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.

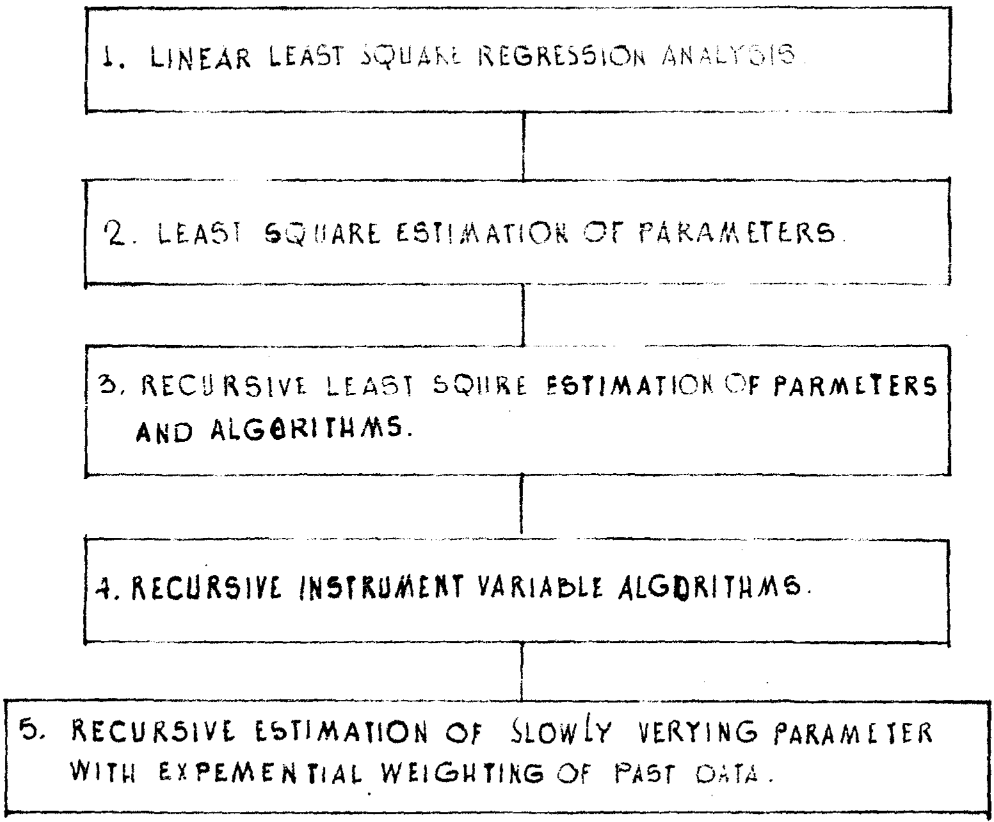

FIG. 2.1.0

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 [$i = 1,2,\ldots n$] of the input signal and y_i [$i = 1,2,\ldots n$] 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

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

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

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

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

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

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

and

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

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

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

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

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

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

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

$$\ldots \qquad (2.1.21)$$

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,

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

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 $E\left\{V(t_k)\ V(t_{k-1})\right\} \neq 0$ for all k which implies $E\left\{Y(t_{k-1})\ V(t_k)\right\} \neq 0$, 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

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

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

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

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

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

$$z(t_{k-1}) = \left[Y(t_{k-1}), Y(t_{k-2}), \ldots, U_1(t_{k-r_1}) \ldots\ldots, U_m(t_{k-rm-n}) \right] \qquad (2.1.34)$$

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

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

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

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

For minimization,

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

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

It follows from equation (2.1.36)

$$\sum_{j=1}^{k} z(t_{j-1})Y(t_j)+S^{-1}(t_o)a(t_o)= \sum_{j=1}^{k} z(t_{j-1})z^T(t_{j-1})+S^{-1}(t_o)a \qquad \ldots (2.1.37)$$

Let

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

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

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

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

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

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. [40]

Volterra series [41] introduced to non-linear system analysis by Wiener, 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 composite 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 be 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 properties, 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.

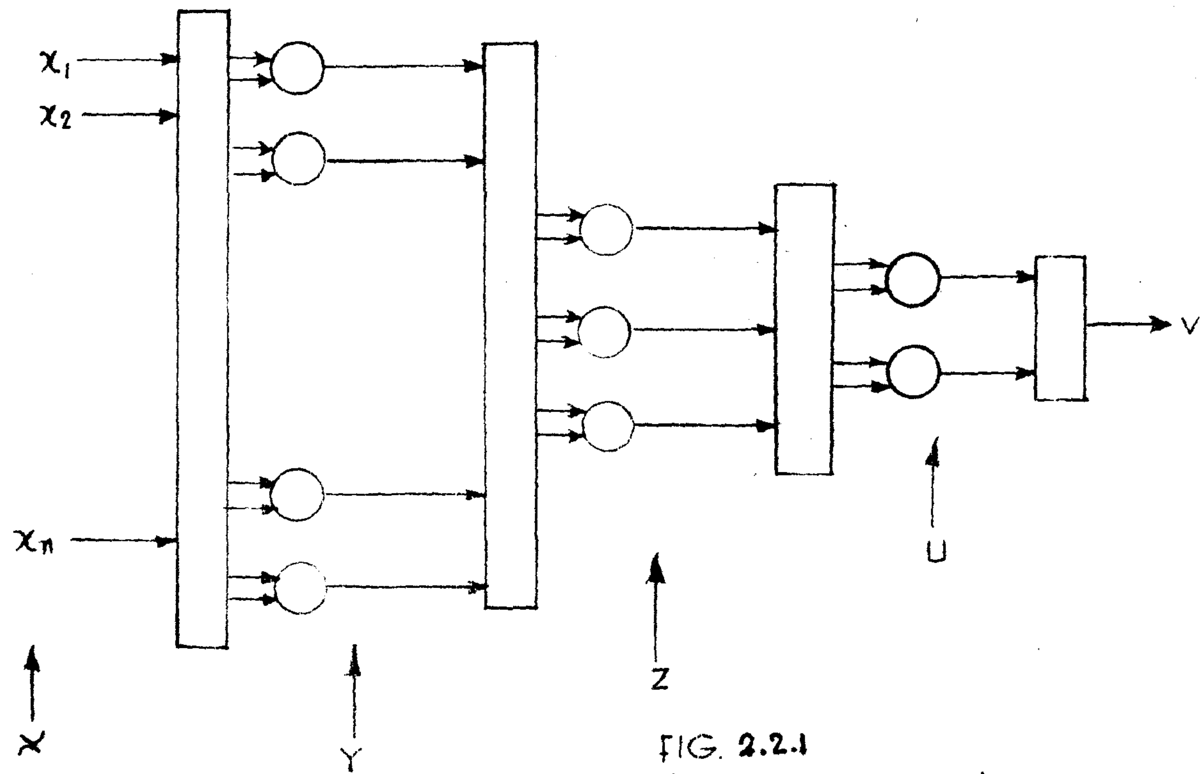

FIG. 2.2.1
Concept of feed forward transformation.

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 $\phi = XA$ where ϕ, X and A are Nx1, Nx6 and 6x1 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

$$\phi = f\ (x_1, x_2, x_3, \ldots, x_n) \tag{2.2.4}$$

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

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

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

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

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

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

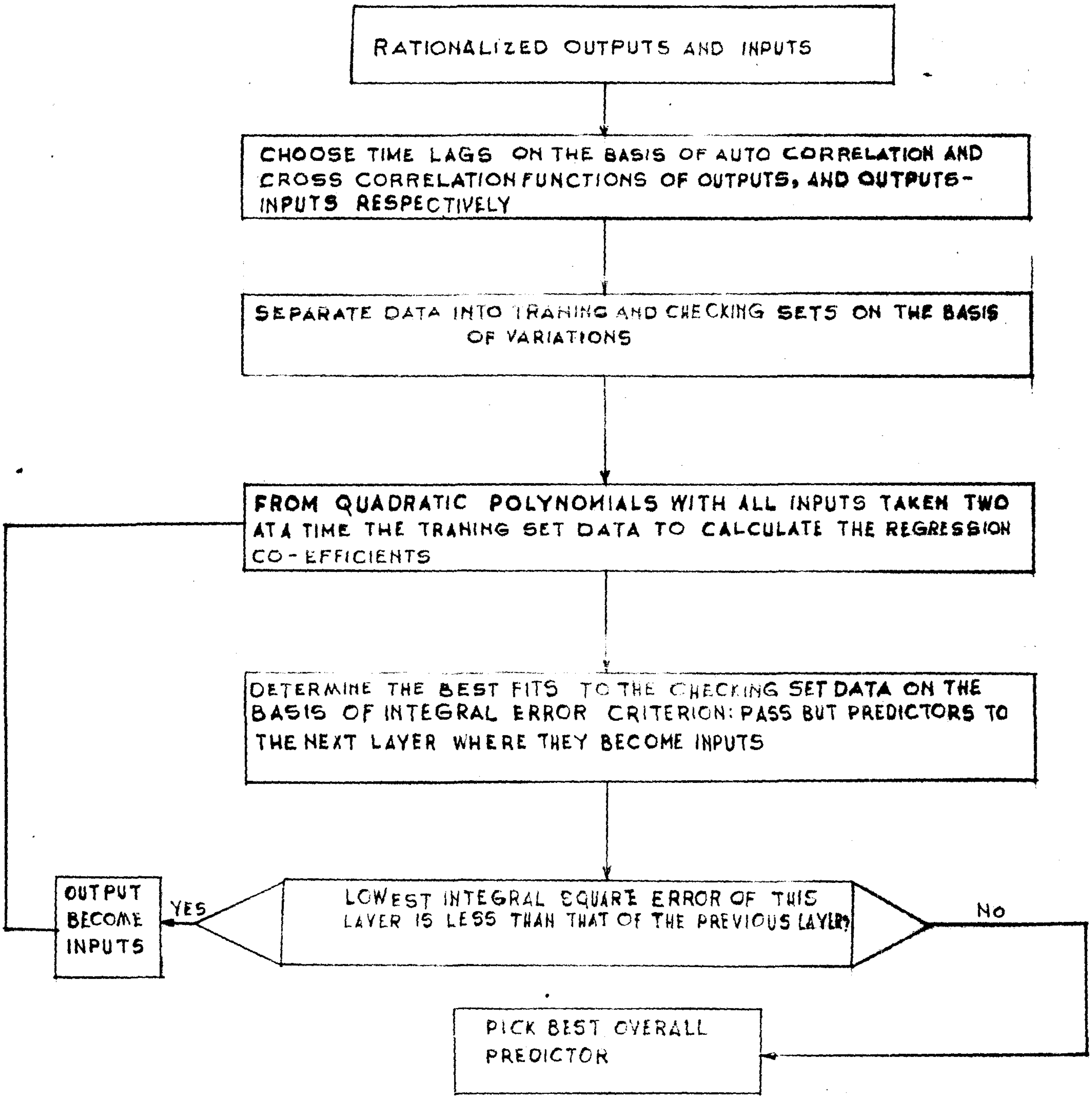

FIG. 2.2.2 FLOW CHART OF MULTILAYER GMDH ALGORITHMS.

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

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

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

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

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

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 Sawargi [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 Kondo [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 prediction 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 prediction 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 $i = 1,2,3,4,5\ ;\quad j = m,n,p,q,r$

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

$$y = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_3 + a_5x_1x_3 + a_6x_2x_3 + a_7x_4 + a_8x_1x_4 + a_9x_2x_4 + a_{10}x_3x_4 + a_{11}x_5 + a_{12}x_1x_5$$

$$+ a_{13}x_2x_5 + a_{14}x_3x_5 + a_{15}x_4x_5 + a_{16}x_6 + a_{17}x_1x_6 + a_{18}x_2x_6 + a_{19}x_3x_6 + a_{20}x_4x_6 + a_{21}x_5x_6 + a_{22}x_7 +$$

$$+ a_{23}x_1x_7 + a_{24}x_2x_7 + a_{25}x_3x_7 + a_{26}x_4x_7 + a_{27}x_5x_7 + a_{28}x_6x_7 + a_{29}x_8 + a_{30}x_1x_8 + a_{31}x_2x_8 + a_{32}x_3x_8 +$$

$$+ a_{33}x_4x_8 + a_{34}x_5x_8 + a_{35}x_6x_8 + a_{36}x_7x_8$$

x_1	x_2	x_1x_2	x_3	x_1x_3	x_3x_4	x_8
$y_1 = a_0 + a_1x_1$	$y_2 = a_0 + a_2x_2$	$y_4 = a_0 + a_3x_1x_2$	$y_8 = a_0 + a_4x_3$	$y_{16} = a_0 +$	$y_{512} =$	....
	$y_3 = a_1 + a_1x_1$	$y_5 = a_0 + a_1x_1 +$	$y_9 = a_0 + a_1x_1$	$+ a_5x_1x_3$	$a_0 + a_{10}x_3x_4$	
	$+ a_2x_2$	$+ a_3x_1x_2$	$+ a_4x_3$	$y_{17} = a_0 + \cdots$	$y_{513} = a_0 + a_1x_1$	
		$y_6 = a_0 + a_2x_2 +$	$y_{15} = a_0 + a_1x_1 +$		$+ a_{10}x_3x_4$	
		$+ a_3x_1x_2$	$a_2x_2 + a_3x_1x_2 +$		$y_{1023} = a_0 + a_1x_1 +$	$y_{2^{36}-1} = a_0 +$
		$y_7 = a_0 + a_1x_1 +$	$\cdots + a_4x_3$		$a_2x_2 + a_3x_1x_2 +$	$+ a_1x_1 + a_2x_2 +$
		$+ a_2x_2 + a_3x_1x_2$			$a_4x_3 + a_5x_1x_3$	$+ a_3x_1x_2 + \cdots$
					$+ \cdots + a_{10}x_3x_4$	$+ a_{36}x_7x_8$

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 [66] 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. Dopaso 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.