEXT  I
FOR I = 1  TO NB2
FOR J=NB+1 TO NB2-1:JN(I,J)=JN(I,J+1):NEXT J:NEXT I
DIM YP(NB2),DELV(NB2),Z(NB2)
FOR I=1 TO NB:DELV(I)=DELE(I):NEXT I:FOR I=NB+1 TO NB2-1
DELV(I)= DELF(I-NB+1)
NEXT I:FOR I=1 TO NB:YP(I)= DELP(I): NEXT I
FOR I=NB+1 TO NB2:YP(I)= DELQ(I-NB): NEXT I
REM FOR  I=1 TO NB2-1: FOR J=1 TO NB2-1:S(I,J)=0
REM S(I,I) = SIGMA: NEXT J:NEXT I
FOR IS = 1 TO NB2
DIM PE(NB2),DK(NB2),A(NB2),DA(NB2),DB(NB2),DC(NB2)
FOR I = 1 TO NB2-1:A(I)= DELV(I): NEXT I
FOR I = 1 TO NB2-1
Z(I)= JN(IS,I):NEXT I:Y=YP(IS)
FOR I=1 TO NB2-1:SUMA=0:SUMB=0:FOR KD=1 TO NB2-1
SUMA=SUMA+S(I,KD) * Z(KD):SUMB=SUMB+Z(KD * S(KD,I):NEXT KD
DA(I)=SUMA:DB(I)=SUMB:DB(I)=DB(I) * THETA:NEXT I:SUMC=1
FOR I=1 TO NB2-1: SUMC=SUMC+DB(I) * Z(I):NEXT I
FOR I=1 TO NB2-1:DC(I)= DA(I)/SUMC: FOR J=1 TO NB2-1
S(I,J)=S(I,J) - DC(I) * DB(J):NEXT J:NEXT I: SUMA=Y
```

```
196: FOR I=1 TO NB2-1: SUMA=SUMA-A(I)* Z(I):NEXT I:PE(IS) = SUMA
197: FOR I=1 TO NB2-1: SUMD=0: FOR J=1 TO NB2-1
198: SUMD=SUMD+S(I,J) * Z(J)
199: DK(I)=SUMD:DK(I)=DK(I)*THETA:NEXT J:NEXT I:FOR I=1 TO NB2-1
200: A(I)= A(I)+PE(IS)* DK(I):NEXT I: NEXT IS
201: PRINT TAB(10); "A(I)": PRINT
202: PRINT "CHANGE OF ACTIVE & REACTIVE BUS VOLTAGES"
203: PRINT
204: FOR I=1 TO NB2-1: PRINT A(I),: NEXT I: PRINT
205: INPUT "S(I,J) PRINT REQD. ? YES = 1 NO=0"; PP
206: IF PP = 0  THEN 666
207: PRINT TAB(10);"S(I,J) MATRIX": PRINT
208: FOR I=1 TO NB2-1:FOR J=1 TO NB2-1:PRINT S(I,J),:NEXT J:
            PRINT:PRINT:NEXT I
209: 666 FOR I = 1 TO NB:DELE(I)=A(I):NEXT I
210: FOR I=2 TO NB:DELF(I)=A(NB+I-1):NEXT  I
211: FOR I=1 TO NB:E(I,1)=E(I,1)+DELE(I):NEXT I:ITN=ITN+1:DELF(J1)=0
212: FOR I=2 TO NB:F(I,1)= F(I,1)+ DELF(I): NEXT I
213: DIM EMAG (NB), GAMMA (NB)
214: FOR I=1 TO NB:A=E(I,1)* E(I,1)+ F(I,1)* F(I,1)
215: EMAG(I)=SQR(A):GAMMA(I)= - (180/3.142)* ATN(F(I,1)/E(I,1)):NEXT I
216: PRINT TAB(5);"BUS NO. ACTIVE AND REACTIVE VOLTAGES VOLTAGE
            MAGNITUDE ANGL
217: PRINT
218: FOR I=1 TO NB:PRINT TAB(10);I,E(I,1),F(I,1):PRINT
```

```
219: PRINT TAB(10); EMAG(I),GAMMA(I):PRINT:NEXT I
220: FOR I=1 TO NB:P(I,1)=P(I,2):Q(I,1)=Q(I,2):NEXT I
221: GO TO 100
222: 64 PRINT
223: PRINT TAB(10);"STATE ESTIMATION RESULTS"
224: PRINT
225: PRINT TAB(10); "ACTIVE BUS VOLTAGES": PRINT
226: FOR I=1 TO NB: PRINT E(I,1), : NEXT I:PRINT
227: PRINT TAB(10); "REACTIVE BUS VOLTAGES": PRINT
228: FOR I = 1 TO NB:PRINT F(I,1),: NEXT I: PRINT
229: STOP
230: END
NO ERRORS DETECTED
CONSTANT AREA :          16
CODE SIZE :            9022
DATA STMT AREA  :         0
VARIABLE AREA  :        568
A
```

A6.1 OPTIMAL ORDERING BY TINNEY'S SCHEME 2

COMPILE OPTILOR $B

MIBASIC COMPILER V2.0

```
LPRINTER WIDTH 80
REM OPTIMAL ORDERING ALGORITHMS TINNEY'S SCHEME 2
INPUT "NO. OF BUSES & NO. OF LINES="; NB, LPQ
DIM AX(NB,NB)
DIM Y(LPQ,7),YBUSR(NB,NB),YBUSI(NB,NB)
INPUT"P-Q-R-X-YPQR/2-YPQX/2-TR FILE YZ=";Y$
OPEN Y$ AS 1: FOR I=1 TO LPQ:FOR J=1 TO 7
READ #1;Y(I,J):NEXT J,I: CLOSE 1
INPUT"LINE DATA PRINT REQRD. YES=1 NO = 0"; PP
IF PP=0 THEN 501
PRINT "LINE DATA": PRINT
FOR I=1 TO LPQ:PRINT "LINE="; I: PRINT
FOR J=1 TO 7:PRINT Y(I,J);:NEXT J:PRINT: NEXT I
501 FOR I=1 TO LPQ:Z=Y(I,3)*Y(I,3)+ Y(I,4)*Y(I,4)
Y(I,3)=Y(I,3)/Z:Y(I,4)= -Y(I,4)/Z: NEXT I
INPUT "LINE ADMITTANCE FILE YA.I"; A$
CREATE A$ AS 1: FOR I=1 TO LPQ:FOR J=1 TO 7
PRINT #1 ; Y(I,J): NEXT J,I: CLOSE 1
FOR J=1 TO NB:FOR I=1 TO NB: YBUSR(I,J ) = 0
YBUSI(I,J) = 0: NEXT I,J
FOR I=1 TO NB: SUMA=0: SUMB=0: FOR J=1 TO LPQ
IF ( Y( J,1) - I )<>0  THEN 10
```

```
IF Y(J,7)<>1  THEN 10
SUMA=SUMA+Y(J,3)+Y(J,5):SUMB=SUMB+Y(J,4)+Y(J,6)
10 NEXT J
FOR J = 1 TO LPQ
IF ( Y(J,2) -I)<>0  THEN  11
IF Y( J,7 )<>1  THEN  11
SUMA=SUMA+Y(J,3)+Y(J,5):  SUMB=SUMB+ Y(J,4)+Y(J,6)
11 NEXT  J
YBUSR  (I,I)= SUMA:YBUSI(I,I)= SUMB
IF I = NB THEN  9
FOR  J=1 TO  LPQ
IF ( Y(J,1) -I)<>0  THEN 21
IF Y( J,7 )<>1  THEN  21
II=I+1: FOR K=II  TO NB
IF ( Y(J,2) - K)<>0  THEN  41
YBUSR  (I,K)= -Y(J,3): YBUSR(K,I)  = YBUSR(I,K)
YBUSI(I,K)= -Y(J,4): YBUSI( K,I)= YBUSI(I,K)
41 NEXT  K
21 NEXT  J
9  NEXT  I
FOR I=1 TO NB: FOR  J=1 TO  LPQ
IF( Y(J,1) -I)<>0  THEN 101
IF Y(J,7) = 1 THEN  101
YBUSI(I,I)= YBUSI(I,I)+Y(J,4)/(Y(J,7)*Y( J,7))
101 NEXT  J
NEXT  I
```

```
49: FOR I=1 TO NB: FOR J=1  TO  LPQ
50: IF (Y(J,2) -I)<>0  THEN  104
51: IF Y(J,7) =1  THEN  104
52: YBUSI ( I,I ) = YBUSI ( I,I ) + Y ( J,4 )
53: 104  NEXT  J
54: NEXT  I
55: FOR  I=1 TO NB: FOR J=1  TO LPQ
56: IF  ( Y( J,1 ) - I)<>0  THEN  102
57: IF Y ( J,7 ) = 1 THEN  102
58: FOR K = 1 TO NB
59: IF (Y ( J,2 ) -K)<>0  THEN  103
60: YBUSI (I,K)= -Y (J,4)/Y (J,7):YBUSI (K,I)= YBUSI (I,K)
61: 103 NEXT  K
62: 102 NEXT  J
63: NEXT  I
64: INPUT "IS ANY SHUNT CAP. YES = 1, NO= 0 ="; B
65: IF B = 0  THEN  105
66: 106  INPUT "CONNECTED TO WHICH BUS & VALUE="; BN,CS
67: YBUSI (BN,BN) = YBUSI ( BN,BN ) + CS
68: INPUT "ANY OTHER BUS WITH SHUNT YES = 1, NO=0="; BB
69: IF BB=0 THEN 105
70: GO TO 106
71: 105 INPUT "BUS ADMITTANCE MATRIX PRINT? YES=1 NO=0"; PP
72: IF PP = 0 THEN  502
73: PRINT "BUS ADMITTANCE MATRIX": PRINT
74: PRINT "R - COMPONENTS": PRINT
```

```
FOR I=1 TO NB : PRINT "ROW="; I : FOR J=1  TO NB
PRINT YBUSR( I,J ); :NEXT J: PRINT : NEXT  I
PRINT "J - COMPONENTS": PRINT
FOR I=1 TO NB : PRINT "ROW=";I: FOR J=1 TO NB
PRINT YBUSI  (I,J); : NEXT J: PRINT : NEXT I: PRINT
FOR I=1 TO NB: PRINT "ROW=";I: FOR J=1 TO NB
PRINT YBUSI (I,J),: NEXT J: PRINT: NEXT I: PRINT
502 INPUT "YBUSR FILE NAME YR"; A$ : CREATE A$ AS 1
FOR I=1 TO NB: FOR J=1 TO NB: PRINT #1;  YBUSR( I,J )
NEXT J: NEXT I: CLOSE 1
INPUT "YBUSI FILE NAME YI"; B$:CREATE B$ AS 2:FOR I=1 TO NB
FOR J=1 TO NB:PRINT #2; YBSI (I,J):NEXT J,I:CLOSE 2
FOR I=1 TO NB:FOR J=1 TO NB:AX(I,J)= YBUSI (I,J):NEXT J,I
FOR I=1 TO NB : FOR J=1  TO NB
IF AX( I,J ) = 0 THEN 200
AX( I,J ) = 1
200 NEXT  J
NEXT  I
INPUT "FILE NAME AXIJ="; C$
CREATE C$ AS 2: FOR I=1 TO NB: FOR J=1 TO NB
PRINT #2; AX( I,J ); NEXT J,I: CLOSE 2
PRINT "MATRIX AX( I,J )"
FOR I=1 TO NB: FOR J=1 TO NB: PRINT AX( I,J );
NEXT J:PRINT : NEXT I: PRINT : N=NB
DIM ORDER (N),A( N,N ), ADJ (N), TA(N), OPD(N)
FOR I=1 TO N: FOR J=1  TO  N
```

```
A( I,J ) = AX( I,J ) : NEXT J,I
FOR I = 1 TO  N: P = 0 : VA ( I ) = 0
FOR J =1  TO  N
IF I = J  THEN  610
IF A( I,J ) = 0  THEN  610
P = P + 1 : ADJ ( P ) = J
610 NEXT  J
IF P = 1 THEN  30
FOR L = 1 TO P - 1 : U = ADJ ( L )
FOR S = L+ 1 TO  P:  V = ADJ ( S )
IF A( U,V )<>0  THEN  20
VA  ( I  ) = VA  ( I  ) + 1
20  NEXT  S,L
30  NEXT I:  STAGE = 0:  VALANCY = 0
35 STAGE = STAGE + 1
IF STAGE = N+ 1  THEN 110
SMALL = N
FOR I = 1  TO  N
IF A( I,I ) = 0  THEN  40
IF  SMALL  = VA(I) THEN  40
SMALL = VA(I)
RE = I
40 NEXT  I
ORDER ( STAGE ) = RE
VALANCY = VALANCY + SMALL
FOR I = 1  TO  N
```

```
A( I,RE ) = 0
NEXT  I
P = 0
FOR  J = 1  TO  N
IF A( RE,J ) = 0  THEN  50
P = P+ 1 : ADJ( P ) = J
P1 = P
50 NEXT  J
IF P = 1  THEN  75
FOR  I = 1  TO  N
IF A( I,I ) = 0  THEN  70
FOR  L = 1  TO P1 - 1:  U = ADJ( L )
FOR S = L+1  TO P1 : V = ADJ( S )
IF I = U THEN 65
IF I = V THEN 60
IF A  ( I,U ) = 0  THEN  65
IF A( I,V ) = 0  THEN  60
P = P+1: ADJ( P ) = I
60 NEXT  S
65 NEXT  L
70 NEXT  I
FOR I = 1 TO P1 - 1 : U = ADJ( I )
FOR J = I +1 TO P1: V= ADJ( J )
A( U,V ) = 1: A( V,U ) = 1
NEXT J,I
75 FOR I = 1 TO P: U = ADJ( I)
```

```
C = 0 : VA( U ) = 0
FOR J = 1  TO  N
IF U = J  THEN  80
IF A( U,J ) = 0  THEN  80
C = C + 1 : OFD( C) = J
80  NEXT  J
IF  C = 1 THEN  100
FOR L = 1  TO  C - 1: K = OFD( L )
FOR S = L+ 1  TO C:M = OFD( S )
IF A( K,M )  0  THEN  90
VA( U ) = VA( U ) + 1
90 NEXT S,L
NEXT I
100 GO TO 35
110 PRINT "ORDER"
FOR I = 1  TO  N
PRINT ORDER ( I ) ;
NEXT  I
PRINT
INPUT "ORDERED SEQUENCE FILE ORD.I"; F$
CREATE F$ AS 6: FOR I = 1 TO NB:PRINT #6; ORDER ( I )
NEXT I : CLOSE 6
PRINT "TOTAL VALANCY= "; VALANCY
PRINT
DIM OYBUSR (NB,NB), OYBUSI ( NB, NB )
INPUT "ORDERED BUS PRINT REQRD. YES=1 NO=0";PP
```

```
IF PP = 0  THEN  503
FOR I=1 TO NB:NN=ORDER(I):FOR J=1 TO NB
NM=ORDER(J):OYBUSR(I,J)=YBUSR(NN,NM):OYBUSI(I,J)=YBUSI(NN,NM)
NEXT  J,I
PRINT "ORDERED R - BUS": PRINT:FOR I=1 TO NB
PRINT "ROW=";I:PRINT:FOR J=1 TO NB:PRINT OYBUSR(I,J);:NEXT J
PRINT : NEXT  I
PRINT "J-BUS": PRINT :  FOR I=1 TO NB
PRINT "ROW="; I: PRINT
FOR J=1 TO NB:PRINT OYBUSI(I,J);:NEXT J:PRINT:NEXT I
INPUT "ORDERED BUS FILE REQRD.? YES = 1 NO=0";PB
IF PB = 0  THEN  503
INPUT "OYBUSR FILE NAME = OYR"; C$
CREATE C$ AS 3:FOR I=1 TO NB:FOR J=1 TO NB
PRINT #  3; OYBUSR( I,J ): NEXT J,I: CLOSE 3
INPUT "OYBUSI FILE NAME = OYI"; D$
CREATE D$ AS 4: FOR I = 1  TO  NB
FOR J=1 TO NB:PRINT #4; OYBUSI( I,J ): NEXT J,I: CLOSE 4
503 END
NO ERRORS DETECTED
CONSTANT AREA :         8
CODE SIZE  :         5996
DATA STMT AREA :        0
VARIABLE AREA  :      384
A
```

A5.2 OPTIMAL ORDERING OF NODES BY DYNAMIC PROGRAMMING ALGORITHM

```
COMPILE DORDEL $E

HIBASIC COMPILER V2.0

  1: REM DYNAMIC PROGRAMMING ORDERING ALGORITHM
  2: LPRINTER WIDTH 80
  3: INPUT "NO OF BUSES"; N
  4: DIM AX( N,N )
  5: INPUT "AXIJ FILE NAME"; A$
  6: OPEN A$ AS 1
  7: FOR I = 1 TO N:FOR  J=1  TO  N
  8: READ #1; AX( I,J ):NEXT J,I: CLOSE  1
  9: DIM ORDER(N),A(N,N),NODE(N),VALANCY(N),CV(N,N),LROW(N),OPD(N)
 10: FOR STAGE = 1  TO  N
 11: IF STAGE = 1  THEN  120
 12: FOR ST = 1 TO N: CHECK = 0
 13: IF CV(ST,STAGE - 1)= -N THEN  106
 14: NODE (STAGE) =  ST: TOV=N * N: PROW = 0
 15: FOR PR = 1  TO  N
 16: IF PR = NODE (STAGE) THEN 100
 17: NODE ( STAGE - 1 ) = PR
 18: IF STAGE = 2 THEN 10
 19: IF CV( PR,STAGE - 1)= -N  THEN 100
 20: FOR K = 2 TO STAGE - 1:V = STAGE - K
 21: U = NODE ( V + 1 )
 22: NOD = NODE (V+1) - CV( U,V+1 )
 23: IF NOD = NODE (STAGE) THEN 100
```

```
NODE (V) = NOD: NEXT K
10 FOR J=1 TO N: FOR I=1 TO N:A( I,J ) = AX( I,J )
NEXT I,J: VA=0: FOR K=1 TO STAGE
R= NODE(K):P=0:FOR C=1 TO N: FOR I=1 TO K
RE = NODE ( I )
IF C = RE THEN 20
NEXT I
IF A ( R,C ) = 0 THEN 20
P = P+1 : OFD(P) = C
20 NEXT C
IF P<= 1 THEN 35
FOR L = 1 TO P - 1: FOR M = L+1 TO P
I = OFD( L ): J = OFD ( M )
IF A ( I,J )<>0 THEN 30
VA = VA+1:A( I,J ) = 1:A( J,I ) = 1
30 NEXT M,L
35 NEXT K
CHECK = CHECK + 1
IF TOV = VA THEN 100
TOV = VA : PROW = PR
100 NEXT PR
IF CHECK<>0 THEN 106
VALANCY (ST)= -1:CV(ST,STAGE) = -N: GO TO 110
106 VALANCY (ST)=TOV:CV(ST,STAGE)=ST-PROW
GO TO 110
106 CV( ST, STAGE ) = -N
```

```
110 NEXT ST:PRINT "STAGE";STAGE:PRINT:FOR I=1 TO N
PRINT "VALANCY"; VALANCY (I);"CV"; CV(I,STAGE)
NEXT I: PRINT
120 NEXT STAGE
SMALL = N * N: P=0
FOR I=1 TO N
IF VALANCY ( I ) = -1 THEN 140
IF SMALL <= VALANCY ( I ) THEN 140
SMALL = VALANCY ( I )
140 NEXT I: FOR I = 1 TO N
IF SMALL <>VALANCY ( I ) THEN 150
P = P+1: LROW ( P ) = I
150 NEXT I
FOR I = 1 TO P
ORDER ( N ) = LROW( I ): FOR J = 1 TO N - 1
C = N -J: RO=ORDER (C+1): ORDER (C)=ORDER(C+1) - CV(RO,C+1)
NEXT J: PRINT "ORDER";I: PRINT
FOR J = 1 TO N: PRINT ORDER ( J );: NEXT J
PRINT : NEXT I: PRINT
END
```

NO ERRORS DETECTED

CONSTANT AREA : 8

CODE SIZE : 1819

DATA STMT AREA : 0

VARIABLE AREA : 248

A

A6.3 GAUSS SEIDEL LOAD FLOW WITH OPTIMALLY ORDERED NODES

COMPILE OPTLF $E

MIBASIC COMPILER V2.0

```
LPRINTER WIDTH 80
INPUT "NO. OF BUSES & NO. OF LINES"; NB,LPQ
DIM ORDER (NB),Y(LPQ,9),YBUSR(NB,NB),YBUSI(NB,NB)
INPUT "YBUSR FILE NAME = OYR.I";A$
OPEN A$ AS 1: FOR I=1 TO NB: FOR J=1 TO NB
READ #1; YBUSR(I,J): NEXT J,I: CLOSE 1
INPUT "YBUSI FILE NAME = OYI.I"; B$
OPEN B$ AS 2: FOR I=1 TO NB: FOR J=1 TO NB
READ #2; YBUSI (I,J): NEXT J,I: CLOSE 2
INPUT "YA.I FILE NAME"; A$
OPEN A$ AS 1: FOR I=1 TO LPQ: FOR J=1 TO 7
READ #1; Y(I,J):  NEXT J,I: CLOSE 1
INPUT "ORDER FILE NAME ORD.I"; A$: OPEN A$ AS 1
FOR I=1 TO NB:READ # 1; ORDER (I): NEXT I: CLOSE 1
FOR I=1 TO LPQ: FOR J=1 TO NB
IF Y(I,1)<>ORDER(J) THEN 420
JJ=Y(I,1):Y(I,1)= J:Y(I,2)= JJ:GO TO 7
420 NEXT J
7 FOR J=1  TO NB
IF Y(I,2)<>ORDER(J) THEN 421
```

```
JJ=Y(I,2): Y(I,8)= J:Y(I,9)= JJ: GO TO 8
421 NEXT J
8 NEXT I
INPUT "LINE ADMITTANCE PRINT ? YES = 1 NO = 0"; BB
IF BB = 0 THEN 422
PRINT "LINE ADMITTANCE": PRINT
PRINT "OP OQ YR YI LCR LCI TTC P Q": PRINT
FOR I= 1 TO LPQ: FOR J=1 TO 9: PRINT Y(I,J);
NEXT J: PRINT: NEXT I
422 INPUT "SLACK BUS = J1="; J1
DIM EBUSR(NB,2), EBUSI(NB,2), P(NB,4),PA(NB),PR(NB)
DIM LPR(NB, LPI(NB), KLP(NB), KLI(NB)
DIM KLPR(NB,NB),KLPI(NB,NB),EBUSRA(NB,2),EBUSIA(NB,2)
FOR I=1 TO NB
IF I=J1 THEN 54
EBUSR(I,1) = 1: EBUSI( I,1 ) = 0
EBUSRA(I,1)= 1: EBUSIA(I,1) = 0
54 NEXT I
INPUT "SLACK BUS ACTIVE & REACTIVE VOLTAGES="; VSVR,VSVI
EBUSR(J1,1)= VSVR: EBUSI( J1,1 ) = VSVI
PRINT "PARAMETERS OF VOLTAGE EQUATIONS": PRINT
INPUT "GENERATION & LOAD POWER FILE = GL ="; B$
OPEN B$ AS 2:FOR I=1 TO NB: FOR J=1 TO 4
READ #2; P(I,J): NEXT J,I: CLOSE 2
INPUT "G - L PRINT ? YES = 1 NO = 0"; BB
```

```
46: IF BB = 0 THEN 2
47: PRINT"GENERATION-ACTIVE & REACTIVE:LOAD-ACTIVE & REACTIVE"
48: FOR I=1 TO NB: FOR J=1 TO 4: PRINT P(I,J),: NEXT J,I
49: 2 INPUT "MVA BASE"; BASE
50: FOR I = 1  TO NB
51: PA(I)=(P(I,1)-P(I,3))/BASE: PR(I)= - (P(I,2)-P(I,4))/BASE
52: D=YBUSR(I,I) * YBUSR(I,I)+ YBUSI(I,I) * YBUSI(I,I)
53: LPR(I)= YBUSR(I,I)/D:LPI(I)= -YBUSI(I,I)/D
54: NEXT  I
55: FOR I = 1  TO NB
56: IF I = J1  THEN 67
57: KLR(I)= PA(I) * LPR(I) + PR(I) * LPI(I)
58: KLI(I)= PA(I) * LPI(I) - PR(I) * LPR(I)
59: 67 NEXT I
60: KLR( J1 ) = 0: KLI( J1 ) = 0
61: INPUT " BUS PAR. PRINT ? YES = 1 NO = 0"; BB
62: IF BB = 0  THEN  3
63: PRINT "BUS PARAMETERS": PRINT
64: FOR I=1 TO NB: PRINT "BUS=" ; ORDER (I): PRINT
65: PRINT "KLR(I),KLI(I)=";KLR(I),KLI(I): PRINT : NEXT I
66: 3 FOR I = 1 TO NB: FOR J = 1  TO NB
67: KLPR( I,J ) = 0: KLPI( I,J ) = 0: NEXT J,I
68: FOR I = 1  TO NB
69: FOR J = 1  TO NB
70: IF J = I  THEN  60
```

```
KLPR(I,J) =YBUSR(I,J)* LPR(I) -YBUSI(I,J)* LPI(I)
KLPI(I,J) =YBUSR(I,J)* LPI(I) +YBUSI(I,J)* LPR(I)
60 NEXT J
NEXT I
INPUT "LINE PAR.PRINT ? YES = 1 NO =0";BB
IF BB = 0  THEN  4
PRINT " LINE PARAMETERS": PRINT
FOR I=1 TO NB: PRINT "ROW=";ORDER(I): FOR J=1 TO NB
PRINT KLPR(I,J), KLPI(I,J) : NEXT J
PRINT : NEXT I: PRINT
4 DIM DLR(NB,2),DLI(NB,2),EBUSCIA(NB,2)
DIM DLRA(NB,2),DLIA(NB,2)
REM EBUSCIA STANDS FOR CONJUGATE FOR VOLTAGE
K=1: FOR I=1 TO NB: EBUSRA(I,1)= EBUSR(I,1)
EBUSIA(I,1) = EBUSI(I,1): NEXT I
INPUT "ACCLERATION FACTOR ALPHA="; ALPHA
INPUT "TOLERANCE = TL = "; TL
INPUT "HOW MANY BUSES ARE VOLTAGE CONTROLLED ?"; L
DIM PV( L,4 )
INPUT "VCB.I FILE NAME"; D$
OPEN D$ AS 6: FOR I=1 TO L: FOR J=1 TO 4
READ # 6; PV(I,J):NEXT J,I:CLOSE  6 : FOR I=1  TO L
LL = PV(I,1): EBUSR( LL,K )= PV( I,2 )
EBUSI( LL,K ) = 0 : NEXT  I
80 EBUSRA ( J1,K+1) = EBUSRA( J1,K )
EBUSIA (J1, K+1) = EBUSIA ( J1, K )
```

```
97: EBUSR ( J1,K+1)= EBUSR ( J1,K )
98: EBUSI ( J1,K+1 )= EBUSI ( J1,K )
99: GOSUB 107
100: FOR I = 1  TO  NB
101: IF I = J1  THEN  61
102: DLRA ( I, 1 ) = 0: DLIA ( I,1 ) = 0
103: DLR ( I,1 ) = 0 : DLI ( I,1 ) = 0
104: 61 NEXT  I
105: FOR  I = 1  TO  NB
106: IF  I = J1  THEN  90
107: EBUSCIA ( I,K ) = -1.0* EBUSIA ( I, K )
108: 90  NEXT  I
109: FOR I = 1  TO  NB
110: IF  I = J1  THEN  62
111: D=EBUSRA(I,K)* EBUSRA(I,K)+ EBUSCIA(I,K)* EBUSCIA(I,K)
112: ER1 =(KLR(I)* EBUSRA(I,K)+ KLI(I)* EBUSCIA(I,K))/D
113: EI1 =(KLI(I)* EBUSRA(I,K)- KLR(I)* EBUSCIA(I,K))/D
114: IF  I=1  THEN  64
115: ER2 = 0: EI2=0: FOR  J=1  TO  I-1
116: ER2=ER2+KLPR(I,J)* EBUSRA(J,K+1)-KLPI(I,J)* EBUSIA(J,K+1)
117: EI2=EI2+KLPI(I,J)*EBUSRA(J,K+1)+KLPR(I,J)* EBUSIA(J,K+1)
118: NEXT  J
119: 64 ER3=0
120: EI3 = 0
121: IF I=NB THEN 700
122: FOR  J = I+ 1  TO  NB
```

```
ER3=ER3+KLPR(I,J)* EBUSRA(J,K)- KLPI(I,J)* EBUSIA(J,K)
EI3=EI3+KLPI(I,J)* EBUSRA(J,K)+ KLPR(I,J)* EBUSIA(J,K)
NEXT  J
700 IF  I = 1  THEN  111
GO TO 112
111 ER2 = 0
EI2 = 0
112 EBUSR( I,K+1 ) = ER1 - ER2 - ER3
EBUSI  ( I,K+1 ) = EI1 - EI2 - EI3
EBUSRA  (I,K+1)=EBUSRA(I,K)+ALPHA*(EBUSR(I,K+1)-EBUSRA(I,K))
EBUSIA  (I,K+1)=EBUSIA(I,K)+ALPHA*(EBUSI(I,K+1)-EBUSIA(I,K))
62 NEXT  I
PRINT "ITERATION ITN="; ITN: PRINT
INPUT "ITERATION  PRINT REQD.? YES = 1 NO = 0"; BB
IF BB = 0  THEN  5
PRINT "BUS VOLTAGES & ACCELERATED BUS VOLTAGES": PRINT
FOR I=1 TO NB:PRINT "BUS NO.="; ORDER  (I): PRINT
PRINT EBUSR(I,K+1),EBUSI(I,K+1),EBUSRA(I,K+1),EBUSIA(I,K+1)
NEXT I : PRINT
5 FOR I = 1 TO NB
IF I = J1  THEN  66
DLR( I,K+1) = EBUSR (I,K+1) - EBUSR( I,K )
DLRA(I,K+1) = EBUSRA(I,K+1) - EBUSRA(I,K)
DLI (I,K+1) = EBUSI ( I,K+1) - EBUSI ( I,K )
DLIA(I,K+1) = EBUSIA(I,K+1) - EBUSIA(I,K)
```

```
66 NEXT  I
DMAXR = DLR( 1,K+1 ): DMAXI = DLI ( 1,K+1 )
IF DMAXR < 0  THEN  701
GO TO 702
701 DMAXR = - DMAXR
702 IF DMAXI < 0  THEN 703
GO TO 704
703 DMAXI = - DMAXI
704 FOR I = 2  TO NB
IF  I = J1  THEN  70
IF  DLR ( I,K+1 )< 0  THEN  705
DEL = DLR  ( I,K+1 )
GO TO 706
705 DEL = - DLR  ( I,K+1 )
706 IF  ( DMAXR - DEL  )> 0  THEN  70
DMAXR = DEL
70 NEXT  I
FOR I = 2 TO NB
IF I = J1  THEN  73
IF DLI ( I,K+1 )< 0  THEN  707
DELL = DLI  ( I,K+1 )
GO TO 708
707 DELL = - DLI  ( I,K+ 1 )
708 IF  ( DMAXI - DELL )> 0  THEN  73
DMAXI = DELL
```

```
73 NEXT  I
IF ( DMAXR - DMAXI )> 0  THEN  68
DL  =  DMAXI
GO TO 71
68  DL = DMAXR
71 IF ( DL - TL )< 0  THEN  75
ITN = ITN + 1
FOR  I = 1  TO  NB
IF  I = J1  THEN  88
EBUSR(I,K)=EBUSR(I,K+1): EBUSI(I,K)= EBUSI(I,K+1)
EBUSRA(I,K)=EBUSRA(I,K+1): EBUSIA(I,K)=EBUSIA(I,K+1)
88 NEXT  I
GO TO 80
75 K1 = K+1
PRINT "LINEFLOWS & POWER SL BUS":PRINT:PRINT
DIM PQ( LPQ,6), QP( LPQ,6 )
FOR K = 1 TO NB : FOR I = 1  TO  LPQ
IF Y( I,1 )<>K  THEN 201
J = Y  ( I,2 )
P1=EBUSRA(K,K1)* (EBUSRA(K,K1) -  EBUSRA(J,K1))
P2= - EBUSCIA(K,K1)*(EBUSIA(K,K1) - EBUSIA( J,K1 ))
P3= EBUSRA(K,K1)* (EBUSIA(K,K1) -  EBUSIA(J,K1))
P4= EBUSCIA(K,K1)*(EBUSRA(K,K1) -  EBUSRA(J,K1))
P1 = P1 + P2 : P2 = P3+PA
PW = ( P1 *Y( I,3) - P2 *Y( I,4 )) * BASE
Q3=EBUSRA(K,K1)*EBUSRA(K,K1):Q4=EBUSIA(K,K1)*EBUSIA(K,K1)
```

```
Q3 = ( Q3 + Q4 ) * Y ( I,6 )
Q =(P1* Y(I,4)+P2 * Y(I,3) +Q3) * BASE
PQ(I,5)=Y(I,8):PQ(I,6)=Y(I,9):PQ(I,1)=Y(I,1):PQ(I,2)=Y(I,2)
PQ(I,3)= PW : PQ (I,4) = - Q
201 NEXT I
NEXT K
PRINT " LINE POWER FLOW "
PRINT
FOR I=1 TO LPQ: PRINT "ROW=";I:PRINT
FOR K=1 TO 6:PRINT PQ(I,K),:NEXT K: PRINT: NEXT I: PRINT
PRINT "LINE POWER FLOW REVERSED" : PRINT
FOR K=1 TO NB: FOR I=1 TO LPQ
IF Y(I,2)<>K THEN 301
J = Y ( I,1 )
P1=EBUSRA(K,K1)*( EBUSRA(K,K1) - EBUSRA(J,K1))
P2= - EBUSCIA(K,K1) *(EBUSIA(K,K1)- EBUSIA(J,K1))
P3= EBUSRA(K,K1) *(EBUSIA(K,K1) - EBUSIA(J,K1))
P4=EBUSCIA(K,K1) *(EBUSRA(K,K1) - EBUSRA(J,K1))
P1=P1+P2:P2=P3+P4
PW=(P1* Y(I,3) - P2 * Y(I,4)) * BASE
Q3=EBUSRA(K,K1)* EBUSRA(K,K1):Q4=EBUSIA(K,K1)* EBUSIA(K,K1)
Q3= (Q3+Q4)* Y(I,6)
Q=(P1*Y(I,4)+P2* Y(I,3)+ Q3)* BASE
QP(I,1)=Y(I,2):QP(I,2)=Y(I,1):QP(I,3)=PW: QP(I,4)= -Q
QP(I,5) = Y(I,9):QP(I,6)= Y(I,8)
301 NEXT I
```

```
NEXT  K
FOR I = 1  TO LPQ
PRINT "ROW="; I:PRINT: FOR K=1 TO 6
PRINT QP(I,k),: NEXT K:PRINT: NEXT I: PRINT
SUMA=0; SUMB=0
FOR I=1  TO LPQ
IF PQ( I,1 )<>J1  THEN  401
SUMA=SUMA+PQ(I,3): SUMB=SUMB+PQ(I,4)
401 NEXT  I
PRINT "SLACK BUS ACTIVE POWER ="; SUMA: PRINT
PRINT "SLACK BUS REACTIVE POWER= "; SUMB : PRINT
DIM EMAG(NB),GAMMA (NB)
FOR I=1 TO NB:A=EBUSRA(I,K1)*EBUSRA(I,K1)+EBUSIA(I,K1)*EBUSIA(I,K1)
EMAG(I) = SQR(A)
GAMMA(I)= 1.0*(180/3.142)* ATN(EBUSIA(I,K1)/EBUSRA(I,K1))
PRINT "BUS NO. VOLTAGE MAGNITUDE ANGLE": PRINT
PRINT ORDER (I), EMAG(I), GAMMA (I)
PRINT : NEXT  I
GO TO 113
107 REM COMPUTATION OF VOLTAGE CONTROLLED BUSES
FOR II=1 TO L:LL=PV(II,1):QEMIN=PV(II,3)/BASE
QEMAX=PV(II,4)/BASE: VCBLL=PV(II,2)
ANGLLL=ATN (EBUSI (LL,K)/EBUSR(  LL,K ))
TETA= ANGLLL* ( 180/3.142 )
EBUSR(LL,K+1)=VCBLL*COS(ANGLLL): EBUSI (LL,K+1)=VCBLL*SIN(ANGLLL)
REM CALCULATION OF REACTIVE POWER AT BUS LL
```

```
QR=(EBUSR(LL,K+1)* EBUSR(LL,K+1)* YBUSI (LL,LL)
QR=QR+(EBUSI (LL,K+1)* EBUSI (LL,K+1)) * YBUSI (LL,LL)
SUM=0: FOR I=1  TO NB
IF  I = LL  THEN  108
A=EBUSR(I,K)* YBUSR(LL,I)+EBUSI (I,K)* YBUSI (LL,I)
A= A  * EBUSI  ( LL,  K+1  )
B=EBUSI (I,K)* YBUSR(LL,I)-EBUSR(I,K)* YBUSI (LL,I)
B= -B * EBUSR( LL,K+1  )
SUM = SUM + A + B
108 NEXT  I
QR= QR + SUM
A = ABS  ( QR  )
IF A  = QRMAX  THEN 109
A = QRMAX  :  GO TO 1111
109 IF A  = QRMIN THEN  110
A = QRMIN  :  GO  TO 1111
110 EBUSR  (LL,K)= EBUSR(LL,K+1): EBUSI (LL,K)=EBUSI (LL,K+1)
REM RECOMPUTE KLR & KLI
1111 QR = A
D=YBUSR(LL,LL)* YBUSR(LL,LL)+YBUSI (LL,LL) * YBUSI (LL,LL)
LPR(LL) = YBUSR(LL,LL)/D: LPI (LL)= -YBUSI (LL,LL)/D
KLR(LL) = PA(LL)* LPR(LL)+ QR * LPI (LL)
KLI (LL)= PA(LL)  * LPI (LL)- QR * LPR(LL)
```

```
  274: NEXT  II
  275: RETURN
  276: 113  STOP
  277: END
NO ERRORS DETECTED
CONSTANT AREA :        24
CODE SIZE  :         8653
DATA STMT AREA  :       0
VARIABLE AREA   :     632
A
```

B I B L I O G R A P H Y

B I B L I O G R A P H Y

[1] HO, Y.C., "Editorial - Is it application or is it experimental science ?", IEEE Transactions on Automatic Control, Vol. AC - 27, No. 6, page 1142, December 1982.

[2] Aström, K.J., and Eykhoff, P., "System Identification, A Survey," Automatica, Vol. 7, pp. 123 - 162, March, 1971.

[3] Beck, M.B., "System Identification, Estimation and Forecasting of Water Quality Part 1 : Theory "Working Paper, WP-79-31, April 1979, International Institute for Applied System Analysis : A - 2361 Laxenburg, Austria.

[4] Young, P.C., "A Recursive Approach to Time Series Analysis," Bulletin of the Institute of Mathematics and its Application, Vol. 10, pp. 209 - 224, 1974.

[5] Soderström, T., L. Ljung and I. Gustavsson, "A Theoretical Analysis of Recursive Identification Methods," Automatica, Vol. 14, pp.231-244, 1978.

[6] Ljung, L., "On Positive Real Transfer Functions and the Convergence of Some Recursive Schemes," IEEE Trans. Automatic Control, Vol. AC - 22, Aug. 1977.

[7] Ljung, L., "Analysis of Recursive Stochastic Algorithms," IEEE Trans. Automatic Control, Vol. AC - 22, Aug. 1977.

[8] ____________, "Convergence Analysis of Parameter Identification Methods," IEEE Trans., Vol. AC - 23, No. 5, 1978.

[9] ____________, "A Theoretical Analysis of Recursive Identification Algorithms," Automatica, Vol. 14, pp. 231 - 244, 1978.

[10] ____________, "Asymtotic Behaviour of Extended Kalman Filter as a Parameter Estimator for Linear Systems," IEEE Trans., Vol. AC - 24, No. 1, Feb. 1979, pp. 36 - 50.

[11] Young, P.C., "Optimisation in the Process of Noise — A Guided Tour," Report Number AS/RR (1975), Centre for Resource and Environmental Studies, Australian National University, Canberra.

[12] Chaudhuri, A.S., "System Analysis of Tea Industry : Identification, Modelling and Prediction," Thesis Submitted to the University of Calcutta - Dec. 1979.

[13] Eykhoff, P., System Identification - Parameter and State Estimation, Willey, London, 1974.

[14] Draper, N.R., and Smith, H., "Applied Regression Analysis," Willey, New York, 1966.

[15] Beck, M.B., "Real Time Control of Water Quality and Quantity," Research Memorandum, RM- 78- 19, April 1978, International Institute of Applied System Analysis, Laxenburg, Austria.

[16] Hastings - James, R. and Sage, M.W., "Recursive Generalized Least Square Procedures for On-Line Identification of Process Parameters," Proceedings of IEE, 116, pp. 2057-2062, 1969.

[17] Young, P.C., "Some Observations of Instrumental Variable Methods of Time Series Analysis," International Journal of Control, Vol. 23, pp. 593 - 612, 1976.

[18] ____________, "General Theory of Modelling for Badly Defined Systems," pp. 103 - 135, Modelling Identification and Control in Environmental Systems, Edited by G.C. Vansteenkiste, Amsterdom, North-Holland, 1978.

[19] Young, P.C., and Jakeman, A.J., "Refined Instrument Variable Methods of Recursivo Time Series Analysis Part I : Single Input, Single Output Systems," Report AS/R12, Centre for Resources and Envoronmental Studies, Australian National University, Canberra, Australia, 1978.

[20] Young, P.C., and Jakeman, A.J., "Refined Instrumental Variable Methods of Recursive Time Series Analysis Part II : Multivariable Systems," Report AS/R13, Centre for Research and Environmental Studies, Australian National University, Canberra, Australia, 1978.

[21] __________, "Refined Instrumental Variable Methods of Recursive Time Series Analysis Part III : Extensions," Report AS/R17, Centre for Resources and Environmental Studies, Australian National University, Canberra, Australia, 1978.

[22] Young, P.C., Jakeman, A.J., and McMurtrie, R.E., "An Instrumental Variable Method for Model Structure Identification," Report AS/R22, Centre for Resource and Environmental Studies, Australian National University, Canberra, Australia, 1978.

[23] Young, P.C., Shellswell, S.H., and Neethling, C.G., "A Recursive Approach to Time Series Analysis," Technical Report CUED/B - Control/TR16, University Engineering Department, Cambridge, U.K., 1971.

[24] Kashyap, R.L., "Estimation of Parameters in a Partially Whitened Representation of a Stochastic Process," IEEE Trans. Automatic Control, Vol. A.C.19, pp. 13 - 21, Feb. 1974.

[25] Kashyap, R.L., and Rao, A.R., "Stochastic Modelling of River Flow," IEEE Trans. Automatic Contorl, Vol. AC - 19, No. 6, December, 1974.

[26] __________, "Dynamic Stochastic Models from Empirical Data," Academic Press, New York, 1976.

[27] __________, "Real Time Recursive Prediction of River Flows," Automatica, Vol. 9, pp. 175-183, 1973.

[28] Whitehead, P., and Young, P.C., "A Dynamic - Stochastic Model for Water Quality in Part of the Bedford Ouse River System," Modelling and Simulation of Water Resources System, G.C. Vansteenkiste, (Ed.), North-Holland Publishing Company, 1975.

[29] __________, "The Bedford Ouse Study Dynamic Model," 2nd Semi-Annual Report to the Steering Group of the Great Ouse Associated Committee, 1974, Control Group Internal Note CN/74/1, University of Cambridge, U.K.

[30] Younng, P.C., Whitehead, P.G., and Bech, M.B., "The Bedford Ouse Study Dynamic Model," 1st Semi-Annual Report to the Steering Group of the Great Ouse Associated Committee, Control Group Internal Note, CN/73/4, 1973, University of Cambridge, U.K.

[31] Sengupta, S.K., and Chaudhuri, A.S., "Real Time Forecasting of Dissolved Oxygen Level of a Non-tidal River," Journal of the Environmental Engineering Division of the Institution of Engineers (India), Vol.62, Pt. EN 3, June 1982, pp. 127 - 130.

[32] Sengupta, S.K., Maulik, M.N. and Chaudhuri, A.S., "Forecasting of Daily Dissolved Oxygen Levels of a Non-tidal River by Dynamic Least Square Recursive Estimation Technique," Journal of the Environmental Engineering Division of the Institution of Engineers (India), Vol. 63, June 1983, pp. 132 - 136.

[33] __________, "Modelling of Bio-Chemical Oxygen Demand (B.O.D.) of a Non-tidal River by a Learning Algorithm and Recursive Instrument Variable Technique," accepted for publication to the Journal of the Environmental Engineering Division of the Institution of Engineers (India).

[34] Maulik, M.N., Sengupta, S.K. and Chaudhuri, A.S., "Real Time Forecasting of Daily Flow of a Non-tidal River," Journal of the Civil Engineering Division of the Institution of Engineers (India), Vol. 63, pt. Cl.3, November 1982, pp. 132 - 136.

[35] __________, "Real Time Estimation of River Flow During A Short Span Storm Period," Journal of the Civil Engineering Division of the Institution of Engineers (India), Vol. 63, pt. Cl.4, January 1983, pp. 213 - 218.

[36] Ivakhnenko, A. G., "Heuristic Self-Organisation of Problems of Engineering Cybernetics," Automatica, Vol. 6, pp. 207 - 219, 1970.

[37] Ivakhnenko, A. G., and Kappa, Yu. V., "Group Method of Data Handling for Solution of Various Interpolation Problems of Cybernetics," Second IFAC Symposium of Identification and Process Parameter Estimation, Prague, Paper 21, 1970.

[38] Ivakhnenko, A. G., "Polynomial Theory of Complex Systems," Transaction of Systems, Man and Cybernetics of Institution of Electrical and Electronics Engineers, IEEE, Vol. SMC 1, No. 4, October, 1971.

[39] __________, "Long Term Prediction and Control of Complex Processes," Tekhnika, Kiev, 1975 (Book in Russian).

[40] Volterra, V., "Theory of Functional and Integral and Integro-Differential Equations," London, Blackie, 1930.

[41] Wiener, N., "Non Linear Problems in Random Theory," New York, Willey, 1958.

[42] Gabor, D., Wilby, W.P.L., and Wood Cock, R., "A Universal Non-linear Filter Predictor and Simulator which Optimizes itself by a Learning Process," Proceedings of the Institution of Electrical Engineers, London, IEE Proc., Vol. 108, Part B, pp. 422 - 433.

[43] Rosenblatt, F., "Principles of Neurodynamics, "Spartan Books, Washington, 1962.

[44] Aström, K.J., and Eykhoff, P., "System Identification, A Survey," Automatica, Vol. 7, pp. 123-162, March, 1971.

[45] Rosenblatt, F., "Perception : A Probabilistic Model for Information Storage and Organisation in the Brain," Psychological Review, Vol. 65, pp. 386 - 408, November, 1958.

[46] Ivakhnenko, A. G., "The Group Method of Data Handling a Rival of the Method of Stochastic Approximation," Soviat Automatic Control, No. 3, 1968.

[47] Ivakhnenko, A. G., et al., "Group Method of Data Handling in the Problem of Pattern Recognition and Decision Making," Soviet Automatic Control, No. 5, 1968.

[48] ____________, "Group Method of Data Handling in the Problem of Identification of Multiextremal Plant," Soviet Automatic Control No. 2, 1969.

[49] Box, G. E. P., "Use and Abuse of Regression," Technometrics, Vol. 8, pp. 625 - 629, Nov., 1966.

[50] Parks, P., and Pyatt, Y., "Controlling the Economy - An Exercise in Collaboration," Electron Power, Vol. 15, August, 1969, pp. 270 - 274.

[51] Ikeda, Saburo, et al., "Sequential GMDH Algorithm and its Application to River Flow Predictions," Institution of Electrical and Electronics Engineers, IEEE Transaction on Systems, Man and Cybernetics, Vol. SMC - 6, No. 7, July, 1976.

[52] ____________, "Non Linear Prediction Model of River Flow by Self-Organisation Method," International Journal of System Science, 1976.

[53] Tamura, H., and Konda, T., "Large Spatial Pattern Identification of Air Pollution by a Combined Model of Source Perceptor Matrix and Revised GMDH," IFAC Symposium of Environmental Systems Planning, Design and Control, Kyoto, Japan, Aug. 1 - 5, 1977.

[54] Duffy, J.J., and Franklin, M.A., "A Case Study of Environmental System Modelling with Group Method of Data Handling," Joint Automatic Control Conference, U.S.A., Proc. pp. 101 - 111, 1973.

[55] __________, "A Learning Identification Algorithm and its Application to an Environmental System," IEEE Trans. Systems, Man and Cybernetics, Vol. SMC - 5, No. 2, March, 1975.

[56] Mehra, R. K., "Group Method of Data Handling Algorithms (GMDH): Review and Experience," IEEE Conference on Decision and Control, December 7 - 9, 1977, New Orleans, U.S.A.

[57] Baier, W., and Williams, G.D.V., "Regional Wheat Yield Prediction from Weather Data in Canada," WMO Symposium on Agrometrology of the Wheat Crop, Braunschweig, Federal Republic of Germany, Oct. 1973, pp. 263 - 283.

[58] Mustafi, A., and Chaudhuri, A.S., "Use of Multilayer Group Method of Data Handling for Predicting Tea Crop Production," Journal of the Society of Indian Agricultural Statistics, Vol.XXXIII, No. 1, April, 1981.

[59] Maulik, M.N., Sen Gupta, S.K. and Chaudhuri, A.S., "Rainfall-run-off Model of Flows of River for A Short Span Storm Period by Learning Algorithm," Indian Journal of Power and River Valley Development, January - February, 1982.

[60] __________, "Prediction of Hourly Flows by Computer Aided Self-Organisation of Mathematical Models," 'Irrigation and Power' Journal, Vol. 40, No. 1, January, 1983.

[61] __________, "Forecasting of River Flow During Storm Periods by a Learning Identification Algorithm," Journal of the Civil Engineering Division of the Institution of Engineers, (India), Vol. 63, Pt. CL 1, July, 1982.

[62] Sen Gupta, S.K., Maulik, M.N. and Chaudhuri, A.S., "Learning Identification Dynamic Algorithms for Modelling of Dissolved Oxygen Incorporating Interacting Parameters," Journal of the Environmental Engineering Division of the Institution of Engineers (India), Vol. 64, February 1984.

[63] Ivakhnenko, A.G., "Present State of the Theory of Computer Aided Self-Organisation of Mathematical Model," Soviat Automatic Control, Vol. 8, September - October, 1975.

[64] Chaudhuri, A.S., "Medium-Term and Long-Term Prediction Model of Annual Tea Crop Production Using Heuristic Learning Algorithm," Journal of the Indian Society of Agricultural Statistics, Vol. XXXIII, No. 8, August, 1981.

[65] Maulik, M.N., Sen Gupta, S.K. and Chaudhuri, A.S., "Forecasting of Hourly River Flow Using Heuristic Learning Algorithm," Journal of the Civil Engineering Division of the Institution of Engineers (India), Vol. 63, Pt. CL 1, July, 1982.

[66] Sen Gupta, S.K., Maulik, M.N. and Chaudhuri, A.S., "Forecasting of Daily Dissolved Oxygen Levels of a Non-tidal River by Self-Organisation of Mathematical Models," Journal of the Environmental Engineering Division of the Institution of Engineers (India), Vol. 63, Journal - EN, June 1983.

[67] Stagg, G.W., and El-Abiad, A.H., "Computer Methods in Power System Analysis," - New York, Mc-Graw Hill, 1968.

[68] Schweppe, F.C. et.al, "Power System Static State Estimation, Pt. 1 : Exact Model," IEEE Transaction on Power Apparatus and Systems, Vol. PAS - 89, January, 1970, pp. 120 - 125.

[69] ________, "Power System Static-State Estimation, Part II Implementation," IEEE PAS Vol. PAS - 89, January 1970, pp. 125 - 130.

[70] Schweppe, F.C. et.al, "Power System Static-State Estimation, Part III Implementation," IEEE PAS Vol. PAS 89 January 1970, pp. 130 - 135.

[71] __________, "Static State Estimation in Electric Power Systems," IEEE Proceedings Vol. 62, pp. 972 - 982, July 1974.

[72] Dopazo, J.F. et.al, "State Calculation of Power Systems from Line Flow Measurements, Part II," IEEE PAS-91, No. 1, pp. 145 - 151, January/February 1972.

[73] Dopazo, J.F. et.al, "Implementation of the AEP Real-Time Monitoring System," IEEE PAS - 95, No. 5, pp. 1618 - 1629, September/October 1976.

[74] Allemovg, J.J. et.al, "A Fast and Reliable State Estimation Algorithm for AEP'S New Control Center," IEEE PAS - 101, No. 4, pp. 933 - 944, 1982.

[75] Arafeh, S.A., et.al, "Estimation Algorithms for Large-scale Power Systems," IEEE Trans. Power App. Syst., PAS - 98, pp. 1968 - 1977 Nov./Dec. 1979.

[76] Sen Gupta, S.K., and Chaudhuri, A.S., "Real Time Forecasting of Dissolved Oxygen Level of a Non-tidal River," Institution of Engineers (India), Vol. 62, pp. 127 - 130, June, 1982.

[77] Alvarez, C., and Albertos, P., "On-Line Observability Determination as a Further Result of State Estimation Algorithms," IEEE PAS - 101, pp. 757-774, April 1982.

[78] Pai, M.A., "Computer Techniques in Power System Analysis," Tata McGraw-Hill Publishing Company Limited, New Delhi, 1979.

[79] Irisani, G., Sasson, A.M. and Hedges, S.F., "An Optimal Ordering Algorithm for Sparse Matrix Application," IEEE Trans. Power Apparatus and Systems, Vol. PAS - 97, Nov./Dec. 1978.

[80] Duff, I.S., "A Survey of Sparse Matrix Research," Proc. IEEE, Vol. 65, No. 4, April, 1977.

[81] Tinney, W.F. and Meyer, W.S., "Solution of Large Sparse Systems by Ordered Triangular Factorization," IEEE Trans. Automatic Control, Vol. AC - 18, pp. 336 - 346, 1973.

[82] Tinney, W.F. and Walker, J.W., "Direct Solution of Sparse Network Equations by Optimally Ordered Triangular Factorization," Proc. IEEE, Vol.55, pp. 1801 - 1809, 1967.

[83] Ogbuobiri, E.C., Tinney, W.F., Walker, J.W., "Sparsity-Directed Decomposition for Gaussian Elimination on Matrices," IEEE Trans. Power Apparatus and Systems, Vol. PAS - 89, pp. 141 - 150, 1970.

[84] Tinney, W.F. and Sato, N., "Techniques for Exploiting the Sparsity of the Network Admittance Matrix," IEEE Trans. PAS - 82, 1963, pp.944 - 950.

[85] Stott, B. and Hobson, E., "Solution of Large Power System Networks by Ordered Eliminations ; A Comparison of Ordering Schemes," Proc. IEE (London), 1971, pp. 125 - 134.

[86] Bellman, R. and Dryfus, S., "Applied Dynamic Programming," Princeton Univ. Press, Princeton, N.J., 1962.

[87] Larson, R. E., "State Increment Dynamic Programming," American Elsevier Publishing Company, Inc., N.Y., 1968.

[88] Jhampati, J. and Chaudhuri, A. S., "On-Line Simulation of Hourly River Flows for Run-of-the River Hydro-electric Plant," send for publication in the Journal of Civil Engineering Division of the Institution of Engineers, India.

[89] Jhampati, J. and Chaudhuri, A. S., "Electrical Energy Consumption Model with Interacting Parameters by a Learning Identification Algorithm," send for publication in the Journal of Electrical Engineering Division of the Institution of Engineers, India.

[90] Jhampati, J. and Chaudhuri, A. S., "Medium Term and Long Term Prediction Models of Annual Installed Plant Capacity and Consumption of Electrical Energy by Computer-Aided Self-Organisation of Mathematical Models," send for publication in the Journal of Electrical Engineering Division of the Institution of Engineers, India.

[91] Jhampati, J. and Chaudhuri, A.S., "States Estimation of Electrical Power System by a Tracking Algorithm," send for publication in the Journal of Electrical Engineering Division of the Institution of Engineers, India.

[92] Jhampati, J., Sen Gupta, S.K. and Chaudhuri, A.S., "Gauss-Seidel Load Flow with Optimally Ordered Nodes by Dynamic Programming Algorithm," send for publication in the Journal of Electrical Engineering Division of the Institution of Engineers, India.

[93] Venikov, V.A., "Automatic Power System Control," Mir Publishers, Moscow, 1982, pp. 36 - 43.