APPLICATION OF ANN IN INFLOW PREDICTION AND RESERVOIR OPERATION

A DISSERTATION

submitted in partial fulfilment of the requirements for the award of the degree

of

MASTER OF ENGINEERING

in

HYDROLOGY

By

ASHUTOSH DASH

UNESCO SPONSORED
INTERNATIONAL HYDROLOGY COURSE
UNIVERSITY OF ROORKEE
ROORKEE – 247 667 (INDIA)

DECEMBER, 1997

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in this dissertation entitled "***APPLICATION OF ANN IN INFLOW PREDICTION AND RESERVOIR OPERATION***" in partial fulfilment of the requirement for the award of the degree of Master of Engineering in Hydrology, submitted in the department of Hydrology of the University of Roorkee is an authentic record of my work done during the period from 16*th* July, 1997 to 7*th* December, 1997 under the supervision of Dr. D. K. Srivastava, Professor and Head, Department of Hydrology, University of Roorkee, Roorkee and Dr. S.K. Jain, scientist 'E', National Institute of Hydrology, Roorkee.

The matter embodied in this dissertation has not been submitted by me for the award of any other degree.

Ashutosh Dash

(ASHUTOSH DASH)

Candidate's Signature

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Jain 10/12/97

(S.K. Jain)

10.12.98

(D.K. Srivastava)

ACKNOWLEDGEMENT

I wish to express my deep sense of gratitude to **Dr. D.K. Srivastava Professor and Head**, Department of Hydrology, University of Roorkee, Roorkee and to **Dr. S.K. Jain, Scientist 'E'** , National Institute of Hydrology, Roorkee for their valuable guidance, encouragement and whole hearted co-operation in carrying out the study, without which it would not have been possible to complete the dissertation in time.

I wish to specially acknowledge Sri Ajay Kushwaha, IIIrd Year, BE,Electrical, U.O.R. for the continuous and untired help extended by him in debugging, designing and ornamenting the programs written in C and C++ for the present study.

I am deeply indebted to Sri P.K.Mishra, Chief Engineer, Designs, Bhubaneswar, Department of water Resources, Government of Orissa for his continuous inspiration and motivation to excel during the tenure of the course. I am extremely thankful to Dr. N. K. Goel, Assoc. Prof., Dept. of Hydrology, for the literature provide by him in the area of Stochastic time-series analysis and the pain he took in examining my results and suggesting modifications and improvements.

I also wish to express my sincere thanks to all other faculty members for their as-and-when required co-operation during the tenure of the course. Especially I am thankful to Dr. Himansu Joshi, because he introduced me to the area of ANN.

Useful co-operation and help received from my fellow trainee officers and friends outside the department are acknowledged with thanks.

November, 1997

Roorkee

(Ashutosh Dash)

SYNOPSIS

Artificial Neural Networks (ANNs) are new computing architectures in the area of artificial intelligence. The recent resurgence of interest in Artificial Neural Networks has its roots in the recognition that the brain performs computations in a different manner than do conventional digital computers. A human information processing system is composed of neurons switching at speeds about a million times slower than the computer gates. Yet humans are more efficient than computers at computationally complex tasks. Artificial Neural Network is the result of long standing effort for mimicking the computational capability of brain system.This derives its inspiration from two distinct yet related fields - Associationist psychology and Neuroscience.

Quite contrast to the conventional digital computers, these ANN based systems can acquire, store and utilize experiential knowledge. For this reason, the scope of its applicability is being explored in many disciplines including Hydrology. The present study aims at application of this Neural Network based computational paradigm in reservoir operation and inflow prediction.

Upper Indravati hydro-electric project, which is a large multi purpose Water Resources Project in the Nowrangpur and Kalahandi districts of Orissa has been selected as the problem area for this study. On completion, the project shall comprise of four dams and a combined reservoir of 110 km^2 area. The project has primarily two objectives: to provide irrigation to 1,28000 ha. of agricultural land and to provide 600 MW of electric power through four numbers of Francis turbines of 150 MW each.

The present study capitalises on 32 years of monthly inflow record of the project. Specifically the objective is to maximise total project benefits from hydro power generation, simultaneously aiming to minimise the irrigation deficit and water losses through spilling. This objective was aimed to be attained through the following steps.

1. The primary objective of maximising benefit from power generation was achieved through a DP model based on DDDP approach taking the entire period of 32 years as the optimisation horizon. The secondary objective of minimising irrigation deficit was taken care of by searching an appropriate loss function through trial and error procedure.

2. Reservoir operation policies are formulated through the conventional DPR models and through the DPN model by segregating the DP model output into two parts : one part was used for calibrating/training the models and the other was used for validation. These models along with a formulated SOP model were compared by adopting customised system simulation techniques.

3. An appropriate time-series model in the category of Box-Jenkins ARIMA family of multiplicative seasonal models is fitted to the monthly inflow data series and a forecast model is developed for river flow prediction. Another forecast model based on Neural Nets is formulated for the same purpose. Both the models were compared during the validation period.

4. In the final step better alternatives from among the competing models were selected and a tentative framework for how the above models can be integrated to serve as a at-site reservoir operation model, was furnished and monitoring modalities for the same were briefly outlined.

In this study, an exclusive chapter has been devoted to introduce the ANN, and discuss briefly the history, background, theory, learning algorithm and its applications in surface water hydrology. A new technique of shuffling has been introduced in this study to desensitize the input pattern sensitivity of Error-Back-Propagation Neural Networks.

Key words : **DDDP** *Discrete Differential Dynamic Programming*, **DPR** *Dynamic Programming with regression*, **DPN** *Dynamic Programming with Neural Nets*, **ARIMA** *Auto Regressive Integrated Moving Average*.

Table of Contents

CHAPTER -3 : ARTIFICIAL NEURAL NETWORK : AN OVERVIEW **27 to 39**

CHAPTER -4 : UIHE PROJECT : THE PROBLEM AREA **40 to 44**

CHAPTER -5 : DYNAMIC PROGRAMMING MODEL FORMULATION AND COMPUTATION **45 to 58**

CHAPTER -6 : DPR AND DPN MODELS 59 to 76

CHAPTER -7 : TIME-SERIES ANALYSIS FOR RIVER FLOW PREDICTION 77 to 107

LIST OF TABLES

Page

LIST OF FIGURES

NOTATIONS AND ACRONYMS

a_t	White noise, random component
ANN	Artificial Neural Network
ACF	Auto correlation function
AR(p)	Auto-regressive Regressive process of order p
Adaline	Adaptive Linear Network
B	Backward shift operator
BP	Back Propagation
BAM	Bidirectional Associative Memory Network Model
CCA	Cultuable Command area
d	Order of non-seasonal difference operator
D	Order of seasonal difference operator
DP	Dynamic Programming
DDP	Discrete Dynamic Programming
DDDP	Discrete Differential Dynamic Programming
DPR	Dynamic Programming with regression Model
DPN	Dynamic Programming with Neural Network Model
FRL	Full Reservoir Level
$f_t(x,u)$	Dynamics function
IACF	Inverse Auto correlation function
IPACF	Inverse Partial AutoCorrelation Function
$L_t(x,u)$	Single stage loss function
MDDL	Minimum Draw Down Level
MLE	Maximum Likelihood Estimate
MSE	Mean Square Error
NN	Neural Network
OCP	Optimal Controll Problem
PACF	Partial AutoCorrelation Function
Q-staticic	Porte-Manteau X_2 distributed test statistic
RACF	Residual AutoCorrelation Function
SDP	Stochastic Dynamic Programming
SIDP	State Incremental Dynamic Programming
SSE	Sum Square Error
SSF	Sum of squares function
TWL	Tail Water Level
ρ_k	ACF at lag K

CHAPTER - 1

INTRODUCTION

1.1.0 GENERAL

Water not only serves as a vital substance for human existence but also plays an important role in advancing the civilization. Owing to the rapid growth in the world economy and civilization, the need for the development of water resources has become more urgent than ever before. Water is becoming a scarce resource as a result of growing demand in its use for various purposes such as hydro-power, irrigation, water supply etc. Judicious and economic use of water is now a matter of great importance. With this in view there has been tremendous increase and expansion in the scientific and technological knowledge about water and since water is related to so many things in nature as well as in human society, this knowledge is extremely broad and inter-disciplinary.

The process by which rainfall is transformed into runoff is complex, nonlinear, time varying and spatially distributed and quite often the availability of water is at variance with those required for certain economic activities. This makes the construction of water resource projects, an inevitable proposition with the objective of development, harnessing, storage, control, allocation and utilisation of water in an optimised manner. The analysis and solution of the issues along with the allied problems form the scope of Water Resources System Engineering. This does not restrict itself into finding of an optimised solution based design of various system components only but also puts emphasis on continuous monitoring of the water resources system, ie., striking a balance between reservoir operation and the inventory problem.

The inventory problem has still been complex and addresses greater interest and research efforts as it is still an important controversy in the literature of river flow modelling regarding what constitutes the basis for appropriate statistical assumptions in river flow predictions on a sustained basis. The scheduling of the stored water in a reservoir for optimised benefit is inherently linked to the inflow sequence thus adding to the fuzziness to the task of " At site Reservoir Operation ".

The earliest reservoir operation studies in the english language appears to be a work of *Little[1955]*, who also addressed to this inter twinned problem as, *"The scheduling of the use of this stored water makes an unconventional inventory problem.*"While comparing the reservoir operations and inventory problems, he wrote,

" The hydroelectric problem differs from the usual business inventory problem in that the input, not the output is the random variable. The power demand is considered fairly well known in advance, but the river flow is not. Further more, reservoirs, unlike most ware houses have the property that the more nearly they are filled, the more valuable is each unit of water, because head is higher."

During the last two decades significant advances have been made in the systems engineering techniques and forecasting methodologies and presently these techniques are extensively used for planning and operation of water resources projects. The main motivation behind the search for better techniques for analysis of water resources systems has been the realisation of the fact that even a small improvement in the solution of the related problems has high economic value attached to it. Further more the advent of modern computers has made it very easy and time saving to use these tools.

Mathematical optimization algorithms such as linear and dynamic programming and various customised simulation techniques are the most widely used tools of systems engineering. These have been successfully applied in the study of the planning and operation of single and multi purpose reservoir systems.

In the planning models the locations and sizes of engineering structures, to meet the identified demands are analyzed taking due care of physical and budgetary constraints for the system. In the operation model the possibilities of maximising the benefits are examined keeping the preset targets intact.

But whether it be planning or operation, collection, analysis, dissemination and function approximation of a large data set, which may be composed of raw and computed/derived data and which consists of some input and output pattern is a foremost and vital aspect of systems engineering. This necessitates the adoption of proper mapping tools/ pattern recognition algorithms which can map a long historical data set for a given input pattern to that of a desired output pattern, with error minimisation and should have capability to adapt to newer environments, ie., the function so approximated is valid for future events also.

Regression techniques with least squares estimates namely multiple linear regression models and multiple non linear regression models are still the well recognised and widely applied mapping tools in the fields of systems engineering. Artificial Neural Network (ANN) is a recent tool in the fields of function mapping, which off late has attracted a lot of research in this field for its validation, application and for exploring the possibilities of ANN as a substitution for conventional mapping techniques like regression.

The development of Artificial Neural Network methods has been motivated by attempts to mimic the exceptional pattern recognition and adaptive learning abilities of biological neural networks.ANN models have been successfully used to model complex non linear input-output time series relationships in a wide variety of fields, *[Vermury and Rogers, 1994]*.

1.2.0 OBJECTIVES OF THE STUDY

The present study attempts to address an important aspect of Water resources systems engineering, ie., reservoir management as discussed in the above paragraphs and is aimed at developing a model of at-site reservoir operation to study the applicability of Artificial Neural Network in river flow predictions and reservoir water release policy. For this purpose Upper Indravati Hydroelectric project which is a multi purpose water resources project in the Nowrangpur and Kalahandi districts of Orissa has been chosen as the focal system.

Specifically, the objective is to maximise total benefits from hydro power generation, simultaneously aiming to minimise the irrigation deficit and water losses through spilling. The aforesaid obje ctive is aimed to be attained through the following steps.

1. To develop a reservoir operation policy through the conventional DPR (Dynamic Programming with

Regression) model and another operation policy through DPN model (Dynamic Programming with a Neural net) and compare the performance of the models during the validation period.

2. To fit an appropriate time series model to the historical inflow data set of Indravati river and develop a forecast model for river flow predictions and emulate the same to develop a neural net based model and test the performance during the validation period.

3. With the better alternatives chosen after comparison of various models, to work out an impleme ntation schedule for the Reservoir Operation model and furnish the monitoring modalities of the same.

1.3.0 METHODOLOGY

For finding out the optimal water releases under the objectives of maximising the benefits from power generation and minimising the irrigation deficit, the discrete, differential, dynamic programming approach has been applied. As for the monthly demand pattern, an existing demand pattern which has earlier been worked out by adopting linear programming approach by other scholars, is being taken into consideration as input to the DP model and for other computations.

The results obtained from DP - simulation are processed through the SYSTAT package for multiple linear regression and multiple non linear regression models, keeping the optimal releases as the output pattern and initial storage at the start of the time period, inflow and the demand as the three input patterns. MSE (Mean Square Error) computation and parameter estimations have been made for both the models. Further the same data set is divided into two parts (ie., training data set and testing data set) and an

ANN architecture is obtained after a rigorous training course. The same is validated after comparison with the DPR models .

A simulation model is developed for four viable options namely,

A. Multiple linear regression based model
B. Multiple non linear regression based model
C. ANN based model
D. SOP (Standard Operating based) model

to study the performances of various alternatives.

For obtaining the river flow predictions, two approaches have been adopted, namely Box and Jenkins ARIMA multiplicative seasonal modelling and the ANN model. For ARIMA forecasting two models have been formulated in FORTRAN language, one for explicitly estimating the parameters and the other for forecasting future time series values at various lead times. For ANN based forecasting a suitable architecture is searched through trial and error by employing the MSE criteria, keeping the validation period of four years, ie., employing the last four years inflow record for testing.The predicted time series sequence from both the alternatives are compared with actual inflow sequence, in order to arrive at a decision as to which one is giving more accurate and reliable predictions.

LITERATURE REVIEW

2.1.0 GENERAL

Significant advances have been made over last few decades in the field of Water Resources Engineering and especially the rapid growth of computing power in the last few decennia has enabled the development of more effective, reliable and exciting system engineering modelling tools and techniques.

Mathematical optimization algorithms such as linear and dynamic programming, with numerous state variables, constraints and decision variables and with exponential growth in computational burden, have been successfully employed with the aid of high speed computers, to arrive at near optimal level solutions.

One of the most exciting ideas emerging from this vast pool of computer based research, is the thought of emulating the low level mechanisms of the brain. Although the biological unit still out-performs any man-made tool in terms of recognition, analysis, prediction and especially learning, the alluring output from the brain simulated models have provided enough motivation to researchers to conduct extensive research into this area of artificial intelligence.

Based on the highly inter connected structure of the brain cells, the artificial neural networks, in which a new breakthrough has started since late 1980's, has by now characteristically demonstrated that this approach is faster compared to its conventional compatriots in the respective fields, robust in noisy environments, very flexible in the range of problems it can solve and highly adaptive to the newer environments.

For these already established advantages, ANN has got by now numerous real world applications such as image processing, speech processing, robotics and stock market predictions, to name a few. There has been extensive ongoing research into its implementation in the system engineering related fields such as enhanced time series prediction, rule based control and optimization, parameter identification for system simulation, runoff prediction etc. and many promising and interesting results are being published from time to time thus encouraging further research.

As the present study is aimed at studying the applicability of ANN as discussed in the previous chapter, an extensive literature review of the subject has been made within the constraints of time and availability of study materials. A separate chapter in this report has been devoted for the discussion of the theoretical aspects, computational algorithm and the literature review.

The literature survey in this chapter is divided into three sub sections namely,

1. Dynamic programming modelling
2. Simulation models
3. Time series modelling :Box and Jenkins ARIMA forecast approach

2.2.0 DYNAMIC PROGRAMMING MODELLING

Water resources problems have provided an excellent impetus and have served as a stimulus as well as a laboratory for the development and further advances in theoretical and numerical aspects of dynamic programming since 1957, when Bellman, in his celebrated book "DYNAMIC PROGRAMMING" explicitly defined it to be '*The theory of multi stage decision processes*'. Since then many inventive numerical techniques, notably DDP (Discrete dynamic programming), DDDP (Discrete differential dynamic programming), SIDP (State incremental dynamic programming) and Howard's policy iteration methods have been applied for implementing the dynamic programming.

In spite of severe limitations imposed on the scale of dynamic programming from the computational considerations, the popularity and increasing utilisation of this technique can be attributed to the fact that this enumeration technique can be used for objective functions, which are linear, non linear and even discontinuous. In addition, it has the advantage of effectively decomposing highly complex problems with a large number of variables into a series of sub problems which are solved recursively.

Another notable advantage in using the DP algorithm is that whereas in other optimisation techniques, the constraints lead to additional computations, in dynamic programming the constraints can be utilised for increased computational efficiency, since these constraints limit the feasible region.

2.2.1 Theory of Dynamic Programming

In the problem formulation, the dynamic behaviour of the system is expressed by using three types of variables, namely stage variables, state variables and control or decision variables. With each state transformation a return is associated which may either represent benefits or costs. The crux of the problem lies in identifying the appropriate control variables which optimize the returns.

Keeping up with the Bellman's principle of optimality that

"The optimal decision made at a particular stage is independent of decisions made at previous stages given the current state of the system", a set of decisions, for each time period corresponding to the finite number of states is obtained. The particular decision in the entire set, which optimises the objective function is called the optimal policy.

A system equation can be written in a discrete form for an optimal control problem (OCP) as follows.

$$X_{t+1} = f_t(x_t, u_t) \qquad 1 \le t \le N$$

Where

$\{u_t\}$	represents the control variables at time t;
$\{x_t\}$	represents the state of the system at time t;
f_t	determines the relation between $\{u_t\}$ and $\{x_t\}$;
[t]	represents the decision stages in the domain for the index t between 1 and N (first N positive integers or the set of all positive integers for an infinite horizon process) and quite often refers to time.

Assuming a minimisation problem, the objective function can be defined by,

$$J(u) = L_t\ (x_t, u_t)$$

where

$L_t(x,u)$	is a single stage loss function.
$J(u)$	is the objective function.

The goal with respect to an optimal control problem is to construct a policy u* which minimizes the objective function J(u). Typically the feasible controls are those which satisfy a vector valued state-stage dependent constraint of the form,

$$g_t\ (x_t, u_t) \le 0 \qquad 1 \le t \le N$$

The main problem with this DDP approach is in generation of impossibly large number of discretized state nodes, which limit the usefulness of this algorithm to at most four or five state and control variables.

Amongst the various potent methods developed during the last three decades, which have overcome the 'curse of dimensionality' of exponentially increasing computational burden with increase in state dimension in case of DDP, the ones particularly suited for water resources systems are DDDP, SIDP and differential dynamic programming.

The discrete differential DP which has been employed for the present study, is an iterative procedure, in which the recursive equation of DP is solved within a restricted set of state variables. The optimal solution is obtained by gradually improving upon the initial solution. The prototypical DDDP is most simply described as DDP with the added constraint that for each time period t and for some specified $\epsilon > 0$,

$$\| x_t - x_t' \| < \epsilon$$

Heidari et al.[1971], to whom the designation DDDP is due, described the above constraint, by saying that the successor trajectory must lie in a 'corridor' of width and centred about the nominal trajectory. Figure (2.1) shows a typical corridor boundary and the state variable (storage) discretization into a number of feasible stages.

2.2.2 DP Application in Reservoir Operations

It is interesting to note that application of stochastic DP algorithms [*Masse 1946; Little 1955*] in reservoir operations precede applications of deterministic dynamic programming by over a decade, although it is more subtle and computationally

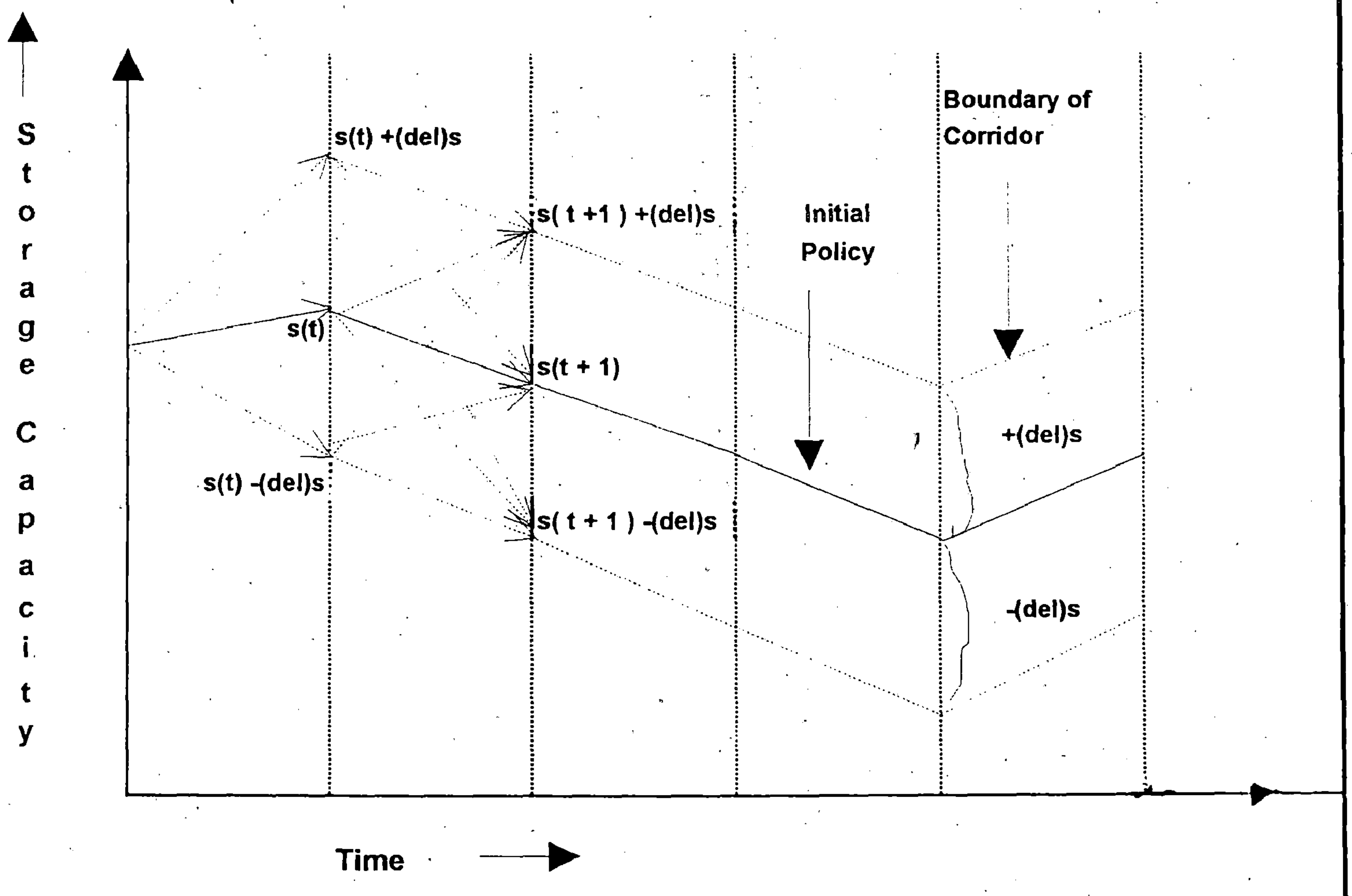


Fig - 2.1 : Discrete Differential Dynamic Programming troublesome *[Yakowitz 1982]*. The first application of deterministic D.P. in reservoir applications was made by *Young[1967]*. He studied a finite horizon single reservoir operation problem, with a viewpoint to support the critical assumption that he made that the inflows are known by the hypothesis that some rivers are regular enough so that their flows are well represented by their expectations.

Hall et al.[1968] presented specific ideas for determining the single stage loss function in their computational study associated with Shasta dam in California. This was further extended to the multi reservoir cases by authors *Roefs & Bodin[1970]*.

A four-reservoir problem served as the bench mark for testing the DDDP developed by *Heidari et al.[1971]*. This same problem was studied by *Larsen[1968]* with the procedure of "incremental Dynamic programming". *Chow[1975]* used the same problem with the Discrete Differential Dynamic Programming.

Studies related to the multi objective DP for reservoir operations were initiated by *Tauxe et al.[1979]*. They have reduced the multi objective problem to one in which a certain objective is minimized, while the others are being maintained below certain threshold levels.

Banerjee & Harikrishna[1975] presented a state incremental DP model to determine the optimal operating policy for the Damodar Valley Corporation system using the critical period of the observed flows. The objective was to maximise the hydropower output from this multi purpose multi-reservoir system serving for irrigation and flood control.

Harikrishna et al.[1981] studied the integrated operation of Bhakra-Beas system using the incremental DP technique. The annual power generation was maximised subject to physical constraints ensuring that the power releases should equal irrigation requirements.

2.2.2.1 DPR Models

This deterministic model (DPR) consists of an algorithm that cycles through three components: a dynamic program, a regression analysis and a simulation model *(Karmouz and Houck, 1987)*. *Young[1967]* first proposed the use of a linear regression procedure to find general operating rules from deterministic optimization. He derived regression equations using inflows and storages to find optimal releases.

The authors *(Karmouz and Houck 1987)* have shown DPR model to be a significant extension of other deterministic models, by deriving regression equations from deterministic DP results. Their DPR model incorporates a multiple linear regression procedure, suggested by *Bhaskar and Whitlach[1980]* and a hypothetical loss function was used in that study.

Optimised release policies obtained from DP-regression methodology, were compared by *Bhaskar and Whitlach [1987]*, with the release policies obtained from chance constrained linear programming for a single multi purpose reservoir system and system performance was derived from simulation techniques.

Raman and Chandramauli[1996] adopted a DPR model along with DPN and SDP models to derive reservoir operating policies and expressed the optimal release as a linear function of initial storage, inflow and demand, which is of the form

$$R_t = aS_t + bI_t + cD_t + d \quad (2.1)$$

They used the DP results for regression using the least squares method.

2.3.0 SIMULATION MODELS

A simulation model is meant to provide the response of the system, for certain inputs, which enable the decision makers to examine the consequences of various scenarios of an existing system or a new system without actually building it. It tries to approximate the behaviour of a system, representing all the system characteristics, largely by a mathematical or algebraic description. A typical simulation model for a water resources system is simply a model that simulates the interval-by-interval operation of the system with specified inflows at all locations during each interval for specified system characteristics and specified operation rules.

Development of a simulation model is governed by the system operating policies, such as the standard operating policy, rule curve based operation, multiple zoning, target storage level based operation or else regression parameter based operation policy.

2.3.1 Application

Now a days, with the advent of high processing computers, standard simulation packages have become available, with a wide range of flexibility to accommodate a varied range of customised conventional problems. However the earliest simulation model associated with a system of reservoirs appearing in the literature seems to be the study performed by the U.S. Army Corps of Engineers in 1953 for an operational study for six reservoirs on the Missouri river with the objective of maximising power generation subject to constraints of navigation, flood control and irrigation.

The simulation model applied by *Maass et al. [1962]*, to the economic analysis of water resources system design, reproduced the behaviour of the system for power generation, irrigation and flood control.Using synthetic stream flow sequences, *Hufschmidt and Fiering[1966]* used simulation in planning the multi-reservoir, multi-purpose Lehigh river system and worked out designs with higher benefits than the existing system.

The HEC-3 model (Reservoir system analysis for conservation) developed by the Hydrologic Engineering Centre has been applied to the operational studies on the Arkansas-White-Red river system in the Southern United States *[Frederich and Beard, 1972]*. The HEC-5 model (Simulation of flood and conservation systems), developed by the Hydrologic Engg. Centre has been applied to the reservoir system expansion study for flood control on the Susquehanna river system *[Eichert and Davis,1976]*.

The DELTA model was developed for Mekong committee by SOGREAH, a French consulting firm, as a tool to simulate with reasonable accuracy, the hydraulic regime of the Mekong delta in Thailand. The model has been used for delta reclamation investigation, optimal sizing of a dam and planning of flood control schemes.

Srivastava et al.[1980], studied the Bargi, Tawa, Narmadasagar and Navagam reservoirs by simulation for Narmada basin in India. In another study by *Ramaseshan[1981]*, the SIMYLD II simulation programme of the Texas Department of Water Resources, was used to get the modified rule curves, for the Bhakra and Pong reservoirs and these were compared with the rule curves derived by Beas Design Organisation.A Hydrodynamic Model (SYSTEM 11F) for river routing and reservoir simulation has been established for real time flood forecasting in Damodar Basin by the Central Water Commission in co-operation with Danish Hydraulic Institute.

Raman and Chandramauli[1996] in their effort to improve the operation and efficient management of available water for the Aliyar dam in Tamilnadu, India, used DP, SDP, SOP & DPN models and studied their relative performances by using reservoir simulation model. For testing the SDP scheme, they used a simulation model where releases are made based on SDP results and the balance demand is met through a SOP norm.

2.4.0 TIME SERIES MODELLING: BOX AND JENKINS APPROACH TO FORECASTING

The Box-Jenkins approach to time-series analysis, forecasting and control is a powerful but rather complicated procedure. The methods are potentially useful in many types of situations which involve the models for discrete time series and dynamic systems. They have also been adopted to the problem of forecasting seasonal time-series*[Chatfield and Prothero 1973]*.

Box-Jenkins approach provides a particular class of time-series models, which require the fitting of a suitable stochastic model and construction of recursive formulae for calculating the linear least square predictors corresponding to the identified model. This family of linear stochastic models, that are now referred to as Box-Jenkins or Auto-Regressive Integrated Moving Average (ARIMA) models, is in fact a culmination of research of many prominent statisticians starting with the pioneering work of *Yule[1927]*. The ARIMA models have been extensively used for modelling of river flow sequences *[Dellur and Kavas,1978]*.

For applying the Box-Jenkins model to any time-series data, three stages of model development, namely, identification, estimation and diagnostic checking, are to be adhered to, *[Box and Jenkins,1976]*. The first step is to identify the form of model that may fit to the given data. The series may need to be differenced at this stage. At the estimation stage, the model parameters are calculated by employing the method of maximum likelihood. Finally the model is checked for possible inadequacies. If the diagnostic checks reveal serious anomalies, while analyzing the residuals, appropriate model modifications are made by repeating the identification and estimation stage.

These models rely heavily on the appropriate use of three familiar time series tools, namely, differencing, autocorrelation function (ACF) and partial auto correlation function (PACF). Differencing is used to reduce non stationarity to stationarity. ACF and PACF are used to identify an appropriate ARIMA model and the required number of parameters.

A brief theoretical description of ARIMA difference equations and three stages of ARIMA model building are furnished below.

2.4.1 ARIMA Model

Let $z_1, z_2, z_3, \ldots\ldots, z_{t-1}, z_t, z_{t+1}, \ldots\ldots z_N$ be a discrete time series measured at equal time intervals. A multiplicative seasonal ARIMA model is written as *[Box and Jenkins 1976]*

$$\phi(B)\,\Phi(B^s)\,[(1-B)^d(1-B^s)^D z_t]-\mu=\theta(B)\,\Theta(B^s)\ldots \qquad (2.2)$$

where

z_t Some appropriate transformation of the time series data, such as a log transformation.(No transformation is also a possible option);

t discrete time;

s seasonal length, equal to 12 for monthly river flows;

B backward shift operator defined by $B^s z_t = z_{t-s}$;

μ mean level of the process, usually taken as the average of the w_t series (if D+d >0 often μ= 0)

a_t normally independently distributed white noise residual with mean zero and variance σ_a^2.

$$w_t = \nabla^d \nabla_s^D z_t \quad where \tag{2.3}$$

w_t becomes the stationary series formed by differencing z_t series (n = N - d -sD is the number of terms in the w_t series);

$$(1-B)^d = \nabla^d \quad ; \ (1-B^s)^D = \nabla_s^D \tag{2.4}$$

$(1-B)^d$ = becomes non seasonal differencing operator of order d to produce non seasonal stationarity of the d*th* differences, usually d=0,1 or 2;

$(1-B^s)^D$ = Seasonal differencing operator of order D to produce seasonal stationarity of the D*th* differenced data, usually D= 0,1 or 2;

$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots\ldots\ldots\ldots - \phi_p B^p$ nonseasonal autoregressive (AR) parameter or polynomial of order p such that the roots of the characteristic $\phi(B) = 0$ lie outside the unit circle for nonseasonal stationarity and the ϕ_i, i = 1,2,3, ...,p are the nonseasonal AR parameters;

$\phi(B^s) = 1 - \phi_1 B^s - \phi_2 B^{2s} - \ldots\ldots\ldots\ldots - \phi_p B^{ps}$ seasonal (AR) operator of order p such that the roots of $\phi B^s = 0$ lie outside the unit circle for seasonal stationarity and the ϕ_i, $i=1,2,\ldots,P$ are the seasonal AR parameters.

$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \ldots\ldots\ldots - \theta_q B^q$ nonseasonal moving average (MA) parameter or polynomial of order q such that the roots of the characteristic $\theta(B) = 0$ lie outside the unit circle for invertibility and the θ_i, $i = 1,2,3, \ldots, q$ are the nonseasonal MA parameters;

$\Theta(B^s) = 1 - \Theta_1 B^s - \Theta_2 B^{2s} - \ldots\ldots\ldots - \Theta_P B^{Qs}$ seasonal (MA) operator of order Q such that the roots of $\Theta B^s = 0$ lie outside the unit circle for seasonal stationarity and the Θ_i, $i=1,2,\ldots,Q$ are the seasonal MA parameters.

The notation $(p, d, q) X (P, D, Q)_s$ is used to represent the multiplicative seasonal ARIMA model of eqn.(2.1). THe first brace with small alphabets contains the nonseasonal AR, differencing order and MA operators and the second set of brackets contains the corresponding seasonal operators. As an example, a stochastic multiplicative seasonal noise model of the form $(0,0,2)X(0,1,1)_{12}$, with no data transformation, which has been identified for the present study and shall be discussed in detail elsewhere can be written as

$$\{(1 - B_{12}) z_t - \mu\} = (1 - \theta_1 B - \theta_2 B^2)(1 - \Theta_1 B^{12}) a_t$$

2.4.2 Stages of ARIMA Model Building

2.4.2.1 *Identification*

The purpose of the identification stage is to determine the differencing required to produce stationarity and also the order of both the seasonal and the nonseasonal AR and MA operators for the w_t series with the help of ACF, PACF plots and the plot of original time series.

Autocorrelation function (ACF) :

The autocorrelation function measures the amount of linear dependence between observations in a time-series, that are separated by lag k. *Box and Jenkins[1976,pp.32-36]*, recommend a specific estimation procedure to determine an estimate r_k for ρ_k and also give approximate standard errors for the ACF estimates. It is shown by *Box and Jenkins[1976,pp.174-175]*, that the estimated ACF at lags that are linear multiples of the seasonal length S doesn't die out rapidly, which indicates that seasonal differencing is needed to produce stationarity.

Partial autocorrelation function (PACF) :

Whenever the model fitting involves an AR process, the appropriate number of lags to use (*ie. order P of the model*) can be determined by analysis of the PACF (ϕ_{kk}), which satisfies the Yule Walker conditions.[*Box and Jenkins,chap-3*]. This is a measure of correlation between z_t and z_{t-k}, after adjusting for the presence of all the z_t's of shorter lag, ie, z_{t-1}, z_{t-2}, . . ., z_{t-k+1}. This adjustment is done to see if there is an additional correlation between z_t and z_{t-k} above and beyond that induced by the correlation, which z_t has with z_{t-1}, z_{t-2}, . . ., z_{t-k+1}.

2.4.2.2 ESTIMATION

Box & Jenkins[1976,chap-7] suggest that the approximate likelihood estimate for the ARIMA model parameters be obtained by employing the unconditional sum of squares method, wherein the unconditional sum of squares function is minimised to get least squares parameters estimate. Various optimization techniques are available to minimize functions such as the unconditional SSF. Some of them, which have been extensively applied include

1. Gauss linearization
2. The steepest descent
3. Marquardt algorithm (combination of above two.this method has been adopted for the present study).

Akaike information criterion (AIC) :

Box & Jenkins [*1976*], stress the need to construct a model which should be parsimonious(ie. to use as few parameters as possible) so that the model passes all diagnostic checks. The AIC [*Akaike 1974*] is a mathematical formulation of the parsimony criterion of model building.

For comparing among competing ARIMA family models, AIC can be written mathematically as

$$AIC(p,q) = N.\ln(\hat{\sigma}_{\varepsilon})^2 + 2(p+q) \qquad (2.6)$$

where

N Sample size;

σ_{ε} Maximum likelihood estimate of the residual variance;

p,q The order of AR and MA operators;

The model which gives the minimum AIC should be considered.

2.4.2.3 Diagnostic checks :

In Box & Jenkins modelling, the residual a_t are assumed to be independent, homoscedastic (ie. variance is constant) and usually, are normally distributed. Most diagnostic tests deal with the residuals to determine the aforesaid assumptions. Homoscedasticity and normality are considered to be less important violations as these can often be corrected by a Box-Cox transformation of the data. But the lack of independence of the residuals indicates that present model is inadequate and the entire process of identification and estimation stages are repeated in order to determine a suitable model.

Various diagnostic checks and tests include, amongst others, overfitting, RACF analysis and Porte-Manteau lack of fit test, to know whether residuals constitute an independent series or not and whether the Homoscedasticity check is satisfied or not.

I. Overfitting :

When an ARIMA(p,d,q) model has been tentatively accepted, overfitting involves fitting a more elaborate model than the one estimated to see if inclusion of one or two parameters greatly improves the fit, ie, one can successfully add and test additional AR or MA terms (but not both simultaneously) to the model until the last term added is not significant.

II. RACF analysis :

Residual autocorrelation function is analyzed to determine whether the residual a_t is white noise or not. many new sensitive techniques are now available for checking the independence assumption of a_t. However examining and plotting of the ACF and PACF of the residuals gives sufficient idea regarding the randomness of the residuals.

Porte Manteau lack of fit test:

Porte manteau lack of fit test, originally proposed by Box and Pierce *[1974]* and later on modified to validate the same for a general seasonal Box-Jenkins model has been adopted for the present study. the test statistic and hypothesis testing are furnished below.

X_t be a time series of size N represented by ARIMA (p,d,q) model. So after d differences, the ARMA(p,q) series, z_t, t = 1,......,N-d is obtained.
ϵ_t be the residual series.

The test is applied to know whether ϵ_t is an independent series or not.

$$Q = (N-d) \sum_{(k=1)}^{L} (r_k)^2(\epsilon) \qquad (2.7)$$

L is the maximum lag considered
$r_k(\epsilon)$ is the correlogram of the residuals.
Q is approximately χ^2 distributed with L-p-q degrees of freedom.
If Q < χ^2(L - p - q), ϵ_t is considered to be an independent series, so the model is considered to be adequate.

2.4.3 Applications

Box-Jenkins analysis provides a systematic approach to model selection, utilising all the information contained in the sample autocorrelation function (ACF) and partial auto correlation function (PACF). With the advances made in the Box-Jenkins model construction, such as, inspection of inverse autocorrelation function (IACF), inverse partial auto correlation function (IPACF) at the identification stage and sensitive diagnostic checks for residual independence through estimation of residual autocorrelation function (RACF) and other diagnostic tests to determine whether the homoscedasticity and normality assumptions are fulfilled or not, the ARIMA models have been shown to be optimal, since for a given model, no other forecasting method can on average, give forecasts, with smaller MSEs *[Pankratz,1983]*.

Chatfield and Prothero *[1973]* made a critical appraisal of Box & Jenkins seasonal ARIMA model in their paper, wherein they have given a step by step account of the analysis and problems encountered *en route*, while applying the model to a forecast of sales problem with a lead time up to 12 months.

Mclleod et al.[1977] considered three different problems, to determine both seasonal and nonseasonal models for actual time-series, by employing Box-Jenkins techniques and carefully demonstrated the utility of the procedure. Apart from the *classic Air line passenger data problem* of Box-Jenkins, where better parameter estimates have been obtained by the authors, than those calculated by using the unconditional sum of squares technique, the two other problems considered by the authors are, annual river flows of Saint Lawrence River, and the Yearly Wolfer Sunspot Number series. In these cases also they could derive better models than that originally derived.

Delleur and Kavvas *[1978]* applied the ARIMA model to the average monthly rainfall time-series over 15 basins located in Indiana, Illinois and Kentucky and have found that the seasonal differencing is effective in removing the periodicities but distorts the spectral structure of the original rainfall series, whereas cyclic standardization introduces negligible distortion in the random component.

Hipel *[1985]* in his review paper, have analyzed the recent developments in time-series analysis and capabilities of various time series models by employing a set of criteria, and outlined therein, some of the advantages and limitations of ARIMA model.

O' Connell and R.T. Clarke[1986] have derived the relative merits of the associated parameter estimation algorithms, from an inter comparison of a number of real time forecasting models, including Box-Jenkins models and state-space/Kalman filtering models and have assessed critically the validity of the underlying assumptions of each, in the hydrological forecasting context.

ARTIFICIAL NEURAL NETWORK : AN OVERVIEW

3.1.0 GENERAL

The term neural network refers to the circuitry of real brains or to technological devices for a mode of parallel computation. Neural networks constitute an important discipline in Artificial Intelligence" (AI), as historically AI grew out of the work in neural networks, way back in 1956. This mode of computation is commonly known as neural computing or study of artificial neural networks. It is also referred tc as connectivism or parallel distributed processing.

Neural networks provide a unique computing architecture whose potential has only begun to be tapped. Used to address problems that are intractable or cumbersome with traditional methods, these new computing architectures, inspired by the structure of brain, are radically different from the computers that are widely used today.

As the subject grew out of a noble attempt for low level imitation of the real brain, before describing the theoretical approaches and computational algorithms, the relevant structure and functioning of an actual biological neuron, which has been imitated by the neural network computing paradigm is described below.

Schematized properties of a basic neuron are given in Fig(3.1). The dendrites comprise the input surface, axon provides the output channel. Tips of axon branches become end bulbs, forming a synapse on the cell, on which they impinge. Through synapse the transfer of signal occurs by potential

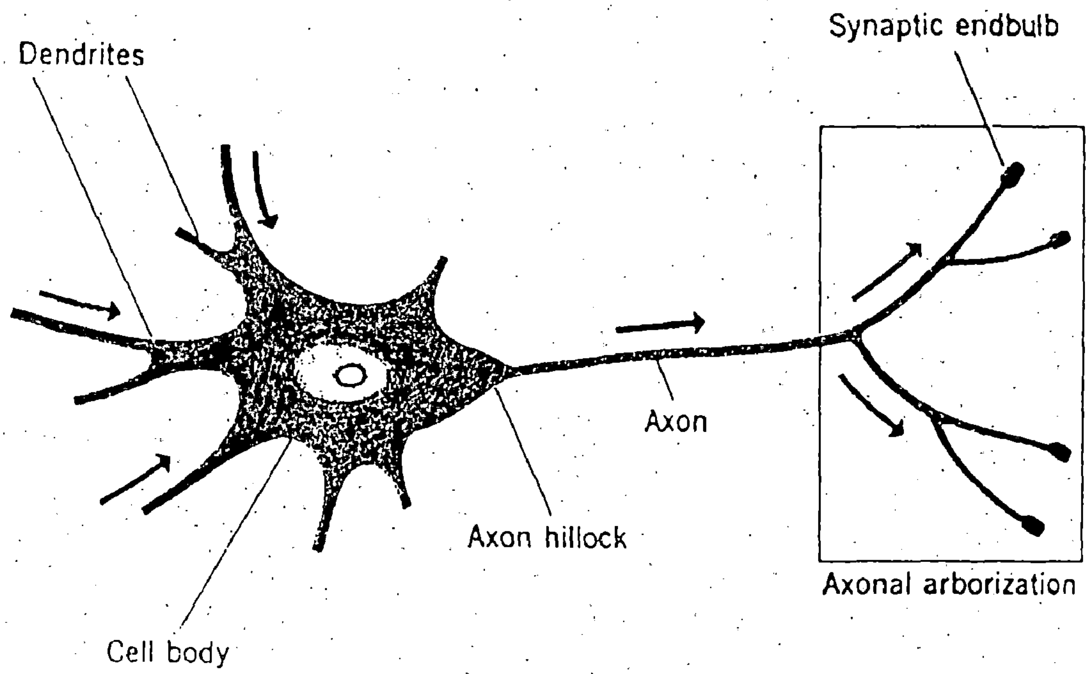


Fig - 3.1 Schematic view of a neuron. Activity from receptors or other neurons modifies membrane potentials on the dendrites and cell body.

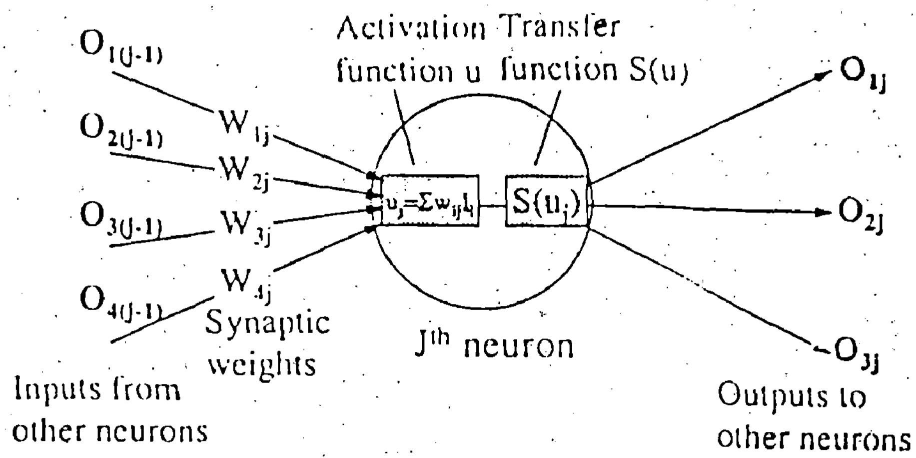


Fig - 3.2 *Neuron schematisation*

difference mechanism. If the change in potential difference across the synaptic end bulb exceeds a threshold, an action potential actively propagates to the neighbouring neuron without decrement. The transmittance depends upon the strength of the signal generated by the neuron after processing the information. This has only two options: either excitory response or an inhibitory response. The unique intelligence, exhibited by human brain is imparted by the large interconnection of billions of neurons in a human body.

Figure (3.2) shows the abstract equivalent of the nerve cell: the artificial neuron. This is based on the following features. First there are weighted input connections to the neuron (dendrites). Then these input signals are added up and fed into an activation function, which determines whether the neuron will react at all, (cell body). If this is the case then the signal will pass through a transfer function, which determines the strength of the output signal, (hillock). Finally the output signal will be sent through all the output connections, (synapse) to the other neurons.

Therefore, the neural networks utilize a parallel processing structure that has large number of processors, in line with the biological neurons described above and provide many inter-connections between them. The power of neural network lies in the tremendous number of interconnections. The major aspects of a parallel distributed processing model, and so for an ANN model are

- A set of processing units, called neurons;
- A state of activation;
- An output function for each unit;
- A pattern of connectivity among various units;
- A propagation rule for propagating patterns of activities through the network of connectivities;

- An activation rule for combining the inputs impinging on a unit with the current state of that unit to produce a new level of activation for the unit;

- A learning rule whereby patterns of connectivity (weights) are modified by experience;
- An environment within which the system must operate.

Depending upon the interconnections of the neurons, which are arranged in many layers, namely, one input layer, one output layer and one or more hidden layers, and by manipulation of network parameters based on above aspects (e.g., network structures or learning algorithm), numerous types of ANNs exist, all with their specific application purposes. An ANN classification tree is furnished in Annexure(II) to get a better overview of various ANN types. However the subsequent discussion is constrained only to the feed forward BP neural networks. For an overall picture of the ANN architecture, various layers, arrangement of neurons in the layers, neural connectivities etc., a three layer feed forward artificial network along with a typical processing element, with an activation function and a threshold function embedded to its body, is shown in Figure(3.3).

3.2.0 *DATA PROCESSING THROUGH ANN*

The first step in starting the data processing is to construct an architecture, by adopting appropriate model attributes as discussed above, clearly delineating the links and interconnections. The input neurons receive the user given signals, process the same and send the corresponding output to other neurons, where the process of action and reaction is continued.

The data passing through the connections from one neuron to another, are manipulated by weights, which indicate the

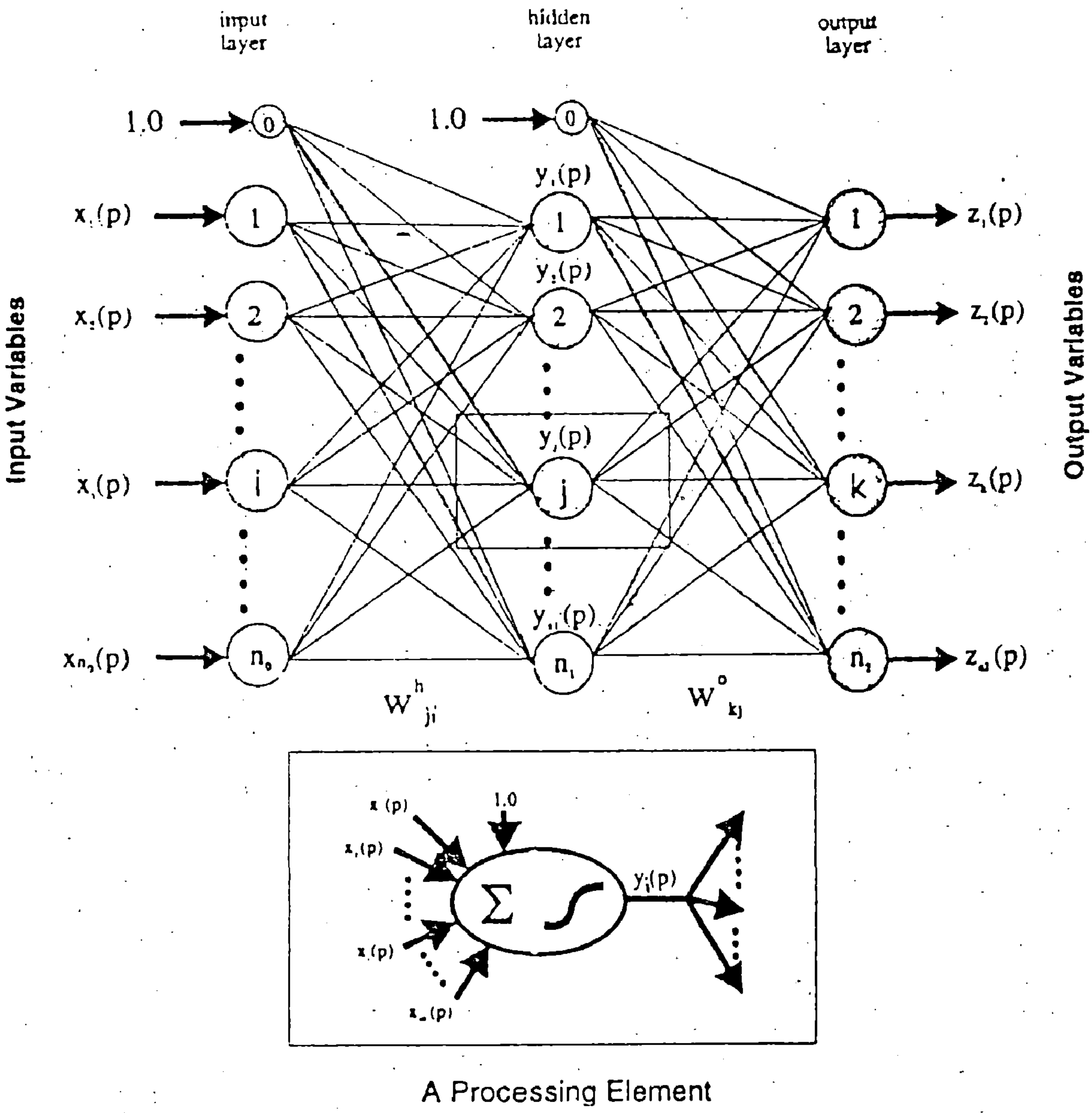


Fig 3.3 A typical 3-layer Neural Network along with a processing element

strength of a passing signal. Consequently, when these weights are modified, the data transferred through the network, will change and the overall network performance will alter.

These new manipulating parameters, can all be adjusted and optimized, in order to get a specific response from an ANN. The process of adjustment and optimization is called learning and is defined by the learning algorithm of an ANN. Learning algorithm is a set of optimization functions which adjust the weights in such a manner, that an input signal is correctly associated with a desired output signal. Several learning examples are presented to the network each one attributing to the optimization of the weight distribution. Finally, when ANN has learned enough example, it is considered to be trained.

After the learning cycles, the learning algorithm is (often) deactivated and the weights are frozen. Then test data is presented to the ANN, which it has never encountered before, enabling a validation of its performance. This is referred to as testing of an ANN. Depending on the outcome, either the ANN has to relearn the examples with some modifications, or it can be implemented for its designated use.

3.3.0 LEARNING ALGORITHM : THE DELTA RULE

The learning algorithm, adopted in the program for the present study is based upon the *"generalised delta rule"* proposed by Rumelhart *[1986]*. The learning procedure involves the presentation of a set of pairs of input and output patterns. The system first uses the input vector to produce its own output vector and then compares with the *desired output*. In case there is difference, learning takes place.

In its simplest form, the delta rule for changing weights following presentation of input/output pair P is given by,

$$\Delta_p w_{ji} = \eta (t_{pj} - O_{pj}) i_{pi} = \eta . \delta_{pj} . i_{pi} \qquad (3.1)$$

where

$$\delta_{pj} = (t_{pj} - O_{pj}) \quad (\eta \text{ is the learning parameter}) \qquad (3.2)$$

t_{pj} = target input for Jth component of the output pattern for pattern p.

O_{pj} = j^{th} element of the actual output pattern produced by the presentation of input pattern p.

i_{pi} = is the value of the *ith* element of the input pattern.

There are many ways of determining this rule. A brief outline of the derivation/algorithm for the delta rule for semilinear activation functions, such as, the sigmoid function, in feed forward networks is given below. A semilinear activation function is one, in which the output of a unit is a non decreasing and differentiable function of the net total output.

In case of batch processing of long sets of input/output patterns with multi layer feed forward networks, which is usually the case and which is adopted for the present study also, let E represent the sum square error function, i.e., the overall measure of error.

Let,

$$E = \sum E_p \text{ where}$$

$$E_p = \frac{1}{2} \sum_j (t_{pj} - O_{pj}) \qquad (3.3)$$

E_p be the measure of error on one set of input/output pattern p.

$$net_{pj} = \sum_i w_{ji} . O_{pi} \qquad (3.4)$$

where $O_i = i_i$, if unit i is an input unit. Thus a semilinear activation function is one in which,

$$O_{pj}=f_j(net_{pj}) \tag{3.5}$$

$$\Delta_p w_{ji} \propto -\frac{\partial E_p}{\partial w_{ji}} \tag{3.6}$$

$$\frac{\partial E_p}{\partial w_{ji}}=\frac{\partial E_p}{\partial net_{pj}} \cdot \frac{\partial net_{pj}}{\partial w_{ji}} \tag{3.7}$$

The first part of equation (3.7) reflects the change in error function of the change in the net input to the unit, and the second part represents the effect of changing a particular weight on the net input. The second factor is in fact,

$$\frac{\partial net_{pj}}{\partial w_{ji}}=\frac{\partial}{\partial W_{ji}}\sum_k w_{jk} . O_{pk}=O_{pi} \tag{3.8}$$

Let,

$$\partial_{pj}=-\frac{\partial E_p}{\partial net_{pj}} \tag{3.9}$$

So equation (3.7) becomes,

$$-\frac{\partial E_p}{\partial w_{ji}}=\delta_{pj} . O_{pi} \tag{3.10}$$

The above equation implies that , to implement gradient descent in sum square error function **E**, the weight changes should be according to the rule,

$$\Delta_p w_{ji} = \eta \partial_{pj} . O_{pi} \tag{3.11}$$

η is a learning parameter for the layers connected by the i^{th} and j^{th} neurons.
To compute,

$$\delta_{pj} = -\frac{\partial E_p}{\partial net_{pj}} \tag{3.12}$$

a chain rule is applied, to write this partial derivative as the product of two factors, one factor reflecting the change in error as a function of the output of the unit and the other reflecting the change in the output as a function of changes in the input. so,

$$\delta_{pj} = -\frac{\partial E_p}{\partial net_{pj}} = -\frac{\partial E_p}{\partial O_{pj}} . \frac{\partial O_{pj}}{\partial net_{pj}} \tag{3.13}$$

The second factor is obtained by differentiating equation (3.5) with respect to net_{pj}. So that,

$$\frac{\partial O_{pj}}{\partial net_{pj}} = f'_j(net_{pj}) \tag{3.14}$$

This is simply the derivative of the squashing function f_j for the j^{th} unit.
The first factor is computed under two considerations - one, when unit U_j is an output unit of the network and the other when it is not an output unit of the network. In case one,

Substituting this and equations (3.14) and (3.13), we obtain,

$$\frac{\partial E_p}{\partial O_{pj}} = -(t_{pj} - O_{pj}) \tag{3.15}$$

$$\delta_{pj} = (t_{pj} - O_{pj}) f'_j(net_{pj}) \tag{3.16}$$

In case two, for U_j not an output unit, the chain rule is used, i.e.,

$$\sum_k \frac{\partial E_p}{\partial net_{pk}} \cdot \frac{\partial net_{pk}}{\partial O_{pj}} = \sum_k \frac{\partial E_p}{\partial net_{pk}} \cdot \frac{\partial}{\partial O_{pj}} \cdot \sum_i w_{ki} O_{pi}$$

$$= \sum_k \frac{\partial E_p}{\partial net_{pk}} \cdot w_{kj} = -\sum_k \delta_{pk} \cdot w_{kj} \tag{3.17}$$

In this case substituting for the two factors in equation (3.13), we obtain,

$$\delta_{pj} = f'_j(net_{pj}) \sum_k \delta_{pk} \cdot w_{kj} \tag{3.18}$$

Equations (3.16) and (3.18) give a recursive procedure for computing the δ's for all units in the network. These are then used to compare the weight changes in the network according to equation (3.11). This procedure constitutes the generalized delta rule for a feed forward network of semilinear units.

3.4.0 APPLICATIONS

The general application areas for ANNs can be divided into prediction, simulation, classification, optimization and identification problems. Translated to possible hydrology applications, ANNs have already been used for runoff/flow predictions, flow/pollution simulation, control strategy definition or system parameter identification. Out of the many ANN structures proposed and explored since 1950s, namely multi layer feed forward networks[*Rumelhart et al., 1986*], self organising feature maps[*Kohonen, 1982*], Hopfield networks [*Hopfield, 1982*], and counter propagation networks[*Hecht Nielsen, 1987*], the multi layer, feed forward networks have been found to have the best performance with regard to input output function approximation, and are mostly used to address the hydrology related problems.

Krajewski and Cuykendall [1992], developed a three layer feed forward neural network, to forecast a rainfall intensity in the fields of space and time, and compared the result with two other methods of short term forecasting, persistence and nowcasting.

Smith and Eli[1996] used neural network model for generating runoff using a synthetic watershed from stochastically generated rainfall patterns. They trained a back propagation network to predict the peak discharge and the time to peak resulting from a single rainfall pattern.

Minns and Hall[1996] used artificial neural networks to generate flow data from synthetic storm sequences and routed the flow data through a conceptual hydrologic model, consisting of a single nonlinear reservoir. important findings of the paper include the importance of various standardisation schemes and redundancy and lack of justification for opting to have more than one hidden layer in the neural net.

Raman and Sunilkumar [1995] investigated the use of artificial neural networks in the field of synthetic inflow generation and compared the model performance with that of a multi variate auto regressive (ARMA) model, proposed by Box &

Jenkins [1976] in case of two reservoir sites at Mangalam and Pothundy.

Lorrai and Sechi [1995] examined the capability of neural nets to provide a suitable forecast of river runoff, for the Araxisi watershed in Sardinia. They divided the observed data into three training sets of ten year periods each, built a neural network with two hidden layers, adopted BP learning rule and Sigmoid as the response function and corresponding to each training set, simulated the other two decades for appraisal of model performance.

Hammerstorm [1993] in his paper, has demonstrated the fact that developing a neural network, is unlike developing a software, because the network is trained, not programmed. It is a prisoner of the instances by which it is trained.

Hsu et al. [1995 - 1997], have worked extensively in artificial neural network modelling of rainfall runoff process, watershed modelling, runoff forecasting and in the field of developing better learning algorithms for ANN structure. They advocated the use of a three layer feed forward network with Linear Least Square SIMplex (LLSSIM) algorithm, for simulating the nonlinear hydrologic behaviour of the watersheds.

Carriere et al. [1996] designed a virtual runoff hydrograph system (VROHS), based on ANN technology, by training a recurrent back propagation neural network. They got very good correlation between the observed and predicted data, while validating the network for testing data set. Chang and Noguchi [1996] demonstrated the fact that by adopting NN based partial intelligent model to rainfall runoff modelling, parameters relating to catchment can be avoided in the input and virtually, no parameter inside the model need to be calibrated manually.

Published works, in the field of reservoir operation using neural network approach, are very scanty. The only paper, published so far, in standard literatures, is of Raman and Chandramauli[1996]. They derived reservoir operating policies, for the Aliyar dam in Tamilnadu, India, by using a neural network procedure (DPN model) and by using a multiple linear regression procedure (DPR model) from the DP algorithm. They also adopted a SDP model and a standard operating policy (SOP) and compared the performance of each during the validation period taking last three years of historic data. they demonstrated the fact that DP algorithm based DPN model provided better performance than other models. this paper has been taken up as the prime guiding literature, for the present study.

UIHE Project : The Problem Area

4.1.0 THE PROJECT

Upper Indravati Hydro-electric Project, which is a large multi purpose project, situated in the Nowrangpur and Kalahandi districts of Orissa, envisages construction of four dams across the four rivers viz: Indravati, Podagada, Kapoor and Muran, and eight number of dykes.

4.1.1 Indravati River System

The river Indravati, across which the main dam is constructed and after which the project is named, originates in Kalahandi district, Thuamalrampur plateau, at an altitude of more than 915m, on the western slope of the Eastern Ghats and traverses in the south westerly direction , through the hilly ranges, until emerging into planes at Khatiguda village in the district of Nowrangpur. On its way through a number of rapids, it is joined by a number of tributaries namely, Podagada, Kapoor and Muran rivers. Thereafter the river flows through the Nowrangpur district and enters Bastar district of M.P. state, near Jagdalpur. After flowing for a total run of 530 km from its origin, ultimately, the river Indravati joins river Godavari.

4.1.2 The Combined Reservoir

The reservoir, draining an area of 2630 km^2 and having a gross reservoir capacity of 2300 million cubic meters and water spread at FRL, of 110 km^2, is formed by four sub basins, created by above four rivers. These four sub basins are inter connected by link canals, so that the reservoir does not get disconnected during the operation. A project map is shown in Figure(4.1). Water for the power generation will be fed by the reservoir at the upstream, at about 6.6 km from the edge of the water spread at FRL of 642.00m, on Indravati river. The entire reservoir is 43 km long and 9 km wide at the widest section. The power intake location is at 37 km from the southern edge of the reservoir and approximately, 13 km upstream of Indravati dam, which is an uncommon and special feature of this project.

4.1.3 Power Generation and Irrigation

Water for power generation will flow through a 7m diameter head race tunnel, designed for a capacity of 210 m^3/s. The power house would contain four numbers of Francis turbines coupled to 150 MW generators each, thus producing 600 MW of effective power. After power generation , flow from tail race channel will be fed to the Hati river and diverted into irrigation canals by a barrage structure near Mangalpur village. The canal sections would irrigate 1,28,000 ha. (CCA) of agricultural land in the watershed of Hati river adjacent to Mahanadi river basin.

The distinct feature of the project is trans-basin diversion of water of river Indravati (Godavari basin) into river Hati (Mahanadi basin), for power generation and subsequent irrigation. The principal data pertaining to various project features is shown in Annexure - I.

4.2.0 PROBLEM FORMULATION

The present study attempts to optimize the water release from the reservoir, and device a reservoir operation model accordingly, in order to explore the optimization potential of the Upper Indravati Hydro-electric Project. As the project is at the verge of completion, all the reservoir parameters,

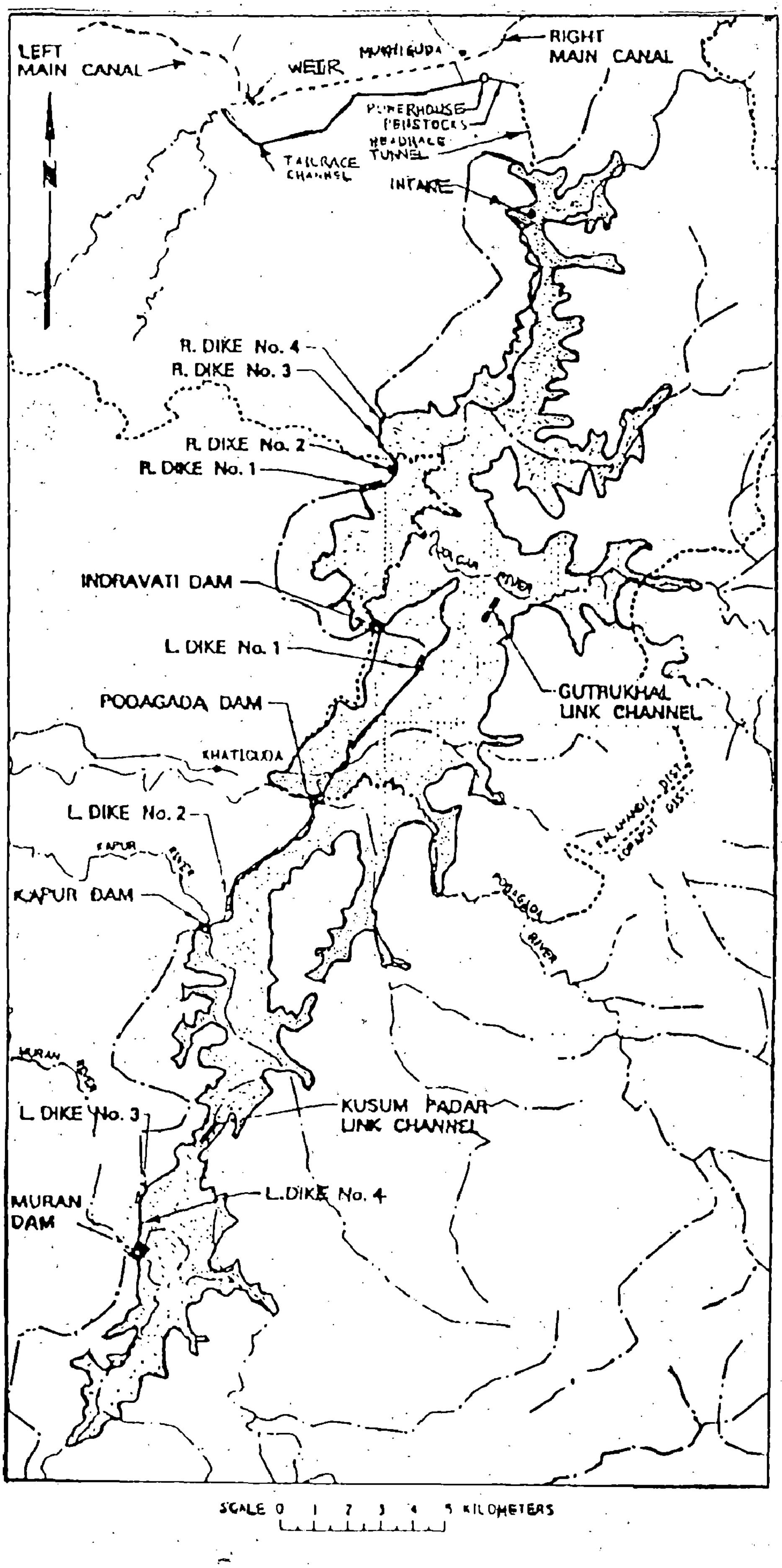


Fig 4.1 The Project map of UIHEP with Reservoir area

namely, gross reservoir capacity, dead storage capacity etc. and the installed generation capacity are kept unchanged in the present study.

4.2.1 Monthly Irrigation Demand

The primary objective of this study is to maximize the energy generation, through various models. However as the project is a multi purpose one, i.e., irrigation demand also needs to be fulfilled, the same has been accounted for by adding suitable constraints into the DP model and will be discussed in subsequent chapters. Monthly irrigation demand information is an essential input for all such models, which take irrigation planning into consideration. A water demand pattern, determined by maximising the net returns of crops, subject to constraints as appropriate for the project system, which has earlier been studied through LP model formulation [*Mohanty, 1994*], has been considered for the present study.

4.2.2 Inflow Data

For the purpose of data requirement, the present study capitalises upon the river inflow record, which was available for 32 years [*Sedimentation assessment study, 1995*].

4.2.3 Reservoir Operation Policy

To start with, a DP model, based on DDDP algorithm is used to study the operation of reservoir for optimum power generation, during the entire period of 32 years. A suitable loss function is identified for simultaneously minimising the irrigation deficit. Dividing the DP result into calibration and validation phases, DPR and DPN models are fitted to the calibration series and finally the system is simulated, during the validation period, to study the performance.

4.2.4 Time Series Modelling

A time series model, through Box & Jenkins approach is identified for the calibration period, (which has been kept different from the previous study), parameters are estimated and a forecast model is prepared. Simultaneously an ANN model is identified and trained for forecast. One-month- ahead forecast results, obtained from both the models, during the validation period, are tested through a customised time series simulator, developed for this study.

4.2.5 Summary

In the final step, a synthesis of above two sets of models, viz: reservoir operation policy and simulation and one-month-ahead river inflow forecast, has been attempted, to obtain a one month ahead predicted inflow based reservoir operation.

Dynamic Programming Model Formulation and Computation

5.1.0 GENERAL

The discrete differential dynamic programming, which is an iterative enumeration technique, is considered for the present study. The recursive equation of dynamic programming is solved within a restricted set of quantised values of the state variables by providing initial state of the reservoir. The optimal solution is obtained by gradually improving upon the initial solution. This technique is particularly suitable for invertible systems. A system is called invertible if for that system, the order of the state vector is equal to the order of the control vector.

5.2.0 MODEL FRAMEWORK

The primary objective of the DP model, herein, is to maximise total energy production of Upper Indravati Project system, subject to typical system constraints. Hydro-electric power, during a certain period of time is proportional to the product of total release, and the average operating head during that period. It is apparent that keeping the release constant, a marginal increase in the operating head can also result in an increased power production. This can be accommodated in the objective function by introducing a bias term for higher reservoir pool, during any time period. But This might increase the likelihood of increasing the spillage; thus there may be a trade off between the two aspects of operation.

The secondary objective of the model is to minimize the irrigation deficit, which, unlike the power generation, is dependent only upon the release amount and not on the average operating head. Therefore the optimum operation should balance the losses due to spill and deficit in meeting irrigation demands on one hand, and on the other, the gain of maintaining the reservoir pool at the highest possible level, during the entire optimization horizon.

The dynamic programming is not basically tailored in such a fashion that generalized programs can be written using it. Thus a new computer program has to be developed or an existing program has to be significantly modified and tested for each new application of the technique. The discussion made in the above paragraphs offered a tricky problem to be resolved. The problem is of adopting the appropriate objective function and in identifying a suitable loss function, which can accommodate both the objectives, viz: maximisation of power and minimisation of irrigation deficit. This was finalised after a trial and error procedure by first choosing a set of functions, then running the DP model and finally evaluating the model performance as per certain criteria and by repeating the entire procedure after altering the functions. Detailed discussion on this aspect is made in subsequent sections.

5.3.0 FORMULATION OF INITIAL OBJECTIVE FUNCTION AND CONSTRAINTS

5.3.1 Objective Function

A DP model having power generation as the primary objective, may sometimes perform better optimisation by introducing a bias term for maintaining higher reservoir elevation, during any time period 't'. Keeping this in view, the objective function of the proposed optimization model has been kept initially as

$$Max\sum_{t=1}^{N}\frac{Av_{head}(t)*Rel(t)}{10^6}+\frac{Av_{st}}{10^4} \qquad (5.1)$$

Where,

$Av_{head}(t)$ Average head during time period t, i.e., (average pool elevation - TWL) after accounting for head losses.

Rel(t) Release in Mcm during period t.

N. Number of time periods within the optimization horizon.

Av_{st} Average storage, obtained from the initial and final storage during the period t.

The second term in this objective function represents the preference to keep higher elevation level as discussed previously.

5.3.2 Constraints

The constraints, that are needed to properly define the system environment and functioning, can be divided into two groups, namely, those which represent the inherent system characteristics and will not change during the optimisation and the others, which are hypothetical loss functions or penalty functions, and may need modification during the sensitivity analysis, while simultaneously optimising more than one objective. The constraints are discussed below.

i. *Water mass-balance equation.*

$$S(t+1)=S(t)+Inf(t)-Elos(t)-Rel(t)-Spil(t) \qquad (5.2)$$
$$t=1,2,\ldots,N$$

Where,

S(t) =the state variable at the beginning of the time period t.

Inf(t)=Reservoir inflow during period t.

Elos(t)=the t*th* period reservoir evaporation loss.

Spil(t)=the spillage from the reservoir after the MWL has reached and power generation is at the peak.

ii. *The state variable s(t) can fluctuate only between the gross capacity of the reservoir and the dead storage of the reservoir.*

$$S_{min} \le S(t) \le S_{max} \qquad (5.3)$$
$$t = 1, 2, \ldots, N$$

iii. *Generation of hydro-electric power is proportional to release and the operating head and is governed by power equation.*

$$Gen(t) = c*9.8*Rel(t)*Av_{head}(t)*\eta \qquad (5.4)$$

Where,

Gen(t)= Energy generated in time t.

c =A constant for converting release in Mcm to cumecs.

η =Efficiency of power plant (i.e., turbine efficiency, and generator efficiency etc.)

iv. *The hydro power generation should be limited to the installed capacity for the said period. For the present study, load factor has been taken as 100% throughout the optimization horizon, because of non availability of the project load curve. the constraint is,*

Gen(t) ≤ Installed capacity (600 MW)

$$t = 1, 2, \ldots, N$$

v. *The penalty function.*

If release is less than the demand, then

$$ben(t) = -\frac{(Rel(t) - Dem(t))^2}{10^8} \qquad (5.5)$$

Where,

ben(t) benefit during time period t.

Dem(t) demand during time period t.

(-)ve symbol converts the benefit into a loss.

5.3.3 Identification of Appropriate Objective function and Penalty Function Through Trial

The trial run was made with the above set of objective function and constraints for the entire period of 32 years of historic inflow record available. But as the release from the reservoir, is first utilized for power generation and subsequently routed through the barrage structure and canal system, it could not be conclusively decided, whether both the system objectives are complimentary or competing with each other. So, a number of objective functions and corresponding penalty functions are formulated and those have been incorporated into the computer program, written in FORTRAN-77, by suitably modifying them. After running the model in each case, the results are compared for choosing the best set of functions.

The strategy adopted for altering the objective functions was to either include or exclude the bias term for higher storage (i.e., higher pool elevation) in the reservoir. Corresponding to each such case, the strategy adopted for formulating the penalty function was to vary the amount of penalty, in case there is a deficit. Conditional penalties have also been imposed into the objective function by segregating the desired objective, to test the model performance.

The criterions adopted for the performance appraisal of various DP models are, spillage, total generation during the optimization period, irrigation deficit and mean square deficit. A summary table of the results obtained from some competing model runs is given in Table (5.1).

Table :5.1

DP performance appraisal for alternative objectives and penalty functions

Alternative No.	Spill MCM	Generation (MW)	Deficit (MCM)	Mean Square Deficit (MCM)	Objective Function	Penalty Function
1	3524	100376	2590	4737.1	(Av_st + h_av*rel) ---------- 10^6	-(rel-dem)^2 ----- 10^{11}
2	2512	100969	5253	9081	(Av_st + h_av*rel) ---------- 10^6	If water is available & rel < dem -ben- (h_av*(rel-dem)^2) ---------- 10^6 else -(rel-dem)2 ------- 10^{11}
3	3795	99641	2326.4	2362.8	Av_st ------ 10^4 + (h_av*rel) ------------ 10^6	-(rel - dem)^2 ----------- -10^8
4	3607	99914	2326.3	2361.9	(h_av*rel) ----------- 10^6	-do-
5	2265	101217	2309.3	2484	Av_st ------ 10^6 + (h_av*rel) ------------ 10^6	- benefit - h_av*(rel-dem)^2 ---------- 10^6
6	**2242**	**101248**	**2309.3**	**2484**	**(h_av*rel)** **----------** $\mathbf{10^6}$	**-do-**

The model selected for further computations of present studies shown in bold. Two typical inferences pertaining to the

Indravati Project System , can be drawn, by observing at the table above.

1. Adding the bias term for higher storage into the benefit in the objective function, has resulted in higher amount of spillage and corresponding reduction in power generation, whereas the deficit remains unaffected.

2. By adopting stringent penalty norms for model not meeting the irrigation demand, not only the deficit is reduced, but also it has resulted in further reduction of spillage and increase in power generation.

The finally adopted objective function is

$$Max \sum_{t=1}^{N} \frac{Av_{head}*Rel(t)}{10^6} \qquad (5.6)$$

and the hypothetical loss function is
If Release < Demand, then

$$ben(t) = -ben(t) - \frac{Av_{head}*(Rel(t)-Dem(t))^2}{10^6} \qquad (5.7)$$

5.4.0 BASIC INPUT DATA

Monthly inflows into the reservoir, for a period of 32 years, obtained from [*Sedimentation assessment study, 1995*], constitutes the basic data, upon which the entire study is based. The same is furnished in Table (5.2). The inflows into the reservoir, given in cumecs, are to be converted to Million cubic meters per month. a source code written in C language, was used for the conversion. Monthly evaporation losses from the reservoir have been taken from the (Project report-1976) and the monthly irrigation demands for different crop plans,

Table -5.2 : Inflows into the Indravati reservoir during the period 1951 to 1982 in cumecs

Year	Jan	Feb	Mar	Apr	May	June	July	Aug	Sep	Oct	Nov	Dec	Annual Average
1951	26	21	17	18	17	120	242	328	133	45	45	36	87.33
1952	28	22	18	18	18	10	248	393	184	77	48	38	91.83
1953	34	28	23	23	23	12	259	365	357	149	59	47	114.92
1954	30	24	20	20	29	173	146	479	117	81	52	41	100.25
1955	26	21	17	17	17	55	141	222	359	91	45	36	87.25
1956	27	22	18	18	18	54	198	253	246	152	47	37	90.83
1957	24	19	16	16	16	54	143	262	242	92	41	33	79.83
1958	31	25	20	21	20	70	278	419	144	108	53	42	102.58
1959	29	24	19	20	19	55	236	276	228	173	50	40	97.42
1960	27	21	17	18	17	77	136	318	261	92	46	36	88.83
1961	28	23	19	19	19	169	229	267	200	83	49	39	95.33
1962	30	24	20	20	20	89	225	275	284	128	52	41	100.67
1963	24	18	17	20	20	42	264	372	153	66	40	31	88.92
1964	24	20	13	11	10	82	102	290	267	184	50	49	91.83
1965	29	24	23	21	24	50	161	302	139	113	48	34	80.67
1966	29	20	16	20	13	28	181	109	120	48	34	29	53.92
1967	16	10	10	12	22	80	209	218	332	37	24	19	82.42
1968	11	8	8	10	9	201	179	335	132	50	20	13	81.33
1969	17	20	14	12	18	25	75	128	150	150	42	26	56.42
1970	26	21	17	15	13	65	432	273	156	59	48	40	97.08
1971	27	20	15	17	23	57	180	256	180	94	42	35	78.83
1972	21	19	14	11	7	97	126	237	233	74	39	31	75.75
1973	25	21	17	17	17	27	234	167	336	82	44	35	85.17
1974	30	51	26	18	45	59	120	167	159	78	30	23	67.17
1975	20	15	14	20	18	142	215	314	115	76	31	22	83.50
1976	23	27	25	19	22	54	149	230	135	49	27	22	65.17
1977	22	26	20	19	23	61	160	277	97	68	39	19	69.25
1978	17	13	13	14	23	23	124	104	137	56	45	29	49.83
1979	25	24	14	17	14	28	79	130	106	58	32	16	45.25
1980	20	25	12	8	8	129	116	277	246	47	34	26	79.00
1981	19	30	25	25	76	116	306	270	317	159	84	38	122.08
1982	25	31	24	20	19	12	237	461	525	440	87	69	162.50
Mean	24.69	22.41	17.53	17.31	20.25	72.38	191.56	274.19	212.19	101.84	44.59	33.50	86.04
Sdv	5.06	7.23	4.38	3.91	12.13	49.12	74.35	93.76	97.82	73.75	14.02	10.99	22.28
Skewness	-0.73	1.68	0.10	-0.67	3.49	1.02	0.97	0.15	1.24	3.31	1.28	0.73	1.04
Kurtosis	0.45	7.52	-0.24	0.12	15.02	0.54	2.05	0.01	1.77	14.22	3.26	2.34	3.60

taken from [*Mohanty, 1994*], are furnished in Table (5.3).

Reservoir elevation vs. area vs. capacity curves, in discretized form, at an elevational interval of 5m is furnished in Table (5.4). Reservoir dead storage and gross storage capacity have been taken as 814.5 and 2300 Mcm respectively. An initial reservoir elevation of 1500 Mcm has been considered for the study. For non availability of the load-curve, a uniform load factor of 100% has been assumed for all the time periods of optimization horizon.

For computing the operating head of the generator, a constant tail water elevation of 265.00m, with an average head loss of 12m has been taken for the present analysis. A uniform turbine efficiency of 92.0 % and generator efficiency of 97.5 % have been used. The above assumptions related to the power generation are based upon the computations made by HARZA Consultants [*Project report, 1994*].

5.5.0 COMPUTATIONS

The computation is carried out by a FORTRAN program, consisting of a main program and two sub-routines namely, DDP, which carries out the discrete dynamic programming computations and a sub-routine BENEF, which is a user supplied sub-routine, meant for evaluating the objective function. A function FINT is also used for linear interpolation, to find the value of the dependent variable (y) corresponding to the independent variable (x), from a table of pairs of (x) and (y) values.

For the purpose of computations, the entire active storage region is divided into a number of divisions. The optimal state trajectory is searched from among the feasible states so that the objective function is maximised.

Table 5.3 : Showing the elevation-area-capacity relationship for the Indravati reservoir

Serial No.	Elevation in M	Area in sq. km	Capacity in cub. met.
1	580	240000	0
2	585	450000	1700000
3	590	2740000	11280000
4	595	5600000	40000000
5	600	10700000	74160000
6	605	16800000	140000000
7	610	23950000	237690000
8	615	33290000	380289984
9	620	44330000	570540032
10	625	59160000	827150016
11	630	74590000	1152960000
12	635	89970000	1562599936
13	640	104870000	2052120064
14	645.5	122800000	2710000128

Table 5.4: Monthly irrigation demand and monthly average evaporation

Serial No.	Month	Irrigation Demand	Evaporation (MM)
1	January	231.9	75
2	February	276	150
3	March	247.3	175
4	April	163.2	200
5	May	115.1	200
6	June	86.4	175
7	July	284.9	75
8	August	40.8	75
9	September	317.9	75
10	October	283.4	100
11	November	106.8	100
12	December	88.9	75

5.6.0 DISCUSSION

Reservoir releases after optimisation through the finally accepted DP model are furnished in Table (5.5) along with the monthly demands, during the entire period of 32 years. By altering the objective or the hypothetical functions, it was seen that total deficit is not varying significantly. It was further observed that by increasing number of iterations in succcessively reducing the corridor width, total deficit is distributed among larger number of time periods with smaller deficits. Corresponding power generations and reservoir storages are shown in Tables (5.6) and (5.7), for comparison with various model performances, to be discussed in subsequent chapters.

TABLE 5.5 : Optimal monthly DP releases during historic inflow record of 32 years

YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	Annual generation
1951	203.7	247.5	218.4	133.8	85.3	86.4	289.1	54.2	336.6	310	109.3	115.5	2189.8
1952	240.8	286.8	254.1	184.5	157.5	138	461	55.4	345	319.2	113.8	93.8	2649.9
1953	330.9	301	293.9	167.3	170.2	142.6	487.9	352.6	422.7	388.4	142.3	117.9	3317.7
1954	320.2	291.3	285.8	283.8	163.3	188	385.2	223.3	357	329.9	124.1	101.8	3053.7
1955	309.5	284.1	277.8	226.3	154.8	130.4	298	93.2	427.9	357	151	94.6	2804.6
1956	262.2	287.1	281.3	200	158.4	128.6	307.2	52.2	382.5	347.1	111.2	91.2	2609
1957	274.5	279.2	275.1	179	151.7	127.4	303.2	76.3	372.1	285.6	170.4	155.2	2649.7
1958	249.2	294.7	287.2	163.7	164	95.7	367.5	186.3	364.9	340.2	126.4	104.2	2744
1959	317.3	290.8	282.4	282.9	159.5	129.7	378.7	113.3	335.4	390.2	118.4	98.8	2897.4
1960	250	282.9	276.4	276.8	153.2	185.9	358.4	102.1	420.9	359.4	108.6	162.8	2937.4
1961	240.7	288.9	283.2	281.2	160.6	177.7	359.6	88.9	386.5	335.3	116.4	96.5	2815.5
1962	320.2	291.3	285.8	283.8	163.3	218.9	349.6	110.9	356.9	331.8	123.8	101.6	2937.9
1963	303.9	276.2	277.1	282.9	162.2	96.1	329.7	184.1	388.2	283.4	111.6	149.3	2844.7
1964	274.5	281.6	267.1	185.9	135.8	105.6	317.1	101.8	357.6	301	118.4	122.9	2569.3
1965	250.4	290.2	247.8	186.8	141.1	87.9	300.8	75.7	351.9	291.5	113.3	85.4	2422.8
1966	228.3	272.5	243.8	159.6	111.4	82.7	280.9	36.7	313.2	278.6	99.3	81	2188
1967	224	268.1	239.2	154.9	106.7	86.4	284.9	50.9	330.9	291.4	128.1	90.8	2256.3
1968	269.8	280.4	255.2	180.2	124.5	92.7	291.2	52.9	333.8	271.9	95.3	77.4	2325.3
1969	220.3	264.3	235.5	151.2	103	74.4	272.8	29.2	306.4	272	95.5	77.5	2102.1
1970	220.4	264.6	235.7	151.4	103.2	86.5	294.2	166.2	396	283.4	106.7	92.2	2400.5
1971	246.4	281.1	271.7	195.5	124.9	89.3	304.5	59.5	334.7	288.5	142.7	112.5	2451.3
1972	239.6	279.1	269.6	165.8	127.3	114.3	287	83.3	348.7	311.6	135.5	120.1	2481.9
1973	233	284.8	261	179.7	127.1	87	295.7	42.7	327.7	286.9	113	91.5	2330.1
1974	232.4	278.4	247.8	166.5	116.4	87.5	285.5	41.3	320.7	288.9	107.7	89.1	2262.2
1975	234.2	280.2	247.6	164.6	115.1	88.5	287.4	45.6	305.4	270.8	94.2	76.3	2209.9
1976	219.2	263.2	234.2	149.9	101.5	72.7	271	27	303.9	269	92.1	70.3	2074
1977	212.4	256.2	225.9	140.9	92	63.1	260.4	16.6	293.9	255.8	73.3	55.1	1945.6
1978	196.3	239.1	209.9	125.4	76.8	47.8	246	1.8	278.4	243.4	66.6	48.3	1779.8
1979	189.4	231.7	202.2	117.5	68.5	39.5	237.5	0	268.2	233.3	54.6	36.5	1678.9
1980	175.5	219	189.3	104.3	55.3	86.4	285	41.6	355.8	474.5	156.2	93.2	2236.1
1981	246.7	276.1	247.3	163.2	115.1	86.5	293.2	96.6	469.4	414.7	206.6	93.4	2708.8
1982	472.2	306.1	247.3	163.3	115.1	86.4	284.9	298.8	469.2	469.2	214.3	473.8	3600.6
Average	256.50	275.58	254.89	186.02	127.03	106.58	314.22	92.53	355.08	317.93	120.02	108.45	2514.84
Demand	231.9	276	247.3	163.2	115.1	86.4	284.9	40.8	317.9	283.4	106.8	88.9	2242.6

Table 5.6 : Monthly power generation after DP optimisation

YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	Annual generation
1951	254.4	307.4	269.3	163.9	104.2	105.9	358.5	68.5	429.4	394.6	138.8	146.5	2741.4
1952	304.8	361	317.6	229.1	194.6	169.7	567.6	69.6	440	407.1	144.9	119.4	3325.4
1953	420.1	379.6	367.9	208.2	210.9	175.8	600	440.2	536.3	495.7	181.6	150.4	4166.7
1954	407.5	368.3	358.9	353.5	202.1	233.2	479.8	282.6	456.2	420.7	158.1	129.7	3850.6
1955	392.9	358.2	347.8	281.3	191.3	160.7	367.9	116.2	541.4	454	191.6	119.9	3523.2
1956	331.5	360.7	350.7	247.7	194.9	157.8	378.8	65.3	484.6	441.5	141.6	116	3271.1
1957	348.3	352.1	344.5	222.6	187.8	157.3	375	95.4	471.5	362.8	216.1	196.4	3329.8
1958	314.3	369.5	357.1	202.3	201.6	117.5	455.4	235.5	466.6	434.8	161.4	133.1	3449.1
1959	404	367.9	354.9	352.7	197.7	160.4	470.2	142.6	427.1	498.7	151.4	126.3	3653.9
1960	319	359.2	348.6	346.5	190.7	230.8	445	128.3	536	458.3	138.3	207.1	3707.8
1961	305.2	364.3	354.5	349.2	198.1	219.6	448.3	112.3	492.9	427.5	148.2	122.8	3542.9
1962	406.5	367.4	357.9	352.4	201.4	269.2	432.2	139.1	453.9	423.8	158.1	129.7	3691.6
1963	387	349.5	348.2	352.7	201	118.7	410.2	233.1	496.4	361.9	142.3	190.2	3591.2
1964	348.3	355.2	334.4	231.1	168	130.6	392.5	127.3	453.6	384.2	151.4	157.1	3233.7
1965	319.5	368.4	312.7	234.6	176.5	109.9	376.8	96	450	372.7	144.8	109.2	3071.1
1966	291.4	346.3	307.8	200.5	139.5	103.4	352.2	46.3	396.4	351.8	125.1	102	2762.7
1967	281.3	334.1	295.7	190.1	130.3	105.6	351.2	63.6	420	371.3	162.7	115.1	2821
1968	340.6	351.4	317.1	222.2	152.6	114.3	363.1	67	426.8	347.1	121.4	98.4	2922
1969	279.4	333.3	295.2	188.4	127.9	92.2	337.7	36.3	383.1	341.2	120	97.4	2632.1
1970	276.3	329.5	291.3	185.8	126.1	105.7	366.1	211.2	506.4	361.8	136.1	117.6	3013.9
1971	313.6	355.7	341.5	244.2	155.3	111	379.9	75.1	426.6	367.9	181.8	143.2	3095.8
1972	304.3	352.2	337.9	206.4	157.8	141.7	356.7	104.6	442.4	396	171.8	152.1	3123.9
1973	294.2	357.5	325.1	222.4	156.5	106.8	365.4	53.4	415.5	365.8	143.9	116.6	2923.1
1974	295.5	352.4	312	208.6	145.5	109.5	357.6	52.1	406.8	366.5	136.4	112.7	2855.6
1975	295.3	351.2	307.8	203.4	141.5	109.3	358.3	57.8	390.5	345.9	120.2	97.3	2778.5
1976	278.9	333.2	294.8	187.9	126.9	90.8	339.6	34.2	387.6	342.7	117.2	89.4	2623.2
1977	269.2	323.1	283.3	175.8	114.5	78.6	325.7	21.1	375.2	326.1	93.3	70.1	2456
1978	249.6	302.5	264	157.1	95.9	59.7	307.5	2.3	351.6	307.2	84	61	2242.4
1979	238.7	290.7	252.3	145.9	84.9	48.9	294.1	0	335.7	291.6	68.2	45.6	2096.6
1980	218.7	271.6	233.2	127.7	67.6	106	351.5	51.9	450.6	600	196.3	116.9	2792
1981	308.4	342.8	304.7	199.8	140.8	106.5	365.6	122.5	600	530.3	264.2	119.4	3405
1982	600	385.5	309.4	203.1	142.7	106.9	354.1	378.5	600	600	274.1	600	4554.3
Average	324.96	346.93	318.69	231.16	157.10	131.69	390.14	116.56	451.60	404.73	152.67	137.77	3163.99

TABLE 5.7 : Monthly storages in Mcm after DP optimisation

YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	Annual generation
1951	1500	1359.6	1151.2	966	866.2	814.5	1028	1381.3	2198.4	2198.4	1998.3	1995.3	17457.2
1952	1968.6	1795.3	1547.7	1326.9	1173.4	1049.6	925.8	1123.8	2114.2	2238	2114.2	2114.2	19491.7
1953	2114.2	1866.6	1619	1371.4	1247.6	1123.9	1000.1	1200.9	1819.8	2300	2300	2300	20263.5
1954	2300	2052.4	1804.8	1557.3	1309.7	1185.9	1433.5	1433.5	2300	2238	2114.2	2114.2	21843.5
1955	2114.2	1866.6	1619	1371.4	1173.4	1049.6	1049.6	1123.8	1619	2114.2	1990.4	1945.8	19037
1956	1940	1742.8	1495.2	1247.6	1079.3	955.5	955.5	1173.4	1792.3	2039.9	2089.4	2089.4	18600.3
1957	2089.4	1871.6	1624	1376.4	1222.9	1099.1	1099.1	1173.4	1792.3	2039.9	1990.4	1916.1	19294.6
1958	1841.9	1668.6	1421	1173.4	1049.6	925.8	1000.1	1371.5	2300	2300	2238	2238	19527.9
1959	2238	1990.4	1742.8	1495.2	1247.6	1123.8	1123.8	1371.4	1990.4	2238	2300	2300	21161.4
1960	2300	2114.2	1866.6	1619	1371.4	1247.6	1247.6	1247.6	1990.4	2238	2114.2	2114.2	21470.8
1961	2039.9	1866.6	1619	1371.4	1123.8	1000.1	1247.6	1495.2	2114.2	2238	2114.2	2114.2	20344.2
1962	2114.2	1866.6	1619	1371.4	1123.9	1000.1	1000.1	1247.7	1866.6	2238	2238	2238	19923.6
1963	2238	1990.4	1742.8	1495.2	1247.7	1123.9	1123.9	1495.2	2300	2300	2182.4	2163.7	21403.2
1964	2089.4	1871.5	1624	1376.4	1203.1	1079.3	1173.4	1123.8	1792.3	2119.1	2300	2300	20052.3
1965	2300	2119.1	1871.5	1668.5	1518	1423.9	1450.6	1574.4	2300	2300	2300	2300	23126
1966	2297.3	2138.4	1898.7	1680.8	1554.8	1460.6	1435.5	1632.7	1880.8	1871.1	1711.4	1690.8	21252.9
1967	1680.4	1492.5	1236.2	1011	874.2	814.5	924.8	1194.5	1721	2242.9	2039.9	1963.7	17195.6
1968	1916.1	1668.5	1394.2	1146.6	978.3	865.2	1281.1	1463.1	2300	2300	2151.1	2097	19561.2
1969	2046.5	1864.1	1633.9	1420.3	1283.8	1213.4	1190.4	1112.9	1420.6	1496.6	1617.4	1621.7	17921.6
1970	1606.9	1449.4	1223.4	1020.5	895	814.5	886	1742.8	2300	2300	2163.7	2170.6	18572.8
1971	2177.4	1995.3	1747.7	1500.2	1331.8	1252.6	1297.1	1468.5	2087.4	2211.2	2163.7	2119.1	21352
1972	2092.4	1901.3	1653.7	1406.1	1252.6	1128.8	1252.6	1297.2	1841.8	2089.4	1965.6	1921.1	19802.6
1973	1876.5	1703.2	1455.6	1225.9	1075.3	980.1	951.5	1277.2	1675.3	2210.9	2132.8	2123.2	18687.5
1974	2117.4	1957.5	1787.5	1592.9	1455.4	1442.4	1492.7	1522	1920.8	2004.6	1914.4	1874.4	21082
1975	1839.5	1651.7	1394.5	1170.5	1043.2	962.9	1229.9	1512.2	2300	2284.3	2206	2181.3	19776
1976	2155.8	1990.3	1777.5	1593.9	1475.4	1415.7	1468	1589.5	2171.1	2209	2060.5	2027.9	21934.6
1977	2008.7	1847.7	1640.1	1452.1	1343.7	1297.1	1377.8	1539.5	2257.2	2206.6	2122.1	2139.2	21231.8
1978	2127	1968.4	1745.9	1554.6	1447.9	1415.7	1412.8	1492.5	1762.3	1831.7	1728.7	1769.2	20256.7
1979	1791.3	1661.7	1474.6	1295.3	1206.2	1160.1	1180	1148.5	1490.6	1490.6	1404	1423.9	16726.8
1980	1423.9	1295.8	1125.8	956.5	860.4	814.5	1051.2	1071.6	1765.6	2039.9	1681.5	1604.3	15691
1981	1574	1371.9	1156.7	964.4	853.7	930	1132.4	1652.6	2271.6	2300	2300	2300	18807.3
1982	2300	1887	1642.1	1444.4	1317.4	1238.5	1170.8	1514.9	2300	2300	2300	2300	21715.1
Average	2006.84	1808.97	1573.62	1350.73	1193.96	1106.54	1174.79	1367.78	1992.38	2141.51	2063.95	2049.08	19830.15

DPR and DPN Models

6.1.0 GENERAL

The assumption behind adoption of a dynamic programming with regression (DPR) model or dynamic programming with neural net (DPN) is that the deterministic dynamic programming result, spanning across the historical flow record of optimization horizon, encapsulates a definite pattern between the release and some or all of the independent variables like, initial storage, inflow and demand. This underlying pattern is unique to the system considered and can be filtered through some pattern recognition tool. The multiple regression models have been widely used in water resources system engineering for generating the operating rules. The regression models smoothen the values of the release function. A brief outline of the multiple regression is furnished in the next section.

6.2.0 THE MULTIPLE REGRESSION

When more than one independent variable is being used to predict the value of the dependent variable, the analysis is termed as multiple regression analysis. In case of regression, the best fit line is identified by the least squares method. The four basic assumptions underlying a multiple linear regression analysis are as follows,

Assumption 1. There exists some hypothesised underlying relationship among y and x_1, x_2, x_3, ..., x_p in the form

$$E(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots \quad (6.1)$$

where E(y) is the regression function.

In essence b_0, b_1, b_2 in equation (6.2) are estimates for $ß_0$, $ß_1$, $ß_2$ respectively.

Assumption 2.The distribution of y, x_1, x_2 follows a normal distribution.

Assumption 3.The variability of y about the regression function is the same for any choice of values of x_1 and x_2, i.e., variance is the same for each set of y values. This is called equal variance condition or homoscedastic condition and is indicated by σ_y^2.

Assumption 4.The observed Y values are based on a random sample from the assumed normal probability distribution and that each observation is independent of all other observations.

Mathematically the multiple linear regression analysis can be expressed as follows:
Total data length be N and Number of parameters be P.
Then degree of freedom (regression) =p
Then degree of freedom (error) =N-p-1
The estimated regression function is

$$\hat{Y}=b_0+b_1x_1+b_2x_2+.....+b_px_p \qquad (6.2)$$

b_0, b_1, b_2, ... , b_p are choosen by least square methods, with the objective of

$$Min\sum(y_i-\hat{y}_i)^2 \qquad (6.3)$$

The following parameters are used to judge how good is the regression line.

$$SSE=\sum (y_i-\hat{y}_i)^2 \qquad (6.4)$$

$$SST=\sum (y_i-\overline{y})^2 \qquad (6.5)$$

$$SSR=\sum (\hat{y}_i-\overline{y})^2 \qquad (6.6)$$

SST = SSE +SSR and for best fit SSE ≈ 0;SSR/SST ≈ 1 ≈ r^2;
MSE = SSE/ Error DF ; MSR = SSR/ regression DF.

SSR measures the amount of variability in y explained by the regression model. The sampling distribution of MSR/MSE follows a F-distribution.

F = MSR/MSE ; when computed F > $F_{.05}$(p,n-p-1), a linear relationship is said to exist.

Multiple non-linear regression is also carried out along the same steps, with some modifications. Interested readers can refer to any standard text book on multi variate regression analysis for this.

6.2.1 Computation

Under the DPR model category, two regression procedures have been attempted, namely, multiple linear regression and multiple non-linear regression. Although authors on this subject have all along suggested a simple linear form to express the optimal release [*refer Chapter-2*], the non-linear regression equation was tried in this study, to assess its relative performance in comparison to other models.

Before deriving general operating rules using regression from deterministic DP results, the DP output was divided into two sets. The result for first 27 years, consisting of 324 patterns was considered for calibration and last 5 years result consisting of 60 patterns was kept aside for validation. The calibration data set was processed through a WINDOWS based statistical package called SYSTAT, for obtaining the multiple linear and non-linear equations.

The expression adopted for linear regression equation is

$$Rel(t)=a_0+b_1*S(t)+b_2*Infl(t)+b_3*Dem(t) \qquad (6.7)$$

and the expression adopted for non-linear regression equation is

$$Rel(t)=p_1*(S(t))^{p_2}+p_3*(Infl(t))^{p_4}+P_5*(Dem(t))^{p_6} \qquad (6.8)$$

The summary result of the output is tabulated below.

Table (6.1) : Regression Analysis

Multiple linear regression

Variable	value	Standard-error	T
Storage	.022	.006	3.76
Inflow	.086	.009	9.43
Demand	.95	.027	35.097
Constant	-23.464	10.189	-2.303

Analysis of variance

Source	SS	DF	MS	F-ratio
Regression	3415177.5	3	1138392.5	514.395
Residual	708179.52	321	2213.061	

Multiple Non-linear Regression

Convergence attained after 200 iterations;

Precision	0.01
Raw R-squared (1-residual/total)=	0.958
Corrected R-squared (1-residual/corrected)=	0.794

Analysis of variance

Source	SS	DF	MS
Regression	$.203894 \times 10^{8}$	6	3398234.383
Residual	738740.08	318	2323.082

Hence the regression equations obtained for linear and non-linear cases are as follows.

$$Rel(t) = -23.464 + .022*S(t) + .086*Inf(t) + .95*Dem(t) \quad (6.9)$$

and

$$Rel(t) = 1.875\ S(t)^{0.461} + .011\ Inf(t)^{1.269} + .105\ Dem(t)^{1.366} \quad (6.10)$$

These equations have been built into the simulation model and detailed discussion on these results will be made in Chapter-8. For evaluating the relative performance of these equations, on the validation data, these are compared with the output of the DPN model, which will be discussed subsequently.

6.3.0 DPN MODEL

In the context of previous discussions, the problem of deriving an optimal release policy has been reduced to searching for an appropriate function approximation for optimal release, by some suitable pattern recognition tool. This has been tried with an artificial neural network architecture.

6.3.1 Identification of N.N. Architecture

Proper identification of the neural network architecture and topology is of prime importance to neural network modelling. Before the training process begins, the architecture needs to be identified, with the following design considerations:

- The structure of the system i.e., numbers of layers
- The synchrony of the system i.e., the mode of control and synchronisation of the processors.
- Symmetry of inter connections.
- Feed forward/feed back structure employed.
- Transfer or activation function relating input to output.
- Learning strategy.

However, all of the above cannot be finalised at the beginning. The learning strategy especially may need modification, through the process of training and testing, when the model as well as the modeller, both gain experience by successively running the model. This is in fact a trial and error procedure.

For the present study, a feed forward , error back propagation network has been adopted under supervised learning mode, as it is, so far the most popular of the ANN architectures available. The structure of the ANN consisted of three layers - the input layer having 3 neurons, the output layer with one neuron and a hidden layer with four neurons. The number of neurons in the hidden layer was decided after a rigorous course of training and testing the data during the calibration and validation period. Neurons in the input layer acted simply as buffers through which input data was sent.

Regarding connectivity, every neuron in the hidden layer is connected to all the neurons in the preceding layer and with the output layer neuron. A BP simulator, coded in C++, is implemented for identification, training and testing of the neural network, under an UNIX environment.

6.3.1.1 Activation function

The sigmoid function is used for the activation function. This function is the most commonly used nonlinear activation function. The basic characteristics of the sigmoid function is that it is continuous, differentiable everywhere, it is monotonically increasing. The output y_j is always bounded between 0 and 1 and the input to the function can vary between $\pm\infty$. Under this threshold function, the output y_j from a neuron in the hidden layer becomes

$$Y_j = f(\sum w_{ji}x_i) = \frac{1}{1+e^{-(\sum w_{ji}x_i)}} \qquad (6.11)$$

here

W_{ji} = Weight of the connection, connecting jth neuron in the hidden layer with ith neuron in the input layer.

x_i = Value of the i^{th} neuron in the preceding layer.

y_j = Output from the j^{th} neuron in the present layer under consideration.

6.3.2 Learning Algorithm

The learning algorithm adopted for the network is of supervisory mode, batch processing type, following the generalized delta rule, [Rumelhart, 1986]. Learning in fact, in the neural network parlance , refers to gradual adjustment of the inter connection weights within the network, to minimize the error between the ANN output and the output pattern used for training. This process is repeated many times with many different input/output tuples until a sufficient accuracy for all data sets has been obtained.

This adjustment of the inter connection weights during training, employs a method known as error back propagation. In this the weight associated with each neuron is adjusted by an amount proportional to the strength of the signal in the connection and the total measure of the error. The total error at the output layer is then reduced by redistributing this error value backwards, through the hidden layers until the input layer is reached. this process continues for the number of sweeps prescribed by the modeller or until reaching a prescribed error tolerance level. In this way the back propagation algorithm can be seen to be a form of gradient descent for finding the minimum value of the multi dimensional error function. A detailed step by step account of the practical problems encountered during training and testing of the ANN is discussed in the subsequent sections.

6.3.3 Momentum, Noise and Shuffling

ANN objective function surface is typically non-convex, which contains multi local optima. It has extensive regions that are insensitive to the variations in the network weights, imposed by the *generalised delta rule*, discussed in Chapter-3. This results in some major limitations of the BP algorithm, such as,

1. These are easily trapped by local optima.
2. The convergence is an extremely slow process.
3. The architecture is often ineffective, when searching weight spaces of high dimension.
4. Performance of BP-ANN simulator is quite sensitive to the initial starting point.

Many new algorithms have come up to improve upon the BP-ANN performance and to counteract above limitations. In the present study, the first and second limitations are managed by adding momentum and noise features.

6.3.3.1 ***Momentum***

Adding a momentum term to the earlier described training law, *sometimes* results in much faster training. This term determines the effect of previous weight changes on the present change in the weight space. Therefore, in addition to improving the convergence speed, this *sometimes* enables in dragging a solution trapped by local optima, as this keeps the weight change process moving. The weight change with inclusion of a momentum term is expressed as,

$$\Delta W_{ji}(s) = \eta . \delta_j x_i + \alpha \Delta W_{ji}(s-1) \qquad (6.12)$$

Where

η = learning rate; which provides the step size during the gradient descent.

α = a momentum rate term (for this study values between .5 and .9 have been taken.)

s = sweep; one cycle of training using the complete batch of input pattern set.

6.3.3.2 ***Noise***

This is another approach to breaking out of local minima, whereby a noise or a random number is added to each input component of the input vector as it is applied to the network. Provision was kept in the simulator to send the noise to input patterns, within a range wished by the user.

6.3.3.3 ***Shuffling***

In order to counter the last limitation, authors adopt mostly, an initial randomized weight space to start with. This helps in breaking the symmetry. In case the convergence is slow or found to be locked up, usually the weight matrix is broken and a new initial weight matrix randomized between -1 and +1 is given. This process continues until convergence is visible during running of the simulator. When this also doesn't work, the input patterns are changed, another set is presented to the simulator.

In the present study, in place of presenting a new set of input pattern, a shuffling strategy is adopted whereby, an algorithm is generated which simply shuffles the existing input pattern and re-sends the same to the simulator. This has been found to be very effective to desensitise the input pattern sensitivity of the BP algorithm.

6.3.4 Preparation of Input and Output Data Patterns

In this study, input to the network are initial storages, inflows, and demands while optimal release is the output. These values during the calibration period of 27 years, are to be filtered from the DP output file, which has a [384 X 11] data matrix structure. Further these data cannot be sent, as such, to the BP simulator, because the sigmoid function bounds the output between 0 and 1. Usually the strategies adopted for this are scaling, normalising and standardising, scaling being the simplest. Since the sigmoid function does not impose any restriction on input patterns, it is a trial and error procedure to select proper scaling factors for input patterns and output response in such a way that the BP will result in speedier and better convergence.

6.3.4.1 Data Manipulator Program

All above requirements on frequent data manipulation warrant the development of a customised data manipulator program. The same is developed, coding through C++ with the following features.

- Usage data_manip <input.filename> <output.filename>
- Accept a 2-dimensional data matrix through input file.
- Obtain user options for seasonalisation and the data columns where manipulation is desired.
- Show the max. value of each desired data column.
- Scale down the selected columns interactively and rearrange them as desired.
- Optionally shuffle the created data set keeping the sequence of shuffling in a file <ref.out>.

  The program is a generalized one and can be used for similar purposes. The code is given in Annexure-III.

6.3.4.2 Scaling

In this case the ultimate scaling factors adopted for the input patterns and output responses are 2500 and 500 respectively. These values are determined after studying different combinations. These scaling factors provided faster convergence compared to others including the one, where the factors were calculated to restrain the patterns exactly between 0 and 1.

6.3.5 Training Strategy

It was observed during training that, the rate of convergence is very fast during initial sweeps, but after some sweeps, it is either static or very slowly converging. Keeping this fact in mind a phased training schedule is adopted. Under this schedule the training process was carried out with a higher learning rate and noise term and (*depending upon model acceptance*), a momentum term, for approximately 500 cycles with randomization of initial weight matrix. The weight matrix obtained after initial training is considered as initial solution, in the next phase of training. As a lower learning rate has the ability to skip the hurdles of local optima, although the convergence is slow, gradually reduced learning parameters and smaller noise terms were taken during successive training.

The number of neurons in the hidden layer of the architecture is finalised after trial and error procedure. If the architecture is too small, the network may not have sufficient degrees of freedom to learn the process correctly. On the other hand, if the network is too large, it may not converge during training or it may overfit the data [*Raman and chandramauli, 1996*].

For obtaining the optimum number of neurons in the hidden layer, 10 different structures were prepared, starting from 3-1-1 up to 3-10-1. Every structure is made to learn till saturation the system characteristics, by the phased training strategy, described in the previous paragraph. The performance of each, was finally tested through a comparison model, developed in FORTRAN-77 for comparing the RMS error of the ANN structures with those of multiple linear and multiple non-linear regression. It was found that the number of neurons in addition to four, although attain higher minimisation level during training, fail to map properly the unseen testing data during the validation period. Therefore, the architecture of 3-4-1 was finalised. The chosen network along with the weight matrix is given in Figure (6.1).

6.3.6 Validation

The BP simulator used for this study has two options : 1 for training, 0 for testing. By activating the testing module the input patterns during the validation period is simulated. The output scaled between 0 and 1 was filtered through another program written in C++ and the output is routed through the comparison program COMPARE.F. This entire job was entered into a batching program, for ease in successive trials.

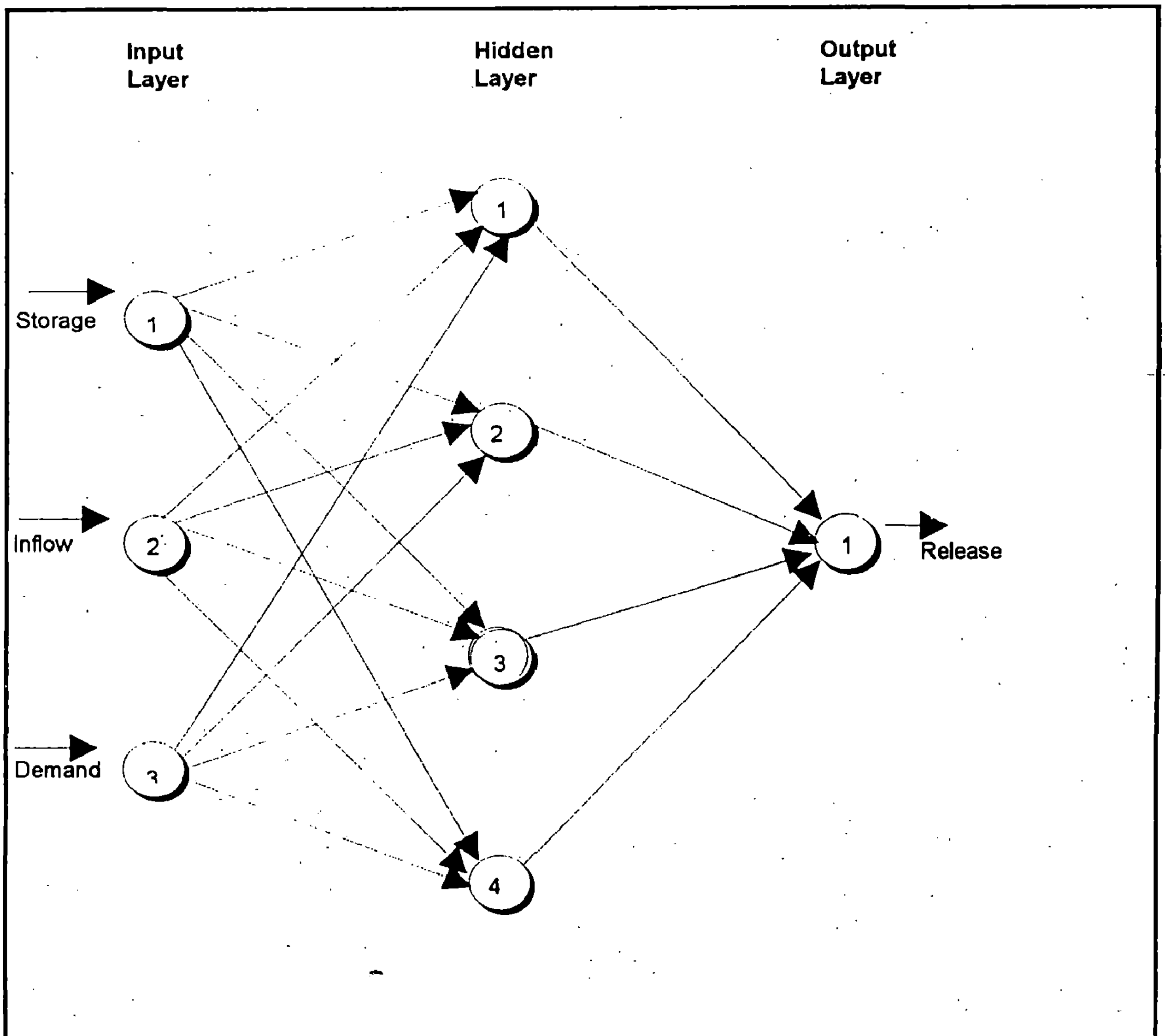


Fig 6.1 : The ANN architecture selected for modelling the reservoir release

The Weights Matrix (Network Architecture 3 - 4 - 1)

1	0.211532	-3.35254	-0.45647	-1.24797
1	-2.73996	0.267676	-0.77974	4.374992
1	-22.1786	11.01551	-53.6145	3.429683
2	-3.45711			
2	-2.99356			
2	-10.0251			
2	3.667947			

6.3.7 Discussion Of Results

The releases from multiple linear, multiple non-linear regression and the ANN models are tabulated in Table (6.2) along with the DP model during validation period of 60 months. The model releases are plotted along with demand in Figure (6.2) and along with release in Figure (6.3). From the plottings it can be seen that the ANN mapping of optimal release is better, compared to other two models. For better comparison of the estimated releases, statistical properties along with RMS Error, evaluated through the comparison model, are tabulated in Table (6.3).

The RMS Error for various models during the validation period is computed by the expression

$$E_{rms} = \frac{1}{60} (DP_{release} - Model_{release})^2 \qquad (6.13)$$

It can be inferred from the tabulated information that the ANN model release has least RMS error; it is better correlated with DP release, whereas the regression model releases are more correlated with the demand. Regarding the other statistical properties like mean, SD, skewness etc., all the models have more or less equal performance.

Table - 6.2 Comparison of models during the validation period of 5 years, from 197 9to 1982 (values are in Mcm)

ACTUAL DEMAND	D.P. RELEASE	MODEL RELEASES		
		LINEAR	NON-LINEAR	ANN
231.9	196.3	247.548	244.2607	267.6945
276	239.1	284.7412	289.4685	276.3455
247.3	209.9	252.8736	254.6849	261.6215
163.2	125.4	168.899	167.1575	203.9365
115.1	76.8	123.0324	124.4151	140.059
86.4	47.8	94.887	101.525	85.8345
284.9	246	306.8332	307.2799	306.529
40.8	1.8	72.0906	85.05382	35.2335
317.9	278.4	347.8502	352.7576	329.3845
283.4	243.4	298.9634	301.2706	294.9605
106.8	66.6	126.055	124.8781	143.7365
88.9	48.3	106.5956	109.9094	100.7125
231.9	189.4	242.0116	240.2641	264.3125
276	231.7	280.29	285.8531	269.0475
247.3	202.2	247.1372	250.3979	249.4615
163.2	117.5	163.8652	162.9733	194.993
115.1	68.5	115.6424	119.1183	126.2575
86.4	39.5	90.38181	97.4225	80.7715
284.9	237.5	291.3486	295.464	268.9715
40.8	0	70.5082	83.41406	36.342
317.9	268.2	334.967	343.1243	300.745
283.4	233.3	291.915	296.1315	279.076
106.8	54.6	116.0134	117.9264	128.399
88.9	36.5	96.0062	102.8386	87.86
231.9	175.5	232.7764	233.7511	245.494
276	219	272.4466	279.7533	247.893
247.3	189.3	238.9992	243.8647	226.462
163.2	104.3	154.3992	155.4878	170.883
115.1	55.3	106.6502	111.4401	107.3275
86.4	86.4	105.2934	105.1635	113.6125
284.9	285	297.0376	299.1015	277.2565
40.8	41.6	102.6746	111.695	84.519
317.9	355.8	372.2178	373.6947	369.6425
283.4	474.5	301.4712	303.0509	297.404
106.8	156.2	122.5656	122.7582	136.5835
88.9	93.2	102.2712	106.9518	96.48
231.9	246.7	235.8464	236.1624	252.4695
276	276.1	275.1614	281.6361	255.27
247.3	247.3	242.6804	245.8517	235.6455
163.2	163.2	158.3656	157.331	181.068
115.1	115.1	122.172	120.1104	144.053
86.4	86.5	104.9362	105.5456	113.093
284.9	293.2	342.5894	339.5134	354.515
40.8	96.6	113.8484	120.4908	101.9595
317.9	469.4	399.1824	396.0761	405.1555
283.4	414.7	332.9934	325.4214	353.512
106.8	206.6	147.3182	138.6545	170.56
88.9	93.4	120.3458	118.6117	108.6935
231.9	472.2	253.203	247.5011	273.5825
276	306.1	286.7	290.0377	281.806
247.3	247.3	253.127	254.2267	263.1915
163.2	163.3	167.8076	165.9061	203.39
115.1	115.1	119.2412	121.6849	133.348
86.4	86.4	88.5376	97.24042	76.5845
284.9	284.9	327.5414	325.082	336.153
40.8	298.8	154.808	163.6641	209.5435
317.9	469.2	446.1698	445.7554	439.9845
283.4	469.2	397.717	388.4221	432.2005
106.8	214.3	147.989	139.1199	172.424
88.9	473.8	127.4838	123.0045	125.093

Figure 6.2 : Comparision of model releases with actual demand

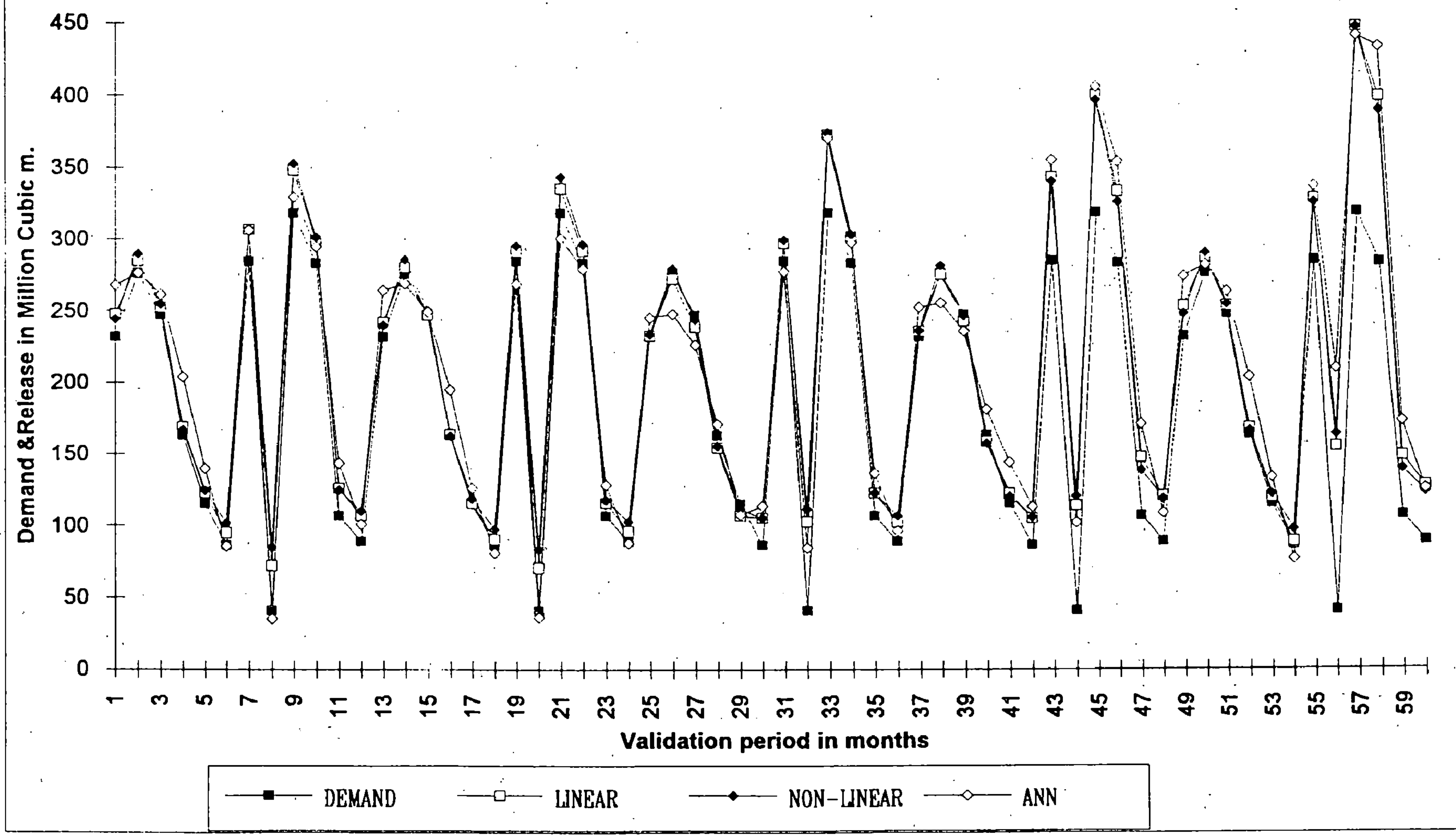


Figure 6.3 : Comparision of model releases with DP release

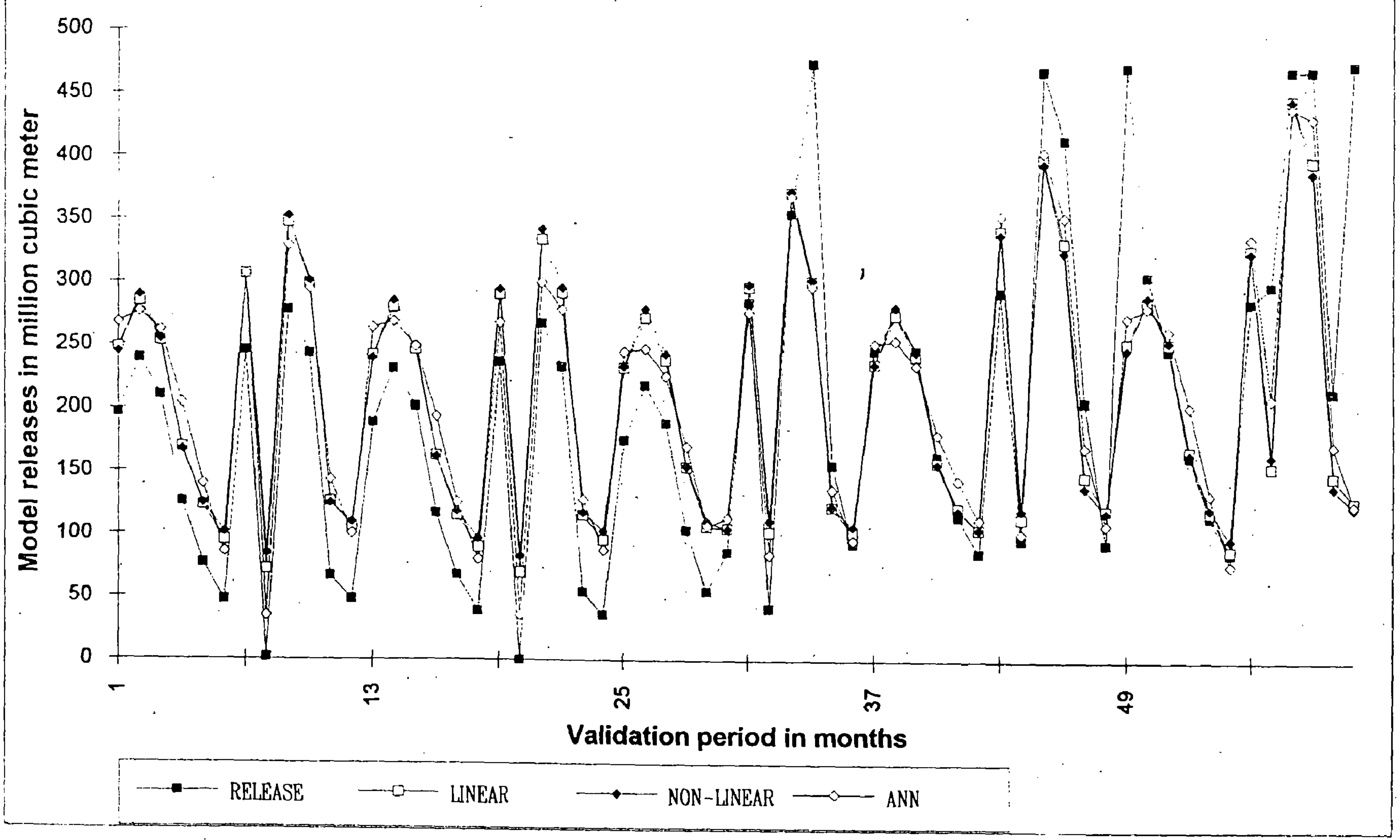


Table 6.3 : Comparision of actual demand, DP release and various model releases with reference to statistical properties and RMS error during the validation period of five years, from 1979 to 1982

Criteria	Actual Demand	DP Release	Linear regression model	Nonlinear Regression Model	ANN model
1	2	3	4	5	6
Mean	186.88	200.07	211.43	210.86	212.15
Standard Deviation	92.59	129.00	97.49	97.08	99.08
Skewness	-0.09	0.64	0.39	0.41	0.24
Kurtosis	-1.59	-0.22	-0.99	-1.05	-0.67
Correlation with Demand	1.00	0.72	0.95	0.95	0.92
Correlation with DP Release	0.72	1.00	0.81	0.80	0.83
RMS error in mapping DP Release	-	-	5715.94	6090.66	5272.49

Time Series Analysis For River Flow Prediction

7.1.0 GENERAL

In time series analysis, stochastic or time series models are fitted to one or more of the time series describing the system for such purposes, as forecasting, generating synthetic sequences for use in simulation studies and investigating and modelling the underlying characteristics of the system under study. One particular area, where time series has played a crucial role is the field of water resources.

In most of the water resources problems, after fitting stochastic models to pertinent hydrological time series, such as, sequences of river flows, the fitted models are employed for simulating possible hydrological inputs to the system. Subsequent to the construction of the system, stochastic models are employed for forecasting the input flows to the system, in order to ascertain an optimal operating policy which maximizes the project output, subject to physical, environmental, economical and system constraints.

The present study herein is aimed at obtaining a suitable river flow forecast model for at least one time period in advance. For this purpose, the Box and Jenkins ARIMA modelling approach and an artificial neural network based approach have been considered.

7.2.0 BOX AND JENKINS ARIMA MODELLING

The Box-Jenkins procedure is one of the most popular procedures for short term forecasting. The basic assumption behind this procedure is that a stationary process can often be parsimoniously represented by a mixture of auto-regressive and moving average models and a non-stationary process can be integrated to yield a stationary model, by adopting differencing technique in the appropriate way.

7.2.1 Stages in ARIMA Modelling

The practical steps involved in the Box & Jenkins analysis and the main stages in setting up the forecasting model, in this study, are described herein.

- *Model Identification* : The data are examined to see which model in the class of ARIMA processes appears to be the most appropriate.
- *Estimation* : The parameters of the chosen model are estimated by least squares approach.
- *Diagnostic Checking* : The estimated residual from the fitted model is examined to see if it is adequate.
- *Alternative Models* : In order to obtain satisfactory model, the above procedure (e.g. steps above), is repeated with alternative models.

7.2.2 The Data

Historical monthly inflow records into the reservoir for a period of 32 years, constituted the basic data. Before model identification, the entire flow record has been examined for important statistical parameters, and tests for randomness, trend and seasonality are conducted.

7.2.2.1 Test for randomness and trend

For checking the randomness of the series , turning Point Test has been adopted. The annual average data series having a data length of 32, showed randomness and the monthly inflow data series having 384 data points showed non-random behaviour. Both the time series were tested for trend through kendal's rank correlation test and linear regression test. The annual data series showed a falling trend. For details of the algorithms for these tests see *Goel [1997]*. The results can be briefed as below, in Table (7.1).

TABLE - 7.1

Turning Point Test

Particulars	No of Peaks	No of Troughs	Turning Points	Value of Z
i.Monthly data series	53	41	94	-19.49
ii.Average annual Flow Series	9	9	18	-0.86

Remark: Monthly data series: not random at 5% significance level.

Annual data series: random at 5% significance level.

Kendal's rank correlation test

Particulars	Value of P	Test Statistic
Monthly data series	35485	-1.02103
Average annual		

Flow Series 159 -2.88653

Remark: Monthly data series: There is no trend in data at 5% significance level.
Annual data series: There is falling trend in data at 5% significance level.

7.2.3 Plotting of Time Series

The inflow data has been tabulated along with the statistical parameters, such as, mean, standard deviation, skewness, kurtosis etc., in Table (5.2). The time series plot, for the calibration period of first 28 years, is given in Figure (7.1). The model identification was based only upon the auto-correlation and partial auto-correlation functions. A plot of the ACF and PACF for the original data series before adopting any standardisation or differencing schemes, is given in figure (7.3 a&b). For the computation of ACF and PACF, 20 lags have been considered.

Strong seasonality is shown by the time series plot and by the sinusoidal behaviour of the ACF plot. This seasonality along with a falling trend is indicative of the fact that instead of an additive model, a multiplicative model can represent properly the seasonal effect. The stability of the seasonal effect was examined by plotting the time series, after adopting various combinations of standardisation, differencing and transformation, and will be discussed later in this chapter. Finally a simple standardisation of the original time series has been selected, which is of the form,

$$Z_t=\frac{X_t-\overline{X_t}}{N} \quad (\overline{X_t}\ representsmean) \qquad (7.1)$$

Where

Z_t = variate of the standardised series at time t;

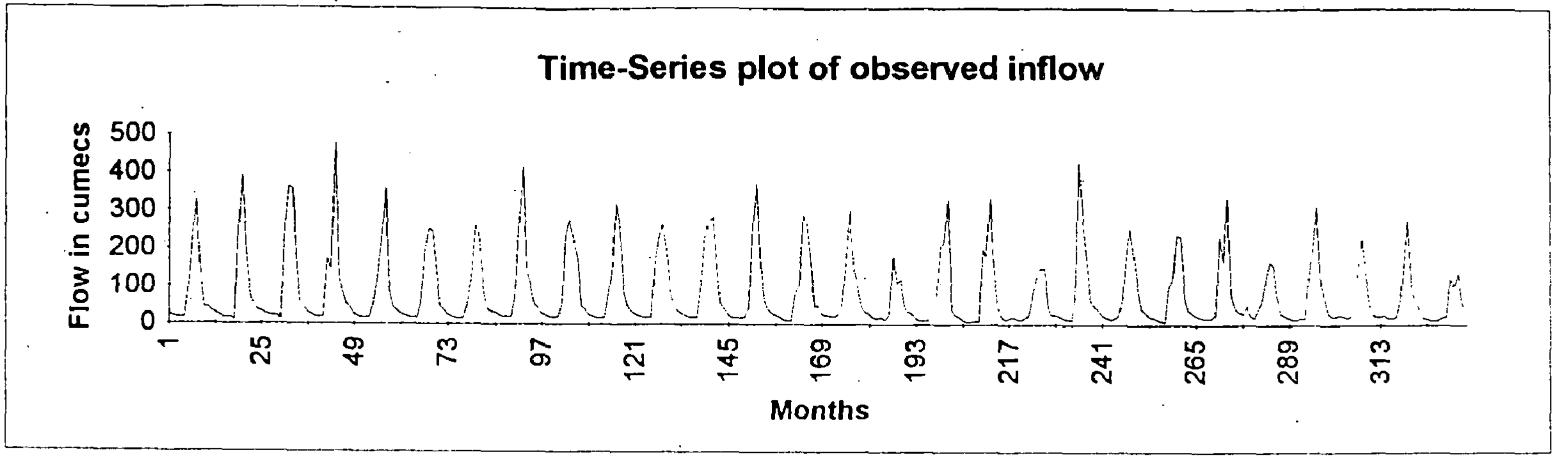


Fig 7.1 :Time series plot of the observed inflow into the reservoir during the calibration period of 28 years, total pattern length being 336.

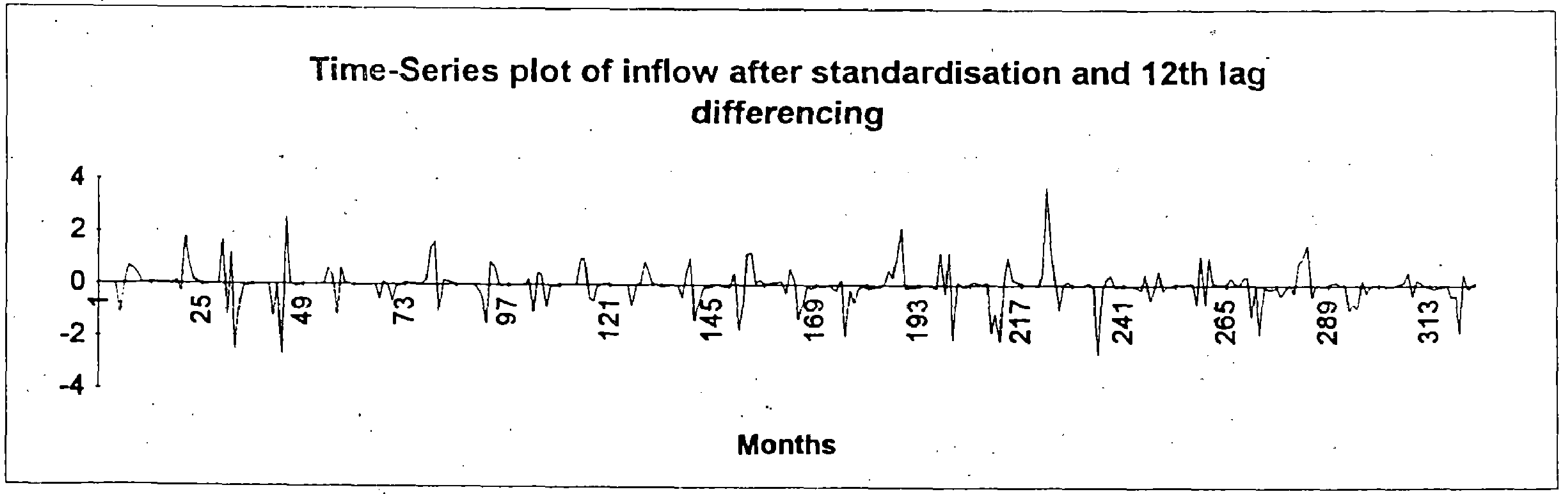


Fig 7.2 : Time series plot of the observed inflow into the reservoir during the calibration period of 28 years after standardisation and differencing. Total pattern length being 324

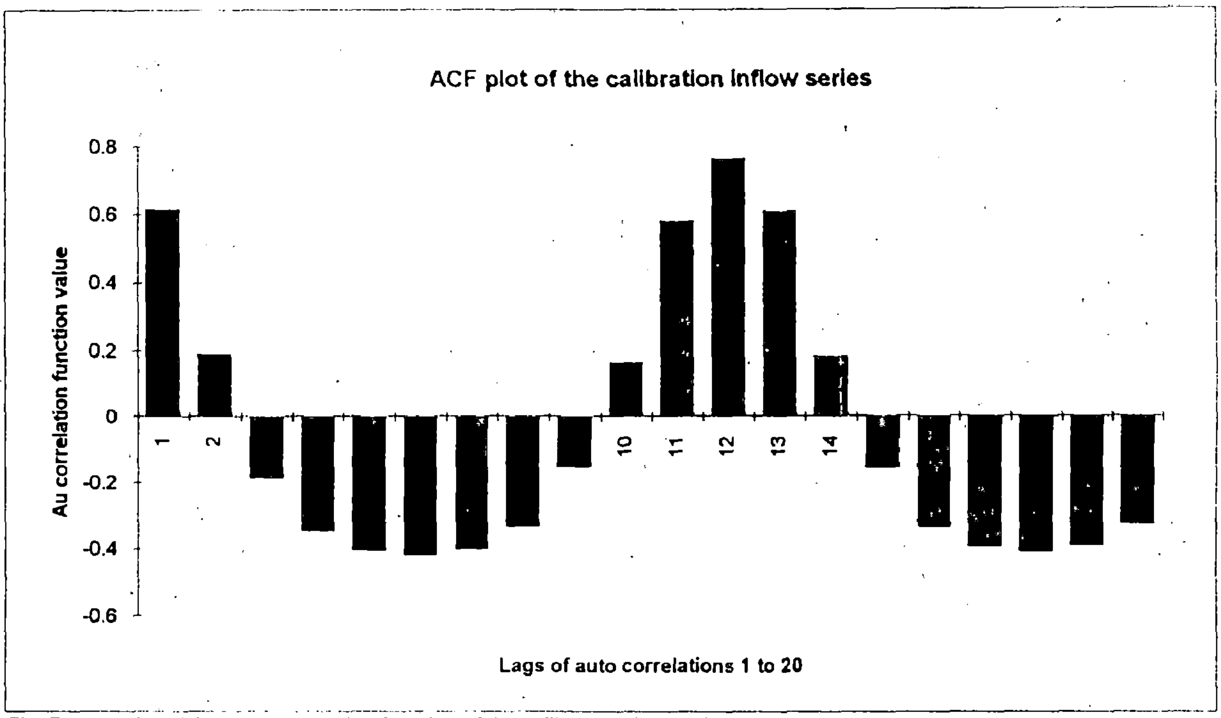


Fig -7.3 ,a : Plot of the auto-correlation function of the calibration data series before standardisation and differencing . (Period of 28 years has been kept as the calibration period)

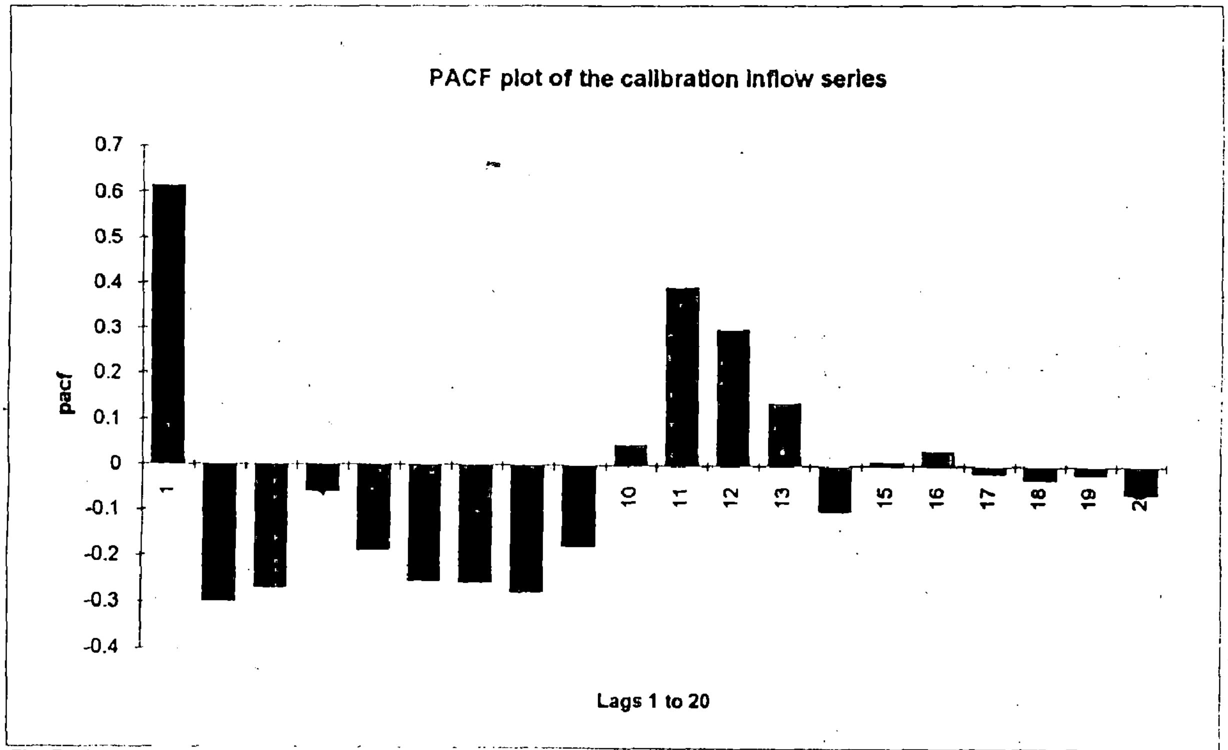


Fig -7.3,b : Plot of the partial auto-correlation function of the the calibration data series before standardisation and differencing . (Period of 28 years has been kept as the calibration period)

x_t = variate of the original time series at time t;
N = total data length.

7.2.4 Differencing to Attain Stationarity

The first stage in Box-Jenkins procedure is to *difference* the series $\{z_t\}$, until a stationary series, say $\{w_t\}$, is obtained. As the series has a falling trend and a seasonal pattern completing one cycle every 12 observations, the sample auto-correlation function of the series, was examined for various integer values of d and D [*Box-Jenkins, 1976*], where for example,

$$\nabla^{12} Z_t = Z_t - Z_{t-12} \qquad (7.2)$$

It has been found that a 12-lag differencing operator removed the annual cycle in the monthly time series but introduced periodicities in the continuous spectral density. Figure (7.2), showing the time series plot after standardisation and differencing and Figure (7.4 a&b), showing the corresponding ACF plot confirm this statement. This limitation of ARIMA family of models has earlier been pointed out by many authors *[Chatfield et al., 1973]*, *[Delleur et al., 1978]*, that whenever the original data are differenced to attain stationarity, spurious auto correlations may sometimes be introduced, particularly at lag-12 or in its neighbourhood, for monthly data.

7.2.5 ACF and PACF Computation

The ACF and PACF up to 20 lags, before and after the standardisation and differencing are shown in Table (7.4 a&b), along with their confidence limits. The mathematical expression adopted for computing the same are given below.

1. w_t series is computed by $w_t = \nabla^d \nabla_s^D Z_t$ (7.3)

Where,

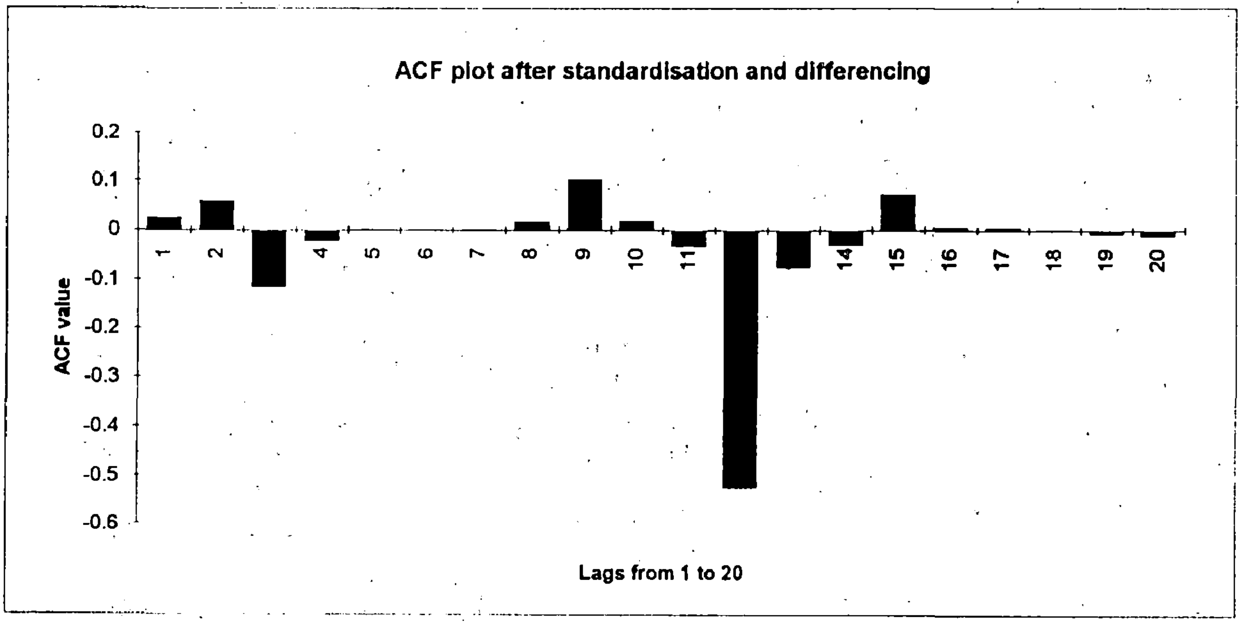


Fig -7.4,a : Plot of the auto-correlation function of the calibration data series after standardisation and differencing

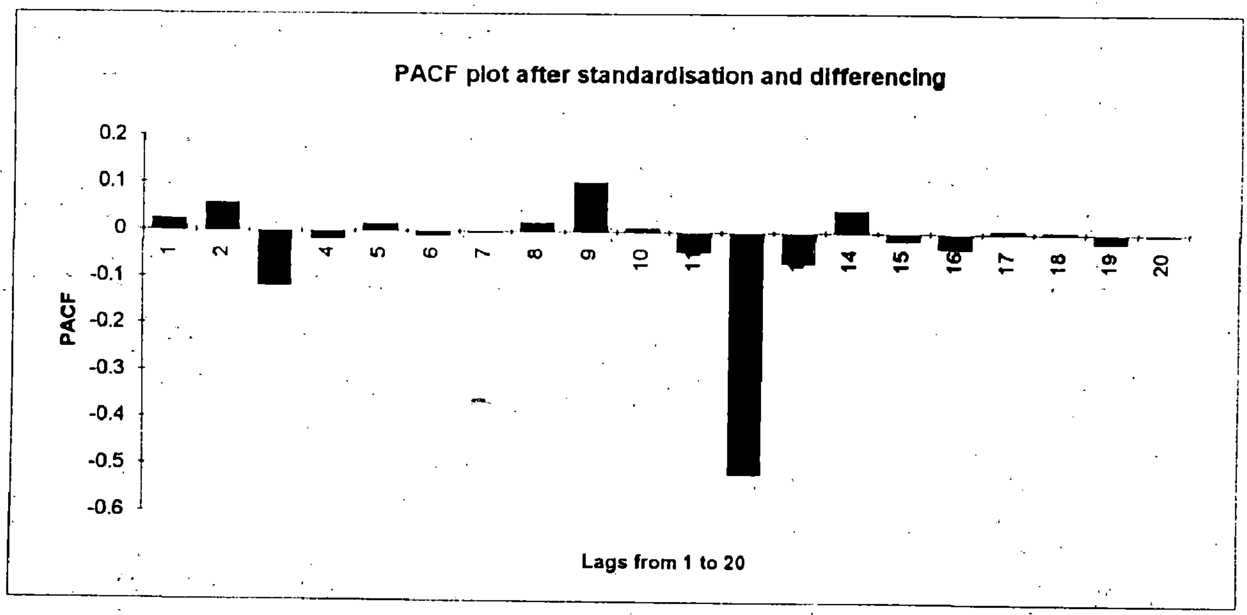


Fig - 7.4,b : Plot of the partial auto-correlation function of the calibration data series after standardisation and differencing

Table- 7.4,a : ACF and PACF values along with 95% probability limits for the original inflow data during calibration period

Lag	ACV	AC	95% LIMITS Lower	Higher	PACF	95%Lim
1	0.61	0.6149	-0.11	0.104	0.6149	0.107
2	0.19	0.1901	-0.11	0.104	-0.3022	0.107
3	-0.19	-0.1881	-0.11	0.104	-0.2733	0.107
4	-0.35	-0.3509	-0.11	0.104	-0.061	0.107
5	-0.41	-0.4087	-0.11	0.105	-0.1883	0.107
6	-0.42	-0.421	-0.11	0.105	-0.2573	0.107
7	-0.4	-0.4043	-0.11	0.105	-0.2597	0.107
8	-0.34	-0.3378	-0.11	0.105	-0.2804	0.107
9	-0.16	-0.1555	-0.11	0.105	-0.18	0.107
10	0.16	0.1643	-0.11	0.105	0.0483	0.107
11	0.58	0.5796	-0.112	0.105	0.3934	0.107
12	0.76	0.7632	-0.112	0.106	0.3	0.107
13	0.61	0.6076	-0.112	0.106	0.1397	0.107
14	0.18	0.1828	-0.112	0.106	-0.103	0.107
15	-0.16	-0.1596	-0.112	0.106	0.0095	0.107
16	-0.34	-0.3396	-0.113	0.106	0.0347	0.107
17	-0.4	-0.3971	-0.113	0.106	-0.0164	0.107
18	-0.41	-0.4113	-0.113	0.107	-0.0296	0.107
19	-0.39	-0.3949	-0.113	0.107	-0.0194	0.107
20	-0.33	-0.3302	-0.113	0.107	-0.0622	0.107
No. of data = 336, Mean =83.726, Variance =.952E+02						

Table 7.4,b : ACF and PACF values along with 95% probability limits for the inflow data after standardisation and differencing, during calibration period

Lag	ACV	AC	95% LL/ HL		PACF	95%Lim
1	0.01	0.0251	-0.112	0.106	0.0251	0.109
2	0.03	0.0596	-0.112	0.106	0.059	0.109
3	-0.05	-0.1167	-0.112	0.106	-0.1201	0.109
4	-0.01	-0.0234	-0.113	0.106	-0.0209	0.109
5	0	-0.0009	-0.113	0.106	0.0151	0.109
6	0	0.0004	-0.113	0.107	-0.0115	0.109
7	0	0.0012	-0.113	0.107	-0.0048	0.109
8	0.01	0.0165	-0.113	0.107	0.0189	0.109
9	0.05	0.1037	-0.113	0.107	0.104	0.109
10	0.01	0.018	-0.114	0.107	0.0096	0.109
11	-0.01	-0.033	-0.114	0.107	-0.0445	0.109
12	-0.23	-0.5258	-0.114	0.108	-0.5169	0.109
13	-0.03	-0.0764	-0.114	0.108	-0.0679	0.109
14	-0.01	-0.0314	-0.114	0.108	0.0454	0.109
15	0.03	0.0717	-0.115	0.108	-0.0184	0.109
16	0	0.0068	-0.115	0.108	-0.0343	0.109
17	0	0.0055	-0.115	0.108	0.0072	0.109
18	0	-0.0005	-0.115	0.109	0.0031	0.109
19	0	-0.0087	-0.115	0.109	-0.0208	0.109
20	-0.01	-0.0131	-0.116	0.109	-0.0048	0.109
No. of data = 324, Mean = -.015, Variance =.4394E+00						

t=1,2,3, ... ,n; d = 0; D =1; s = 12;

$$\nabla w_t = w_t - w_{t-1} \qquad (7.4)$$

$$\nabla_s w_t = w_t - w_{t-s} \qquad (7.5)$$

$$n = N - d - SD \qquad (7.6)$$

2. Mean and variance of the differenced series is obtained by

$$\overline{w} = \frac{1}{n}\sum_{t=1}^{n} w_t \qquad (7.7)$$

$$let \quad variance\ S_w^2 = c_0 \qquad (7.8)$$

3. Auto-covariance function :

$$c_k = \frac{1}{n}\sum_{t=1}^{n-k} (w_t - \overline{w}) \cdot (w_{t+k} - \overline{w}) \qquad (7.9)$$

Where, k = 0,1,2,... , K (ie., maximum 20 lags)

4. Auto-correlation Function :

$$r_k = \frac{c_k}{c_0} \qquad (7.10)$$

Where, k = 0,1,2, ... , K

5. Partial Auto-correlation Function :

$$\hat{\Phi}_{ll} = \frac{r_l - \sum_{j=1}^{l-1} \hat{\Phi}_{l-1,j} \cdot r_{l-j}}{1 - \sum_{j=0}^{l-1} \hat{\Phi}_{l-1,j} \cdot r_j} \qquad (7.11)$$

Where, L.H.S. of the equation represents PACF and l = 2,3,..,L

7.2.6 Identifying the Stationary Process

After differencing and examining the ACF and PACF plots, the problem remains to find a model in the ARIMA class, with as few parameters as possible, using parsimony considerations and which adequately should describe the series $\{w_t\}$. The general class of mixed autoregressive moving average seasonal model can be written as

$$\phi_p(B)\,\Phi_P(B^{12})\,w_t=\theta_q(B)\,\Theta_Q(B^{12})\,a_t \qquad (3.12)$$

Where,

B = backward shift operator

$\phi_p, \Phi_P, \theta_q, \Theta_Q$ = polynomials of order p,P,q,Q respectively.

$\{a_t\}$ = independent random variable series with mean zero and variance σ_a^2.

The first problem encountered is to select reasonable values of p,P,q,Q. This is done by mainly examining the sample auto-correlation functions, keeping parsimony, overfitting, and good diagnostics as the guidelines for selection. Regarding the identification of AR and MA parameters from ACF and PACF plot, the tips offered by many authors including Box and Jenkins, has been adhered to. The criterions adopted and some of the observations during the trials, are summarised below.

Table 7.2 Tips for model identification through ACF and PACF

Model	ACF	PACF
AR (P)	Dies out.	Cuts off after lag p
MA (q)	cuts off after lag q.	Dies out
ARMA (p,q)	Dies out after lag q-p.	Dies out after lag p-q.
if q ≥ p	ACf should decay after lag q-p	PACF should decay from beginning.
If q < p	It should drop off from beginning	PACF should decay after lag p-q

During identification it was observed that, amongst the competing models, by examining only the ACF/PACF values, the properties of the time series could not be detected. Therefore, before the final identification, all such competing models are tested for residual analysis & diagnostics. During this job, it was seen, whether the identified model succeeds in retaining the stationarity and invertibility criteria or not. A few selected trials, along with the residual analysis and other pertinent details are tabulated in Table(7.3). On the basis of minimum AIC and Q-statistic, a (0,0,2) X (0,1,1) mixed ARIMA model is selected.

7.2.7 Initial Parameter Estimation

The initial estimates of the parameters are obtained from the equations, governing their auto-covariance structure. Program-2, titled as " *Univariate Stochastic Model Preliminary Estimation (USPE)*", given by Box-Jenkins, in form of an algorithm is implemented in FORTRAN-77, to obtain the initial estimates. Moving average models present more problem, since their ρ_ks are nonlinearly related to the θ_js. During running this module, many alternatives were eliminated for not satisfying the invertibility condition.The initial estimates for non-seasonal MA parameters obtained are

MA1 = -0.0237 MA2 = -0.0598

7.2.8 Final Parameter Estimation

Box-Jenkins [*1976*], suggest that the approximate maximum likelihood estimate(MLE) for the ARIMA model parameters be obtained by employing the unconditional sum of squares method. The modified sum of squares function is minimised, through a recursive procedure, in order to obtain the improved parameter estimates. A FORTRAN code employing the program-3, titled as "*Univariate Stochastic Model Identification*" and "The Marquardt Algorithm for Nonlinear Least Squares", given by Box-Jenkins, is implemented to obtain the least squares estimates of the

Table -7.3 : Diagnostic checking of alternative models in ARIMA family during identification

Sl. No.	ARIMA Modelling AR & MA orders				12th lag diff.	standa--rdise	Trans-formed	Residual Statistics		DOF	Q statistic	AIC
	Non-Seasonal		seasonal					mean	variance			
	AR	MA	AR	MA								
1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	1	0	1	no	yes	yes	-0.4386	113.59	19	43.292	-275.12
2	0	1	1	1	no	yes	yes	Error : Marquardt algorithm				
3	0	2	0	1	no	yes	yes	-0.4123	103.59	18	37.598	-281.076
4	1	1	0	0	yes	yes	yes			18	96.454	-217.161
5	1	1	0	0	yes	yes	no			17	83.626	-313.923
6	1	2	0	0	yes	yes	no			16	76.973	-322.982
7	0	2	0	2	no	yes	yes	-0.2556	123.5947	18	37.362	-281.082
8	0	3	0	3	no	yes	yes	Error : invertibility				
9	**0**	**2**	**0**	**1**	**yes**	**yes**	**no**	**-0.0001**	**66.17771**	**17**	**11.89**	**-481.139**
10	2	1	1	1	yes	yes	no	Error : invertibility				
11	2	1	0	1	yes	yes	no	Error : invertibility				
12	1	1	0	1	yes	yes	no	-0.0214	65.6366	17	14.916	-480.704
13	2	0	1	0	yes	yes	no	Error : Marquardt algorithm				
14	2	0	0	1	yes	yes	no	0.0014	66.2511		17	-480.374
15	0	2	0	2	yes	yes	no	-0.0001	66.1769	17	11.89	-481.121

parameters as,

Seasonal MA parameter = -0.951
MA parameter (1) = -0.123, Confidence limit = 0.0551

MA parameter (2) = -0.125, Confidence limit = 0.0552

7.2.9 Diagnostic Checks

In the context of previous discussions, the diagnostic tests deal with the residual assumptions, in order to determine, whether the a_ts are independent, homoscedastic and normally distributed. However, these estimates, along with the residual auto-correlation function, Porte-Manteau test statistic and AIC, are calculated , during the implementation of the program for final parameter estimation. The RACF values, mean and variance of the a_t series and Q-value are shown in Table(7.4 c).

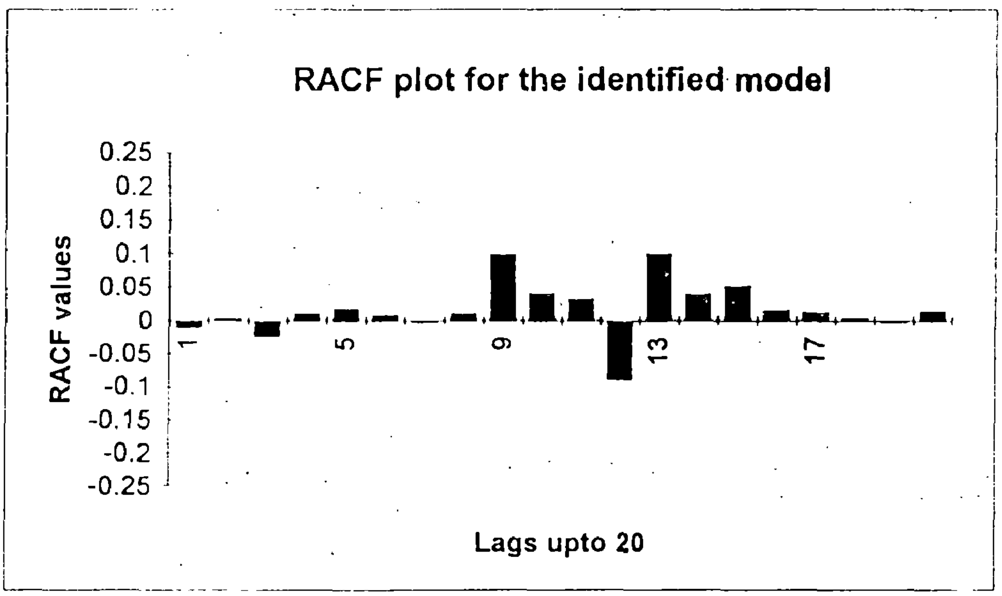
It was found during diagnostic testing that, a mere data transformation couldn't correct dependence of residuals. So, in order to attain independence, many times the identification and estimation stages have been repeated for determining a suitable model. Selected trials are summarized in Table (7.3).

During diagnostic checks, model adequacy is usually tested by overfitting also. This involves fitting a more elaborate model, than the one estimated, to see if including one or more parameters, greatly improves the fit. The same Table (7.3) on diagnostic checks may be referred, to observe the changes in Q-statistic, AIC or in the residual variance, when the order of the AR or MA parameters is increased, one at a time. As it can be seen from that table that, any further increase over and above those selected, i.e., (0,0,2)X(0,1,1) model, doesn't significantly improve the model characteristic, it is considered to be final. The RACF values for the identified model are furnished in Table(7.4c) and the corresponding plot is given in Figure (7.5).

Table 7.4,c : Auto-correlation function of the residuals,
The Q_statistic and the AIC value for the TS model

Lag	ACV	AC	RESIDUAL ANALYSIS
1	-0.003	-0.012	
2	0.001	0.002	
3	-0.005	-0.025	
4	0.002	0.01	Statistics of Residuals
5	0.004	0.018	Mean : -.0001, Var : 66.1771
6	0.002	0.008	
7	-0.001	-0.004	
8	0.002	0.01	
9	0.021	0.098	Portmanteau Test Result
10	0.009	0.041	Q-Statistic = 11.890,
11	0.007	0.032	Degrees of Freedom = 17
12	-0.02	-0.091	
13	0.022	0.099	
14	0.009	0.04	
15	0.011	0.052	AIC = -481.139
16	0.003	0.015	
17	0.003	0.012	
18	0.001	0.004	
19	-0.001	-0.004	
20	0.003	0.014	

Fig - 7.5 : The Racf plot for the identified ARIMA (002 X 011) model



7.2.10 Summing The Model Selection

The above computations, starting from obtaining data statistics up to RACF analysis, were carried out by computer programs, developed in FORTRAN-77. For better identification, provision was kept in the programs for instantaneous display of the time series plots, in original and after user suggested transformations, standardisation and differencing. For a detailed and step by step algorithm of the computations described above, readers are requested to refer program-1,2,3 of *Box and Jenkins [1976]*.

7.3.0 FORECASTING

After selection of the multiplicative, seasonal ARIMA model and computing the model parameters, the task remaining is to use the model to forecast future values of the observed inflow time series. It may be borne in mind before forecasting that estimation errors in the parameters will not seriously affect the forecasts unless the number of data points, used to fit the model, is small. Before forecasting, the model is expressed mathematically as per Box-Jenkins notations.

7.3.1 The Identified Model

One of the most general form of multiplicative seasonal ARIMA (p,d,q) X $(P,D,Q)_w$ model is written as

$$\begin{aligned}&(1-\Phi_1 B^{w}-\Phi_2 B^{2w}-\ldots\ldots-\Phi_P B^{Pw})\\&(1-\phi_1 B-\phi_2 B^2-\ldots\ldots-\phi_p B^p)\\&(1-B^{w})^{D}.(1-B)^{d} Z_t \quad = \\&(1-\Theta_1 B^{w}-\Theta_2 B^{2w}-\ldots\ldots-\Theta_Q B^{Qw})\\&(1-\theta_1 B-\theta_2 B^2-\ldots\ldots-\theta_q B^{q})\, a_t\end{aligned} \tag{7.13}$$

Which in a short form can be written as

$$\Phi(B^w).\phi(B).(1-B^w)^D(1-B)^d.x_t=\Theta(B^w).\theta(B).a_t \qquad (7.14)$$

In the present case

p = P = 0; q = 2; Q = 1; D = 1; d = 0;

So along similar lines, the identified model can be expressed as

$$\begin{aligned} &(1-B^{12})Z_t=(1-\theta_1 B-\theta_2 B^2)(1-\Theta_1 B^{12})a_t \\ \Rightarrow\quad &Z_t-Z_{t-12}=(1-\theta_1 B-\theta_2 B^2)(a_t-\Theta_1 a_{t-12}) \qquad (7.15) \\ =\quad &a_t-\theta_1 a_{t-1}-\theta_2 a_{t-2}-\Theta_1 a_{t-12}+\theta_1\Theta_1 a_{t-13}+\theta_2\Theta_1 a_{t-14} \end{aligned}$$

or the final expression can be written as,

$$Z_t=Z_{t-12}+a_t-\theta_1 a_{t-1}-\theta_2 a_{t-2}-\Theta_1 a_{t-12}+\Theta_1(\theta_1 a_{t-13}+\theta_2 a_{t-14}) \qquad (7.16)$$

Where Z_t is the standardised series.

Equation (7.16) represents the elaborate form, which has been used recursively in the forecast algorithm, provided by Box-Jenkins in program-4, titled "Univariate Stochastic Model Forecasting" with scope for some generalisation.

7.3.2 Input Parameters for Forecast

The following information was provided to the program in the form of an input file: the $\{x_t\}$ series during the calibration period, total data length, order of differencing operators d & D, seasonality S, order and least squares estimates of corresponding nonseasonal and seasonal AR and MA parameters, maximum lead time of forecast, maximum value of backward origin, and least squares estimate of the residual variance. Apart from these, the other information, such as, data manipulation options like standardisation, differencing orders and lags, transformation etc., and the options relating to updating the forecast are accepted from the user, during the program execution.

7.3.3 Difference Equation Approach

Box and Jenkins approach to forecasting provides three different ways to view at an eventual forecast function and its updating. These are

- Forecasts from difference equation approach;
- Forecasts in integrated form;
- Forecasts as a weighted average of previous observations and forecasts made at previous lead times from the same origin.

In the present study, for generating the forecasts at various lead times, the difference equation approach is used. However, for computing the probability limits of forecast and for updating the higher lead time forecasts. the psi weight is considered. This is a linear filter, that is supposed to transform a white noise into the Z_t series. An infinite series of a_ts can be represented as,

$$\phi(B) Z_t = a_t \qquad (7.17)$$

This is equivalent to

$$Z_t = \psi(B) . a_t \qquad (7.18)$$

$$\text{with} \qquad \psi(B) = \phi^{-1}(B) \qquad (7.19)$$

Under the difference equation approach, equation (7.16) is rewritten for forecast in the following form:

$$\hat{Z}_t(l) = [Z_{t+l}] = Z_{t+l-12} + a_{t+l} - \theta_1 a_{t+l-1} - \theta_2 a_{t+l-2} - \Theta_1 a_{t+l-12} + \Theta_1(\theta_1 a_{t+l-13} + \theta_2 a_{t+l-14}) \qquad 7.20$$

Where, $[Z_{t+l}]$ is the conditional expectation of Z_{t+l}, taken at origin t.

In the program, the maximum lead value of 14 has been kept. The white noise a_t is computed by using the concept of back-casting, proposed by Box-Jenkins. This concept is useful in estimating values of the series, which have occurred before the first observation was made.

7.3.4 Updating and Forecasting Error Variance

In order to determine the updating formulae and to obtain the variance of the forecast error $a_t(1)$, the 'psi weights' are computed for a finite length. The recursive equation used for updating the forecasts, is

$$\hat{z}_{t+1}(l) = \hat{Z}_t(l+1) + \psi_l . a_{t+1} \qquad (7.21)$$

where a_{t+1} is computed after providing the actual value of Z_{t+1}. This provision was kept in the program in an user interface mode. The forecast error variance at any lead time l is calculated by the formula

$$V(l) = [1+\psi_1^2+.....+\psi_{l-1}^2]\sigma a^2 \qquad (7.22)$$

7.3.5 Discussion of Results

The last four year data, kept aside for validation, was used for testing the forecast results. Although a maximum lead time up to 14 has been kept in the forecasting program, only lead-1 forecast values are filtered for 48 months flow data. This is done by successively calling the updating provision of the program. As the purpose of the study was to compare the relative prediction performance of an ANN based forecast model with that of the Box and Jenkins multiplicative seasonal model, the discussions on the results is shifted to subsequent sections in this chapter.

7.4.0 FORECAST THROUGH ANN MODELLING

The same error back propagation algorithm and the BP simulator used earlier for mapping optimal releases, is used in this case also. Hence all the discussions on ANN, training strategy, shuffling technique etc. hold good here. However the data manipulator program and the comparison models do not hold good in this case. Therefore, only the special features and the problems encountered during ANN forecasting of inflow time series are discussed below.

7.4.1 Application of Neural Network Model

The neural network approach was executed with the historic inflow series of Indravati river for 32 years. For training purpose, the data set is divided into two parts, i.e., one part consisting of first 28 years is used for training the network, and last 4 years data was kept aside for validation. The mean square error over the training samples was used as the objective function. The MSE is given mathematically for all input patterns as,

$$E = \frac{1}{2N} \sum_{p=1}^{N} \sum_{n=1}^{m} (T_{pn} - O_{pn})^2 \qquad (7.23)$$

Where,

T_{pn} = target value T_n for the pth pattern;
O_{pn} = neural network output value O_n for the p^{th} pattern;
N = total number of patterns;
m = Total number of output neurons;

7.4.2 Preparation of Input and Output Data Patterns

The basic data which was to be appropriately manipulated into a number of input and output patterns, was the continuous flow record in the form of an one dimensional data matrix. In case of an univariate time series, searching for the appropriate lags and their algebraic manipulations posed a typical problem. In addition to above, the sigmoidal function limitation which requires the patterns to be in the range of 0 and 1, opened many options. Again, Because the identified Box-Jenkins model is of purely MA nature, a 12-lag differencing option became a possible alternative to be included into the input pattern.

So, a customised data manipulation program was written [data_manip1.c] in C language with the following provisions.

- Accept an one dimensional data matrix of any size.
- Obtain the options for standardisation and differencing. Under differencing option interactively obtain the lag order.
- Accept the data length in each pattern and their corresponding lags.
- Show the maximum and minimum value of the entire sample.
- Scale down the patterns obtained, between 0 and 1.

Basically this program maps an one dimensional vector space into a multi dimensional one and gives the maximum and minimum values. The source code for the same is given in Annexure- IV.

7.4.3 Deciding On Size Of Input and Hidden Layers

Before deciding about the size of the hidden layer, it is necessary to decide on which values should constitute the input pattern. A number of trials had to be made to decide this. As an initial guess, three consecutive inflow values and two consecutive seasonal values have been tried. Gradually number of parameters are reduced on parsimony considerations.

The final input pattern consisted of an output pattern $X_{v,\tau}$ and the input pattern consisted of $\{ X_{v,\tau-1}, X_{v,\tau-2}, X_{v-1,\tau} \}$, where τ= number of months, 1 to 12 and v = number of years of testing.If τ-1 < 1 then $X_{v,\tau-1}$ was replaced by $X_{v-1,12+\tau-1}$ and if (τ-2) < 1 then $X_{v,\tau-2}$ was replaced by $X_{v-1,12+\tau-2}$. This indicated that every piece of data was dependent on two previous data values and upon the same season of previous year.

The number of intermediate units was obtained through a trial and error procedure, adopting a similar strategy as discussed in Chapter-6. Out of many trial nets where intermediate layer neurons varied between 1 to 10, it was observed that the 3-8-1 network performed well on the minimum MSE criterion during training, testing and during comparison with other models.

7.4.4 Training

During initial phases of training, it was observed that adopting a momentum term blows up the initial weight matrix and brings the simulation process to a halt. However adding a noise term up to 0.1, helped in speedier convergence. Under the phased training programme, for successive improvement of the system error, the noise was reduced and gradually momentum was introduced incrementally. The entire process of training consisted of 14 phases and total 4000 cycles. A learning rate below 0.4 was found to extremely slow down the process. Therefore in all the training process learning rate was made to vary between 0.4 to 0.8. The finally chosen architecture is given in Figure (7.6).

7.4.5 Testing and Comparison of Results

When training was considered to be finalised, the weights were collected from the training module of the BP simulator to test the network and monitor its performance on test samples in terms of MSE criterion. In addition to the MSE criterion, two other criterions, namely, average percentage error and average monthly deviation are kept for monitoring the performance of the forecast models. For computing these values on individual years of testing and on the entire validation period, another time series simulator program is written in FORTRAN-77 and included into a batching program, similar to that described in Chapter-6.

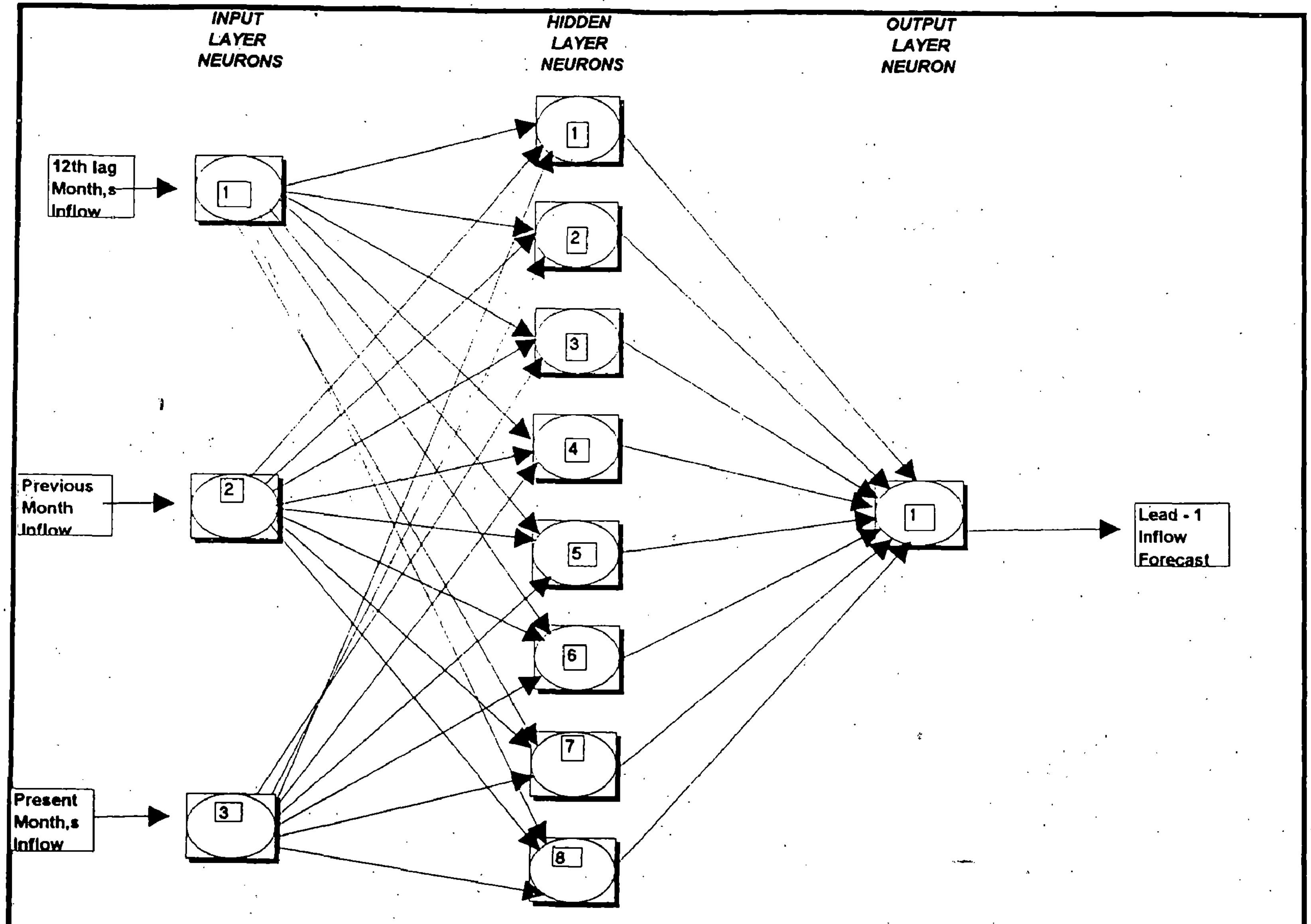


Fig. 7.6 : A 3 - 8 - 1 Neural Network Architecture, for the ANN based forecast model

Criterions for performance monitoring :
Average monthly deviation was computed by following expression.

$$\frac{1}{12}\sum_{i=1}^{12}(Actual_{Inflow}(i)-Model_{Inflow}(i)) \qquad (7.24)$$

Average percentage deviation and the RMS Error were computed by the following expressions.

$$\frac{1}{12}\sum_{i=1}^{12}\frac{(Actual_{inflow}-Model_{Inflow})}{Actual_{Inflow}}*100 \qquad (7.25)$$

$$\frac{1}{12}\sum_{i=1}^{12}(Actual_{inflow}-Model_{Inflow})^2 \qquad (7.26)$$

The RMS Error has a bias for higher deviations while the average percentage deviation has a bias towards low observed flows. Quite contrast to above two, the average monthly deviation parameter as expressed in equation (7.24) is an unbiased interpreter of the forecast performance. Therefore above three criterions have been considered to suffice the performance monitoring.

Finally an one-month-ahead forecast of inflow time series was generated from 1979 to 1982 and plotted in Figure (7.7). The monthly forecasts by ANN and Box-Jenkins model for individual years of testing along with the criterions defined

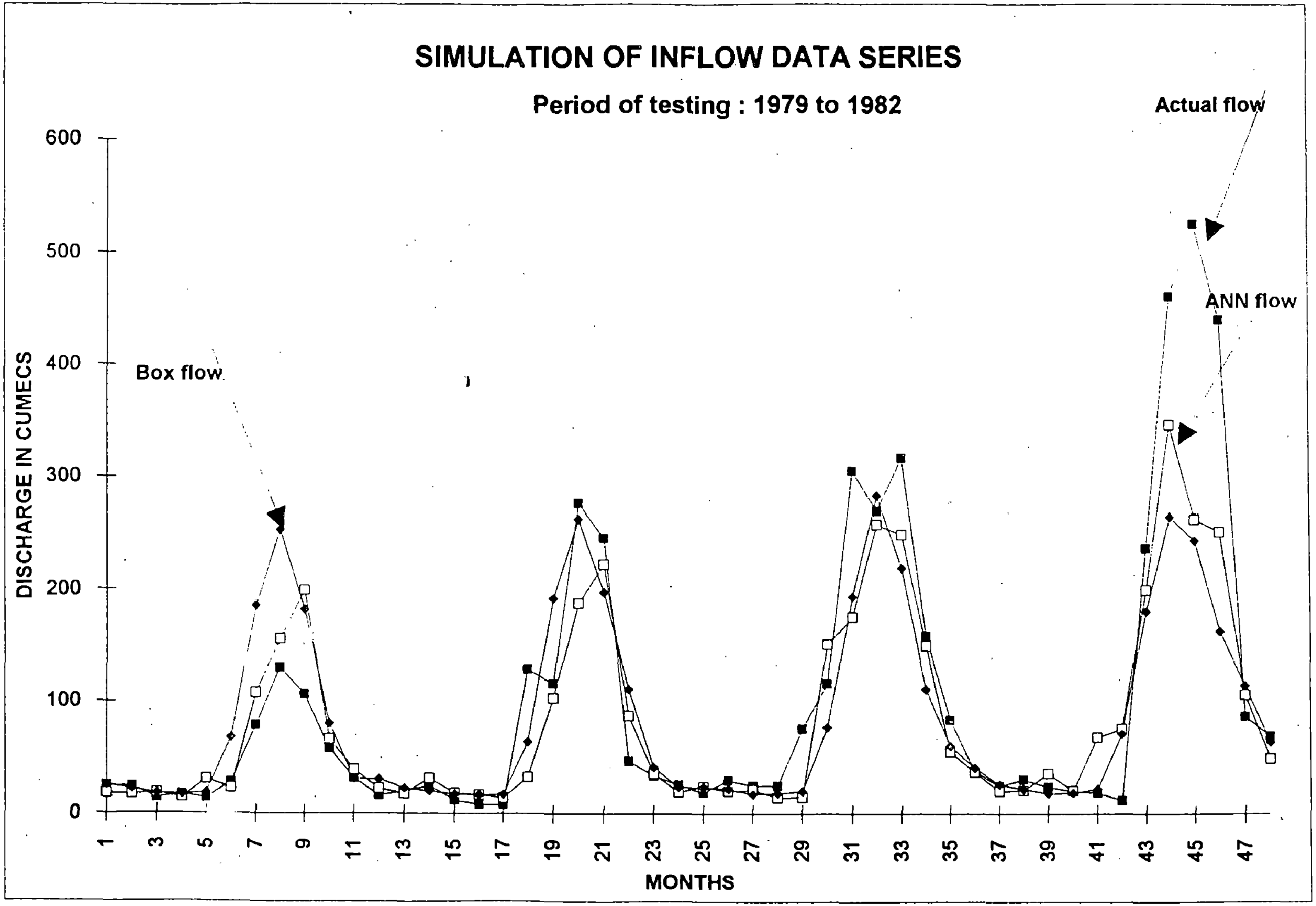


Fig. 7.7 : showing the comparision of prediction by ANN and Box model during the validation period

YEAR OF TESTING : 1979

Actual Flow	Flow by ANN	% Vol Deviation	Flow by Box model	% Vol Deviation
25	17.92	28.34	25.4	1.6
24	17.01	29.11	21.86	8.92
14	18.21	30.06	17.85	27.5
17	14.44	15.07	17.01	0.06
14	30.65	118.92	18.96	35.43
28	23.09	17.55	67.91	142.54
79	107.73	36.37	185.45	134.75
130	155.68	19.76	253.03	94.64
106	199.2	87.92	181.75	71.46
58	66.7	15	80.4	38.62
32	39.36	23	31.46	1.69
16	22.26	39.14	30.65	91.56

CRITERIA	ANN	BOX
RMS ERROR	897.01	2879.97
AVERAGE PERCENTAGE ERROR	38.35	54.06
AVERAGE MONTHLY DEVIATION	17.69	32.84

YEAR OF TESTING : 1980

Actual Flow	Flow by ANN	% Vol Deviation	Flow by Box mode	% Vol Deviation
20	18	9.99	22.71	13.55
25	31.57	26.3	20.18	19.28
12	18.21	51.73	17.71	47.58
8	17	112.55	17.1	113.75
8	14.34	79.23	17.34	116.75
129	32.63	74.7	64.25	50.19
116	103.28	10.97	192.28	65.76
277	187.69	32.24	262.92	5.08
246	222.48	9.56	197.28	19.8
47	87.43	86.02	111.11	136.4
34	35.76	5.17	41.39	21.74
26	19.11	26.5	23.67	8.96

CRITERIA	ANN	BOX
RMS ERROR	1655.83	1415 55
AVERAGE PERCENTAGE ERROR	43.75	51.57
AVERAGE MONTHLY DEVIATION	25.09	25.78

YEAR OF TESTING : 1981

Actual Flow	Flow by ANN	% Vol Deviation	Flow by Box model	% Vol Deviation
19	24.1	26.86	23.59	24.15
30	20.31	32.29	22.2	26
25	21.43	14.29	17.58	29.68
25	14.32	42.73	18.64	25.44
76	14.8	80.53	20.33	73.25
116	150.97	30.15	77.24	33.41
306	175.48	42.65	193.81	36.66
270	257.87	4.49	283.5	5
317	249.3	21.36	219.45	30.77
159	149.63	5.9	110.9	30.25
84	55.51	33.92	60 84	27.57
38	37.1	2.37	41.13	8.24

CRITERIA	ANN	BOX
RMS ERROR	2323 36	2493 63
AVERAGE PERCENTAGE ERROR	28 13	29.2
AVERAGE MONTHLY DEVIATION	31 19	34.85

YEAR OF TESTING : 1982

Actual Flow	Flow by ANN	% Vol Deviation	Flow by Box mode	% Vol Deviation
25	20.85	16.59	26.48	5.92
31	21.08	32	22.27	28.16
24	36.16	50.68	18.46	23.08
20	20.88	4.41	18.93	5.35
19	68.17	258.77	22.26	17.16
12	75.84	531.97	71.61	496.75
237	199.62	15.77	180.42	23.87
461	346.26	24.89	264.59	42.61
525	262.51	50	243.7	53.58
440	251.53	42.84	162.6	63.05
87	106.07	21.92	114	31.03
69	49.31	28.54	63.9	7.39

CRITERIA	ANN	BOX
RMS ERROR	10541.2	16857 3
AVERAGE PERCENTAGE ERROR	89.87	66 5
AVERAGE MONTHLY DEVIATION	65.16	76.96

Table -7.5: showing the comparision of ANN model performance and the Box & Jenkins model performance during the validation period of four years along with RMS error, Average percentage errors and Average monthly deviations

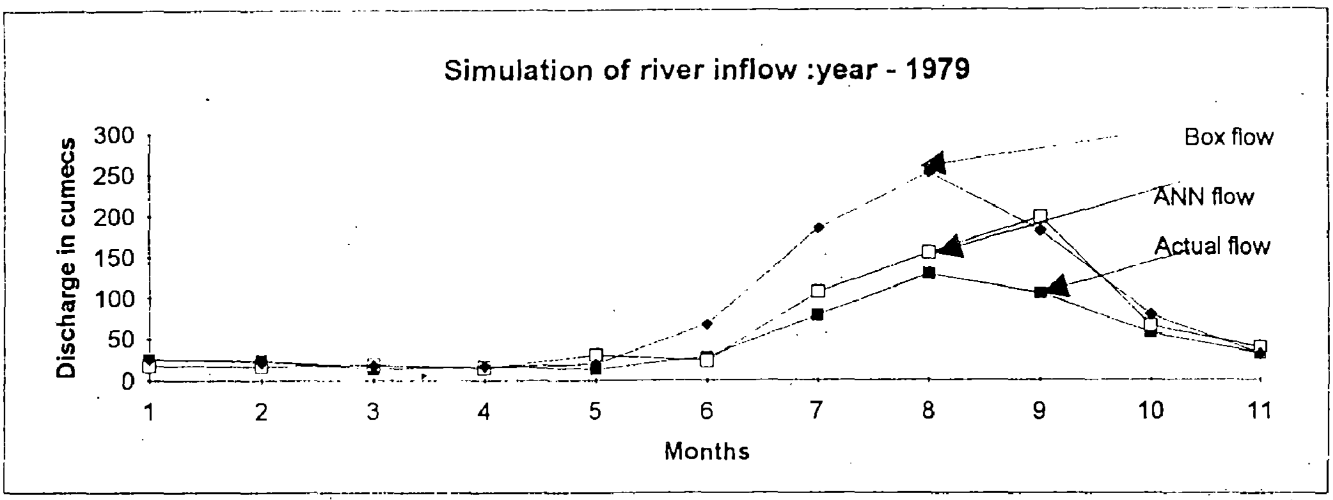


Fig - 7.8 : Figure showing the prediction of actual river inflow by the ANN model and the Box-jenkins model for the year 1979

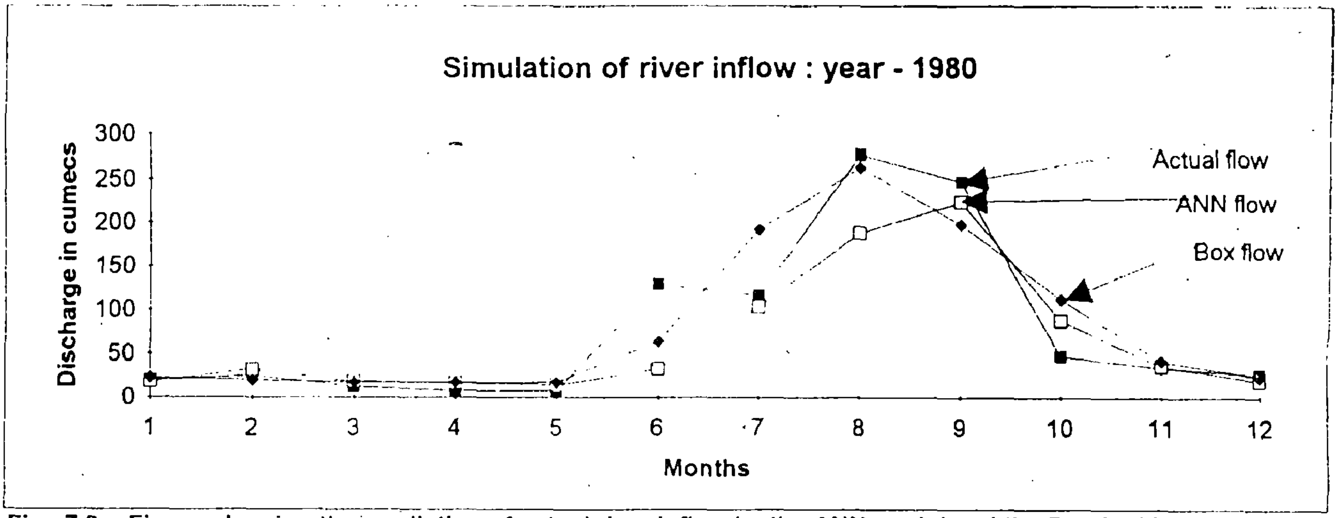


Fig - 7.9 : Figure showing the prediction of actual river inflow by the ANN model and the Box-jenkins model for the year 1980

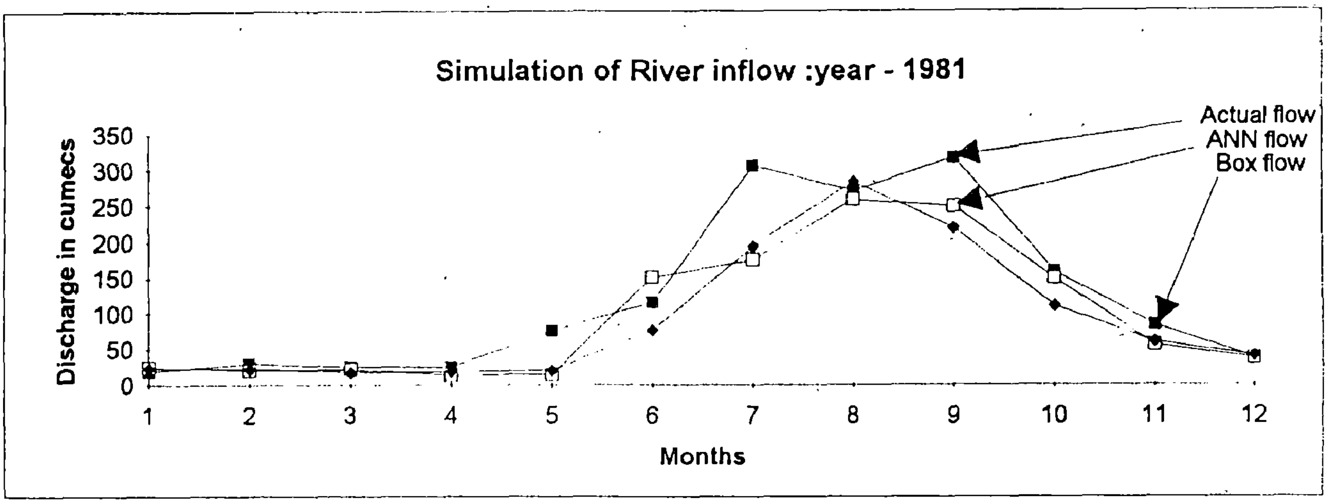


Fig -7.10 : Figure showing the prediction of actual river inflow by the ANN model and the Box-jenkins model for the year 1981

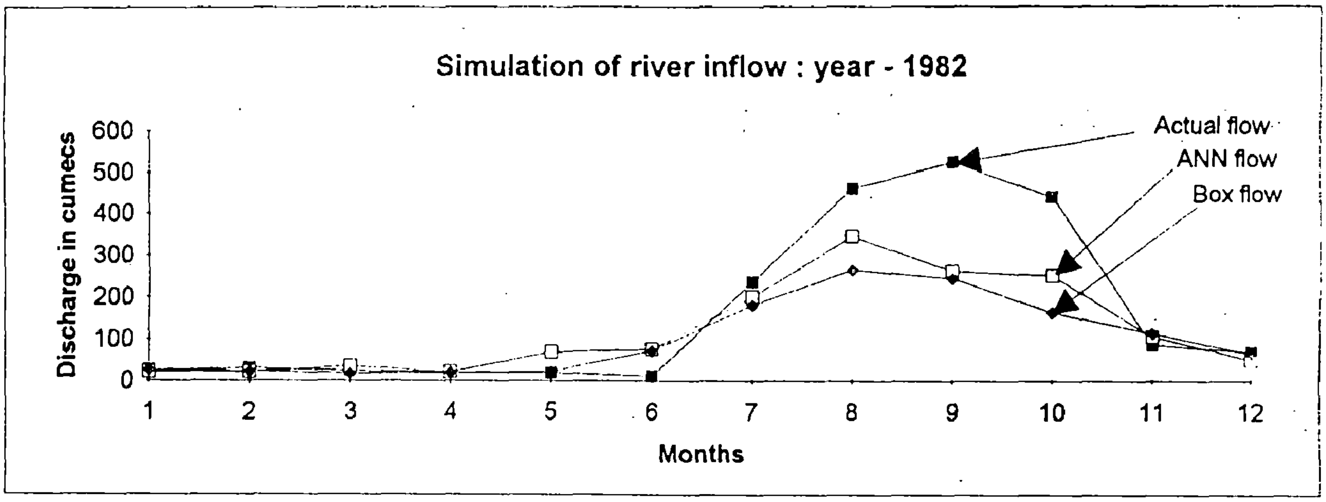


Fig - 7.11 : Figure showing the prediction of actual river inflow by the ANN model and the Box-jenkins model for the year 1982

above are furnished in Table(7.5). The corresponding river inflow predictions are given in Figure (7.8) through Figure (7.11). The bar graph for RMS error, average monthly deviation and average percentage error for individual years of testing and for entire validation period, have been plotted in Figure (7.12), from where a comparison can be made between the ANN and Box-Jenkins forecast models.

7.5.0 DISCUSSION

A summary table is prepared from the various tables and figures to have a quick grasp over the prediction ability of the competing forecast models. The same is furnished in Table (7.6). The statistical parameters for various model outputs are computed in order to see, whether the predictions have been able to retain the parent distribution and statistical properties or not. The same is furnished in Table (7.7). From observing the graphs and the summary Tables (7.6) and (7.7) the following inferences are drawn:

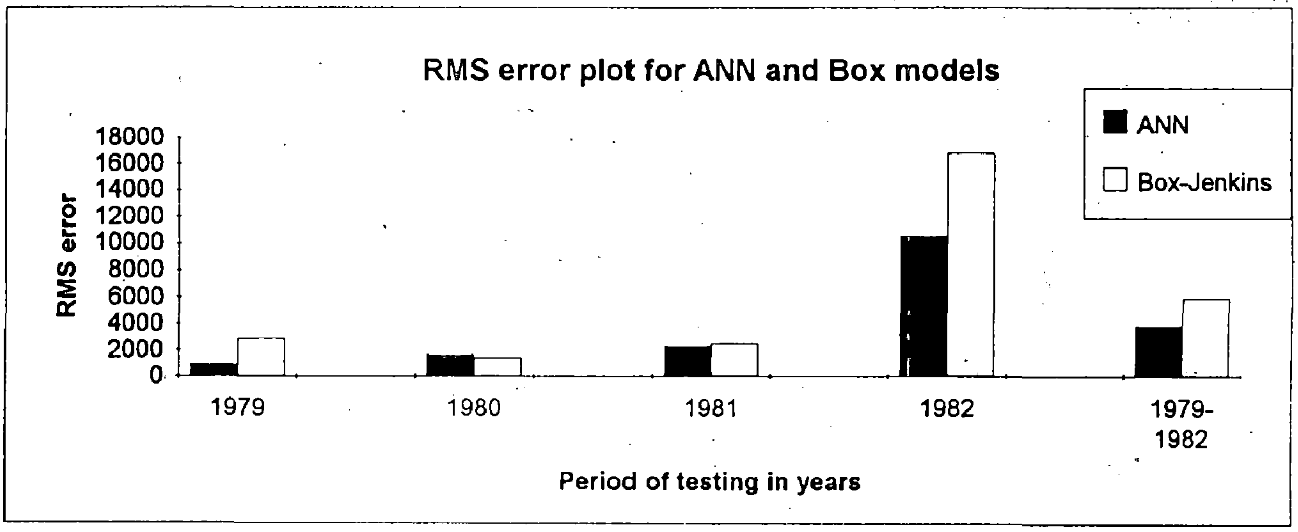
1. The ANN prediction is better correlated with the observed inflow for years 1980, 1982 and during entire validation period, whereas Box and Jenkins model shows better performance for years 1979 and 1981.

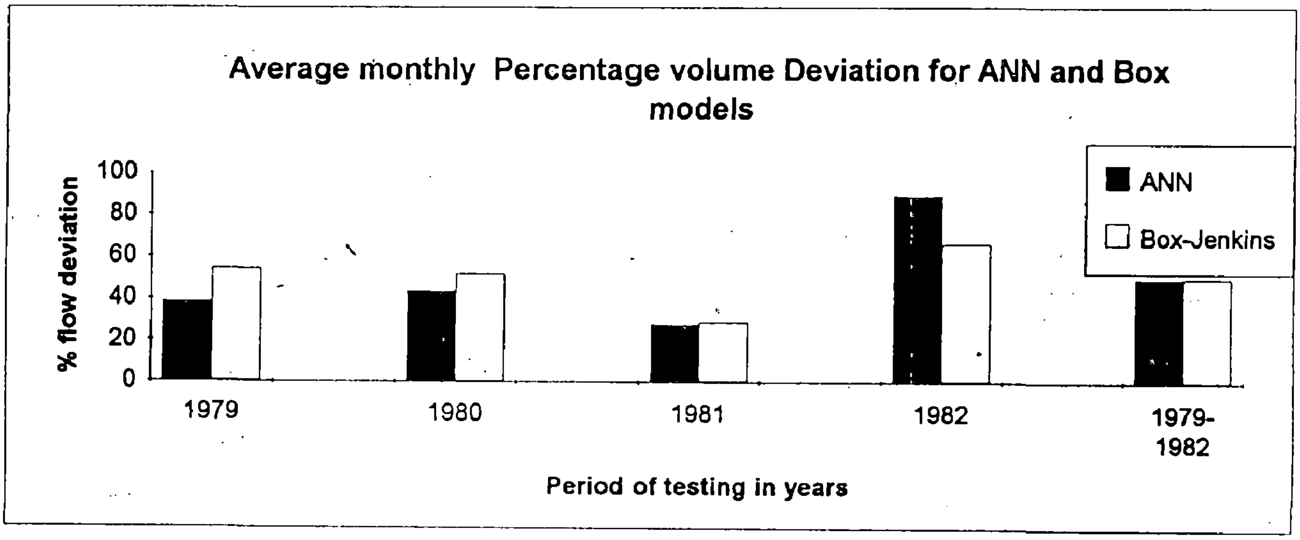
2. Average monthly deviation for all the individual years as well as for the entire period is less for the ANN model.

3. For the year 1982, during which the monsoon flows are relatively higher ANN prediction is much better than the ARIMA model.

4. Invariably, very high flows are mapped better through the ANN forecast model, whereas, extremely low values are predicted better through the ARIMA model.

Fig -7.12 : Comparision of ANN and Box & Jenkins models taking RMS error, Annual average of monthly percentage deviation and Annual average of monthly deviations





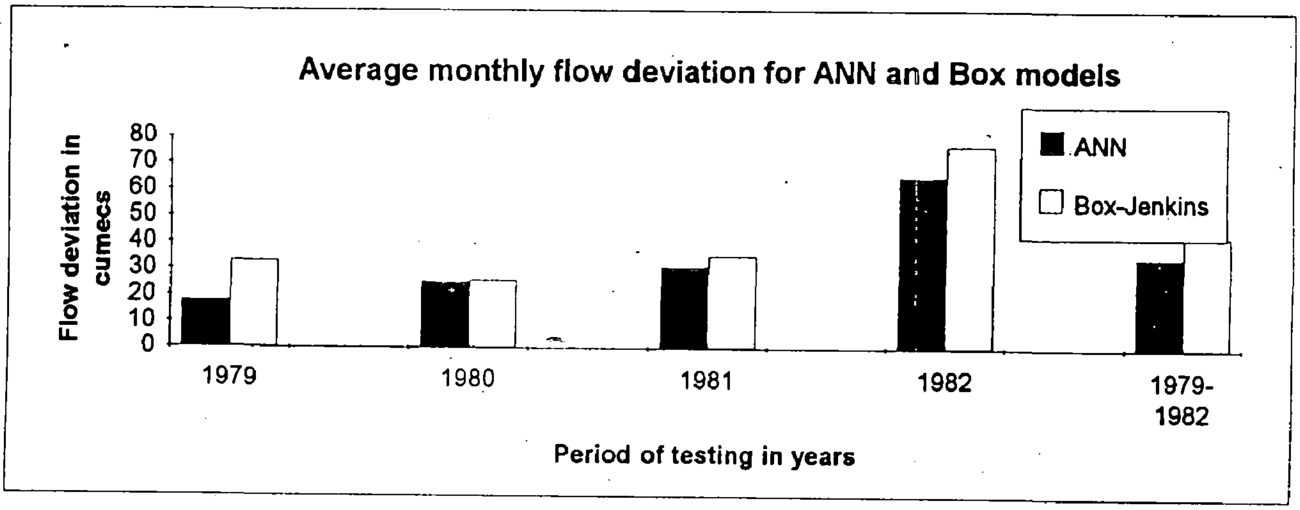


Table -7.6 : Showing the comparision of ANN forecast and Box-Jenkins forecast as per the criterias used in the time- series simulator

Period of	RMS ERROR		Monthly % vol. Deviation		Avg. Monthly Deviation	
Testing	ANN	Box-Jenkins	ANN	Box-Jenkins	ANN	Box-Jenkins
1979	897.01	2879.97	38.35	54.06	17.69	32.84
1980	1655.83	1415.55	43.75	51.57	25.09	25.78
1981	2323.36	2493.63	28.13	29.2	31.19	34.85
1982	10541.24	16857.32	89.87	66.5	65.16	76.96
1979-1982	3854.36	5911.62	50.02	50.33	34.79	42.61

Table -7.7: Showing the comparision of ANN forecast and Box-Jenkins forecast as per the statistical parameters

Period of	Mean and Standard Deviation			Skewness and Kurtosis			Coeff. of Correlation	
Testing	Observed flow	ANN Estimation	Box-jenkin Estimation	Observed flow	ANN Estimation	Box-jenkin Estimation	Observed & ANN	Observed & Box-jenkin
1	2	3	4	5	6	7	8	9
1979	45.250 39.446	59.354 62.059	77.644 82.131	1.314 0.542	1.508 1.189	1.303 0.342	0.949	0.973
1980	79.000 94.286	65.625 71.485	82.328 87.521	1.424 0.807	1.515 1.158	1.174 -0.057	0.919	0.910
1981	122.083 114.295	97.568 93.770	90.768 91.951	0.910 -0.813	0.746 -1.073	1.193 0.147	0.932	0.945
1982	162.500 199.189	121.523 113.383	100.768 91.022	1.076 -0.665	0.937 -0.535	0.796 -0.837	0.949	0.930
1979-1982	102.208 129.549	86.018 88.361	87.877 85.840	1.841 2.725	1.234 0.464	1.000 -0.467	0.914	0.824

Reservoir Simulation Analysis

8.1.0 GENERAL

A modern water resources system may be created, through almost infinite combinations of a large number of system variables. Conventional methods of analysis including the optimization techniques like LP, NLP or DP are practically unable to study the behaviour of complex systems in a continuous fashion. These can consider only selected parts of the system, generally using historic hydrologic data of a limited period of record. However, it is possible to simulate by simplified systems, the behaviour of relatively complex water resources systems for periods of any desired length, to perform numerous and repetitive computations needed for many combinations of the system variables, and finally evolve an optimal or near-optimal design of the system.

A simulation model provides a rapid means for evaluating the anticipated performance of the system, for the given set of design and operating policy parameters. Thus simulation is essentially a search technique, which resembles trial and error approach, used in traditional operation studies, using which a near-optimal solution can be achieved.

In the present study, the simulation technique has been adopted to reproduce the behaviour of the Indravati river basin system. The system was-operated according to various models considered, such as, linear regression based model, non-linear regression based model, conventional standard operating policy and the ANN based model. After obtaining the inherent characteristics and probable responses of the system for each option separately, the best model has been selected.

8.2.0 PROCEDURE

For the above purpose, a customised, menu driven, user friendly computer program is prepared in FORTRAN language under UNIX environment. The following procedure has been adopted for conducting the simulation analysis.

1. Assembling and arranging the basic data in the system in a form, easily handled by the computer.

2. Formulating various operating procedures to serve as fundamental control for the simulation.

3. Code the basic system data and the operation procedure in FORTRAN language to serve as the fundamental control for the simulation.

8.3.0 ASSEMBLING THE BASIC DATA

With reference to the discussions made in the earlier chapters, the period of five years (from 1978 to 1982) was the validation period for testing the relative performance of the various models under consideration. Hence the monthly river inflow record in Mcm, during this validation period constituted the basic hydrologic data input for the simulation model.

Monthly irrigation demands and evaporation values, as earlier used in the case of DP optimization, have been used in this case also. Regarding the system parameters, such as, size of the storage reservoir, its upper and lower bounds, MDDL, MWL, TWL values, efficiencies relating to power generation, power plant capacity, and the discretized values of elevation-area-capacity curve, the same data, as was used earlier in case of DP, has been retained.

The parameters obtained from regression by least squares approach, have been built into the customised program, for formulating the regression based operation policies. As the prime objective of this study is to validate the applicability of ANN, the ANN based model parameters have been suitably accommodated into the data input structure. For reasons, to be described later in subsequent paragraphs, the weight matrix generated by the neural network simulator, which represents the ultimate certificate of learning on part of a neural network, has been kept as an altogether separate input file with its original name intact, i.e., < weights.dat > .

Apart from all above data, the input of prime importance that remains, is to define the initial state and initial period for starting the simulation process. The starting period has been kept as January, 1978, as it is the starting month of the validation phase. The initial reservoir capacity has been kept the same as the dynamic programming optimization scheme, i.e.,2127 Mcm. The reason behind selecting this value is that, keeping all other system constraints and parameters unchanged, this aids in observing the relative performance imitation by various models as compared to the initial DP.

8.4.0 FORMULATION OF OPERATION POLICIES

The operation policies for multiple regression based models have been formulated by framing the corresponding release equations.

A. Linear Regression:

Compute release by using the following equation.

Cal_rel(t) = -23.464 + 0.022 * S(t) + 0.086 * Inf(t) + 0.95*Dem(t) (8.1)

t = 1 to 60,

If Dem(t) > cal_rel(t) and reservoir elevation > MDDL

then Act_rel(t) = Dem(t) (8.2a)

else Act_rel(t) = cal_rel(t) (8.2b)

B. Non-linear Regression:

Compute release by using the following equation.

$$Cal\_rel(t) = 1.875*(S(t))^{0.461} + 0.011*(Inf(t))^{1.269} + 0.105*(Dem(t))^{1.366} \quad (8.3)$$

t = 1 to 60,

If Dem(t) > cal_rel(t) and reservoir elevation > MDDL

then Act_rel(t) = Dem(t) (8.4a)

else Act_rel(t) = cal_rel(t) (8.4b)

Where,

cal_rel(t) = Release computed from the corresponding regression equation during the period t;

act_rel(t) = Actual release to be made during the time period(t);

C. ANN based Model:

It is apparent by now, that the neural network does not yield any parameters after training, rather the intelligence acquired after learning, is distributed within the network in form of connective weights, which cannot be interpreted to have any physical significance. So the model formulation in this case required the embedding of a part of the testing algorithm of the ANN simulator, into this module of the program. The weight matrix, freezed after the completion of training, is given as an input for this module.

The algorithm adopted for obtaining release through the ANN net is,

$$Y_j = \frac{1}{(1+e^{-}(\sum_{i=1}^{3} x_i . W(1,i,j))} \quad (8.5)$$

J = 1 to 4

Where,

y_j = Output of j^{th} hidden layer neuron;

x_i = Scaled down/normalised value of the i^{th} neuron input in the input layer;

W(1,i,j) = Weight connecting i^{th} neuron in the 1^{th} layer with j^{th} neuron in the $(1+1)^{th}$ layer;

The above recursive equation connects the input layer neurons with those of the hidden layer.

and

$$Z_k = \frac{1}{(1+e^{-(\sum_{j=1}^{4} y_j W(1,j,k))})} \qquad (8.6)$$

where,

Z_k = Output of the kth output layer neuron Release during time period t is obtained from $z_k(t)$ scaling it up appropriately.

The equations 8.2a,b or 8.4a,b have also been implemented in this case. These additional constraints are built into the model keeping in view the fact that if water is available, it will not be a prudent decision to cause irrigation deficit by sticking to the model releases.

D. Standard Operating Policy:

A monthly simulation model based on the SOP is also constructed for the system considered. The SOP is formulated as follows. If the available water, during time period t, is less than or equal to the demand, then the available water is released. In the second case, when the water availability is higher than the demand, the quantity equal to the demand only is released and the remaining quantity is stored if possible, otherwise it is spilled.

8.5.0 CODING THE SYSTEM DATA AND OPERATION PROCEDURES

One of the major limitations of the application of simulation analysis to the design of water resources systems is that, it is not flexible in handling various operating procedures of the system. The computer can be instructed to follow only one operating procedure at a time. Thus provision was kept in code, developed in FORTRAN-77, to accept the option from user to select any one of the four operating procedures, described in the previous section and the system is simulated for that operating procedure only. The model has to be re-run for obtaining the results from some other procedure.

The additional model specific input data, apart from those discussed in section 8.3.0, are obtained in an interactive mode. This is especially useful in computing the ANN based model simulation and in fact, this is also helpful in identification of proper structure of the net and for effective training of the architecture. It is always not possible to minimise the Sum Square Error surface to attain a hypothetical zero value or to reach a global minima, because of the inherent shortcoming of the feed forward BP neural networks.These get locked up in local minima points. So every conceivably trained architecture has to be routed through the simulation model also, after finding its mapping ability. This was accomplished by formulating batch programming through UNIX-Macro development.

The scheme adopted for developing this was,

Step-1 *Training the ANN through ANN simulator -training module;*

Step-2 *Testing the ANN through ANN simulator - testing module;*

Step-3 *Filtering the required output from output file or the BP simulator.*

Step-4 *Testing the mapping ability of the trained net in comparison to other operating models;*

Step-5 *Sending the output through the reservoir simulation module, for final testing of the performance.*

As training and testing of an ANN architecture involves various options, like changing the structure parameters, choosing proper scaling parameters and increasing or decreasing testing data length, a number of trial and errors had to be performed. Therefore, instead of adopting cumbersome exercise to attain full compatibility with other programs, some developed in FORTRAN and some developed in C and C++ language, an interactive data input mode was preferred.

As the program developed is a customised one, the regression parameters have been built into the release subroutine. However, with minor modifications, the program can be implemented for other similar cases by assigning variables to the regression parameters. The code generated in FORTRAN is given in Annexure - V.

8.6.0 DISCUSSION ON RESULTS

The simulation model was run for each of the above four options and the reservoir working tables are obtained for a simulation period of five years. For final comparison of the various models, a reservoir behaviour table is prepared with the following reservoir attributes, namely, number of times the model fails in meeting the irrigation demand, average deficit, number of times reservoir goes empty or becomes full or spills in a season within the simulation period, average spill and average power generation.

The statistics of reservoir behaviour are furnished in Table (8.1). The corresponding releases from each model during the simulation period is given in Figure (8.1). It may be observed from the time series plot of the releases that, all

TABLE 8.1 : RESERVOIR BEHAVIOUR DURING SIMULATION PERIOD, FOR VARIOUS MODELS

Period	No of years model failed				Average Deficit in irrigation				No.of times reservoir empty				No.of times reservoir full				No.of times reservoir spilled			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Jan	1	1	1	1	147.1	175	182.7	144.5	1	1	1	1	0	0	0	0	0	0	0	0
Feb	1	1	2	1	224.3	224	131.2	224.3	1	1	2	1	0	0	0	0	0	0	0	0
Mar	2	2	2	2	221.7	220	222.7	162.25	2	2	2	2	0	0	0	0	0	0	0	0
Apr	2	2	2	2	142.45	142	142.45	142.45	2	2	2	2	0	0	0	0	0	0	0	0
May	2	2	2	2	97.35	97.4	97.35	97.35	2	2	2	2	0	0	0	0	0	0	0	0
June	1	1	1	1	24	24	24	24	1	1	1	1	0	0	0	0	0	0	0	0
July	1	1	1	1	77.7	77.7	77.7	77.7	1	1	1	1	0	0	0	0	0	0	0	0
Aug	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	2	1	1	1	1
Sep	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2
Oct	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	1	1	1	1
Nov	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	0	0	0	0
Dec	0	0	1	0	0	0	18.8	0	0	0	1	0	1	1	1	2	0	0	0	0
Total validation period	10	10	12	10	1395.91	1420	1490.47	1274.45	10	10	12	10	8	8	8	10	4	4	4	4

Period	Average spill during time period				Power generation			
	1	2	3	4	1	2	3	4
Jan	0	0	0	0	281.46	258.36	253.88	255.5
Feb	0	0	0	0	291.68	294.9	288.7	289.8
Mar	0	0	0	0	206.44	202.34	201.58	227.8
Apr	0	0	0	0	157.9	134.74	133.7	132.3
May	0	0	0	0	112.04	99.56	99.88	94.6
June	0	0	0	0	114.34	112.78	116.58	101
July	0	0	0	0	365.42	367	366.02	334.7
Aug	9	84.9	86.2	225	186.54	206.16	216.66	242.3
Sep	572.5	597.75	586.45	613.65	490.44	498.74	502.98	478.6
Oct	698.1	698.1	698.1	698.1	441.02	443.36	445.74	438.5
Nov	0	0	0	0	201.7	193.74	193.88	187.4
Dec	0	0	0	0	141.52	147.5	144.7	135.4
Total validation period	1852.08	1978.58	1957.16	2150.4	14952.5	14795.9	14821.5	14590

1 : ANN MODEL ; 2 : LINEAR REGRESSION BASED MODEL ; 3 : NON-LINEAR REGRESSION BASED MODEL ; 4 : SOP MODEL

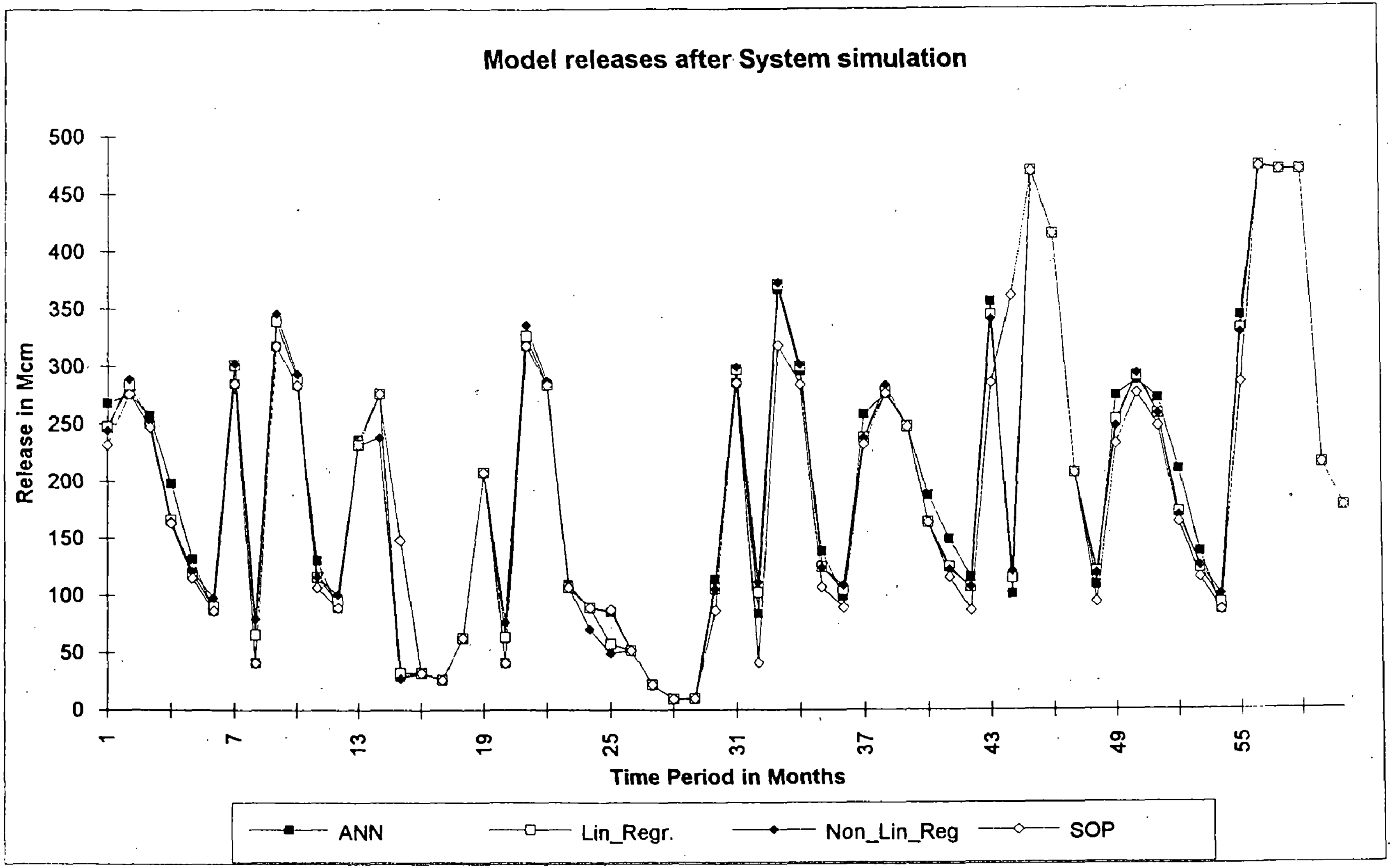


FIG 8.1 : Showing various model releases after system simulation during the validation period of 5 years

the models are more or less on an equal footing and no major deviation occurs amongst them. The reservoir behaviour also remains more or less same. For most of the months, the average deficits, spills and number of times reservoir has gone empty or full, are the same. For a better comprehension of the performance appraisal of the models another summary table is furnished below in Table 8.2.

Table 8.2 : Summary of reservoir behaviour by various models in comparison to the DP optimisation.

Criterion	DP	ANN	Linear	Non_lin	SOP
Cumulat. Generat.	15090	14953	14795.8	14822	14590
Cum. Deficit	1317	1396	1420	1490	1274
Cum. Spill	2041	1852	1979	1957	2150
Failure Months	30	10	10	12	10
Spilling Months	4	4	4	4	4

Some important inferences can be drawn based upon the result obtained from simulation.

1. DP being an optimization tool, is able to generate maximum power. However all other models have been able to reach near the DP performance, while the ANN model performance is the best.

2. SOP gives minimum power generation, but irrigation deficit is also the minimum, i.e., even less than the DP performance.

3. The DP model has maximum number of deficit months. But as

the total deficit is more or less the same, it is obvious that while optimising, the total deficit has been divided into larger number of smaller deficits.

4. The performance of the ANN model is better than all other models, on all the criterions selected for comparison excepting the irrigation deficit, where ANN performance is marginally inferior to the standard operating policy.

Summary and Conclusion

9.1.0 OBJECTIVES AND METHODOLOGY

Objectives of the multi purpose, Upper Indravati Water resources Project are basically two fold : Irrigation and hydro power generation. The objective of the present study herein, is to assess the application potential of the Artificial Neural Network (ANN), in attaining the above project objectives, compared to the conventional models used for the purpose.

The scope of the present study, for optimal utilisation of Indravati Project water resources is two fold : Time-series analysis and prediction and reservoir operation. The study related to the reservoir operation consisted of optimisation through Discrete Differential Dynamic Programming (DDDP) approach, framing appropriate DPR and DPN models and finally screening through a customised simulation model. Time Series analysis consisted of identifying an appropriate multiplicative seasonal model from among the Box and Jenkins ARIMA family of models and predicting one month ahead river inflow into the Indravati reservoir. The prediction procedure is repeated through a prediction model, developed through the ANN approach, under more or less similar conditions. Prediction performance is then simulated through a time series simulator. In the final step, the best of the alternatives from among the competing models have been chosen, in both cases, providing suitable justifications.

9.2.0 AT-SITE RESERVOIR OPERATION

In order to meet the last objective in section 1.2.0 of the introductory chapter of this report , the study on time series analysis for prediction has been included. It serves in providing a pragmatic touch to the finalised reservoir operation model, i.e., with the help of one-month-ahead forecast, an at-site reservoir operation module can be monitored.

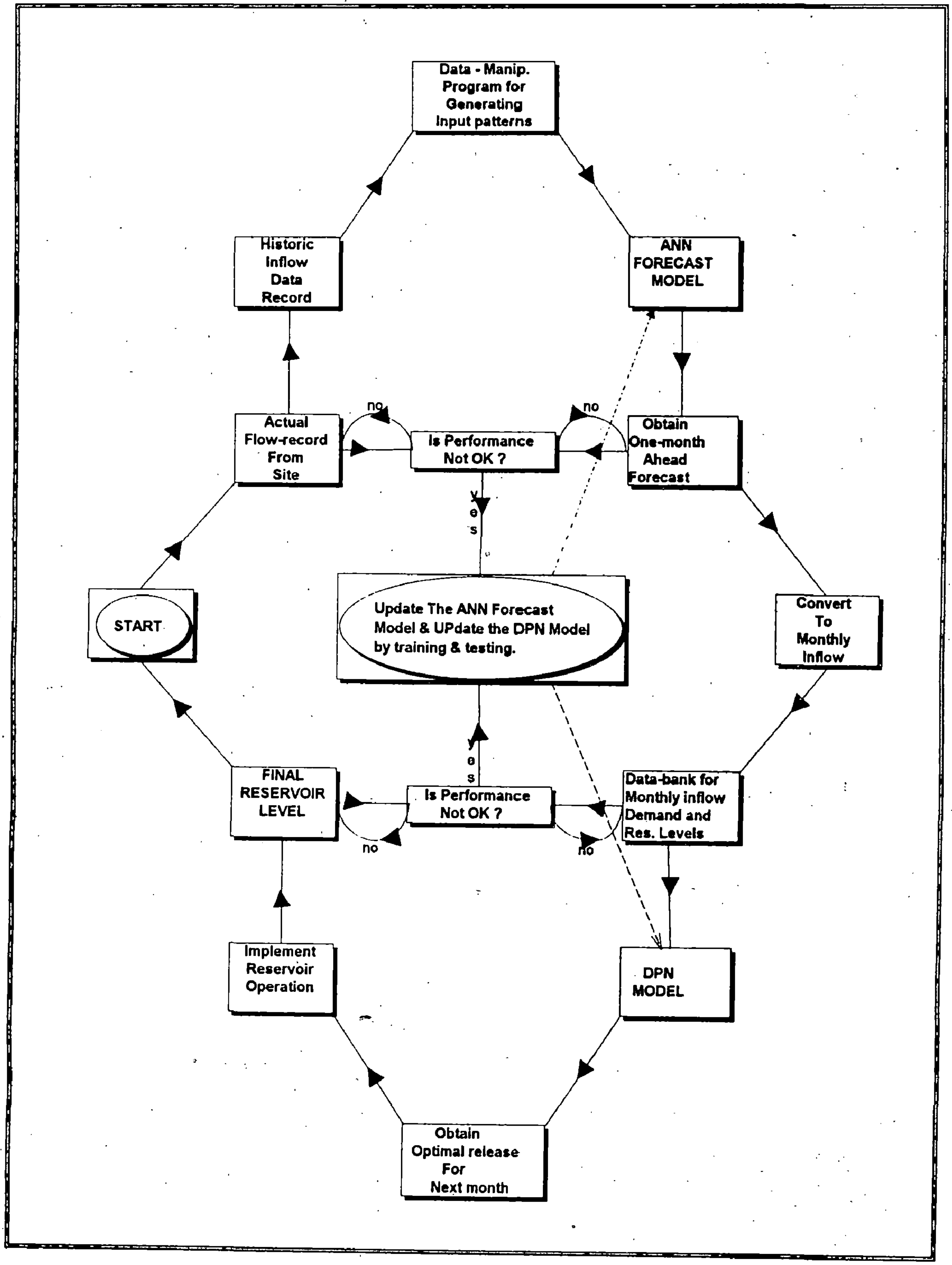
A reservoir operation monitoring scheme can consist of obtaining the prediction from the ANN forecast model, feed it through the DPN model for obtaining monthly optimal release. This can be phased over 30 days in the conventional way. During the month end, the inflow data record will be updated with actual flow value. This process can be repeated during every month end. For future feedback to the system, after the arrival of actual record, the deviation should be monitored for performance appraisal of the models. When it is considered to be obsolete the weight matrix has to be rebuilt by following the entire stage of training, testing and validating, as was discussed in previous chapters. A tentative model outline for a reservoir monitoring scheme is furnished in Figure (9.1).

9.3.0 LIMITATIONS OF THE STUDY

Major constraints for this study were mostly time, appropriate tools and relevant literature. The present study envisages application of DDDP, ARIMA modelling, system simulation and ANN, in a single reservoir, multi objective, Water Resources Project. Various new procedures are now available to strengthen the methodologies, procedures and loopholes, in constructing the models in above areas. Time acted as the major constraint in reaching these procedures. Various limitations, encountered while carrying out this study were listed below.

1. For optimising the release, DDDP concept has been used in the present study. It is clear that a single DDDP iteration will typically be much less expensive than a DDP solution. However DDDP effort still grows exponentially with state

FIG 9.1 : A TENTATIVE MODEL FOR AT-SITE RESERVOIR OPERATION MONITORING SCHEME



dimension n and hence not totally free from *curse of dimensionality*. As SIDP is able to sidestep the curse, that could have been tried to reach a better optimisation.

2. The loss function, used in the final solution of DP objective function, is selected from among a limited number of trials. Experience and availability of time could have resulted in better search for this function.

3. For the Time-Series modelling through ARIMA approach, it could be seen that the spurious autocorrelations generated at the seasonal lags could not be removed even after differencing and standardisation, as complained by many authors previously. With the help of better diagnostics and employing IACF and IPACF criterion, this might have been removed. In this study it has been shown that log transformations could not yield better results. But by adding Box-Cox transformation algorithm into the identification module of the Time-Series program, better identification might have been possible.

4. ANN application potential has been tested in this study in the areas of reservoir operation and univariate river flow prediction. Within the purview of standard journals available, only one publication was available in application of ANN in reservoir operation having a single objective *[Raman et al. 1996]*. Therefore, the present study attempting to accommodate two objectives of irrigation and power generation, is likely to contain many seen and unseen errors and omissions.

5. The BP simulator used for the study is a customised one, based upon the basic fundamentals of *generalized delta rule*, without having any graphic interface. An instantaneous display of the error surface and location of minima would give possibly, better insight into the error minimisation ability of the model. Better alternatives in the BP algorithm like BPA, BPX (available in MATLAB software) or the Linear Least Square with SIMplex (LLSSIM) algorithm could have been tried to

improve the result.

6. The entire study is based upon the output of DP model. In this study, a compromise is made with the prime objective of power generation. The maximum generation during any time period was fixed at installed capacity of the project. In case of availability of the actual load curve, the same could be incorporated into the DP constraint domain and more realistic results could be obtained.

9.4.0 CONCLUSIONS

A study of application of ANN has been carried out in designing a pragmatic reservoir release scheme for Upper Indravati Project. A step-by-step account of problems encountered during the identification, construction and implementation of various models is given across the length of this report.

A new technique of shuffling of the training data set has been introduced in this study, for breaking out the input pattern sensitivity of the BP algorithm. This has been implemented successfully and suggested for further study and research.

The present study successfully demonstrates the utility of Artificial Neural Networks to become a strong, effective and viable compliment to the existing conventional optimization techniques, although not as a complete substitute. The purpose of simulation techniques in refining the optimization or prediction results has been established.

REFERENCES

Box, G. E. P., G. M. Jenkins, Time Series analysis, Forecasting and Control, *Holden-Day Press, San Franscisco, California, USA, 1976.*

Bolzern, P., M. Ferrario, G. fronza, Adaptive real time forecast of river flow rates from rainfall data, *Journal of Hydrology, 47, pp251-267, 1980.*

Bhaskar, A.N., E.E. Whitlatch, Comparison of reservoir linear operation rules using linear and dynamic programming, *Water Resources Bulletin, Vol-22, No.6, Dec,1987.*

Carriere, Patrick, shahab Mohaghegh, Razi Gaskari, Performance of a virtual runoff hydrograph system, *Journal of water resources Planning and Management, Vol.121, No.6, pp.421-427, Nov/Dec,1996.,*

Chatfield, C., D. L. Prothero, Box-Jenkins seasonal forecasting: problems in a case study, *Read before the Royal Statistical Society on 17. 1.1973.*

Dayhoff, Judith, E., Neural Network Architectures: An Introduction, *Van Nostrand Reinhold, Newyork.*

Delleur, W. Jaques, M. Levent Kavvas, Stochastic models for monthly rainfall forecasting and synthetic generation, *Journal of applied meteorology, Vol-17, Oct, 1978.*

French, N. Mark, Witold F. Krajewski, Robert R. Cuykendall, rainfall forecasting in space and time using a Neural network, *Journal of Hydrology, 137(1992), pp.1-31.*

Galvao, Carlos de Oliveira, Shuichi Ikebuchi, Tohiharu Kojiri, Reservoir operation and climate forecasting in a tropical region, Proceedings of the International Conference on Water Resources and Environmental Research, Vol.-II, Kyoto, Japan, *Oct29-31, 1996*

Goel, N,K., Lecture notes, Stochastic Hydrology,(un published), U.O.R., March, 1997.

Gourbesville, Philippe, The Stochastic Approach : a solution for Hydrological analysis and forecasting, Proceedings of the International Conference on water resources and environmental research, Vol.-II, Kyoto, Japan, *Oct29-31, 1996,*

Gupta, Hashin V., Kuo-lin Hsu, Sorosh Sorooshian,

(A) Superior training of Artificial Neural Networks using weight space partitioning. *(To be published)*

(B) An Artificial Neural Network Model for rainfall estimation from satellite Infrared imagery. *(To be published)*

(C) Application of Recurrent Neural Networks to rainfall runoff modelling *(Accepted for publication in the 1997 ASCE Water resources Planning and Management Division Conference, April 7-10, Houston, Texas)*.

(D) Precipitation estimation from remotely sensed information using Artificial neural Networks *(Accepted for publication in Journal of Applied Meteorology, Feb, 1997)*

Hammerstorm, Dan, Working with Neural Networks, *IEEE spectrum, July, 1993.*

Heidari. M., V.T. Chow, P.V. Kokotovic and D.D. Meredith, Discrete Differential Dynamic Programming approach to water resources system optimization, *Water Resources Research, 7(2), 273-282, 1971.*

Hipel, Keith W., Time series analysis -in perspective, *Water Resources Bulletin, American water resources association, Aug, 1985.*

Hipel, Keith William, Angus Ian Mclead, William C. Lennox, Advances in Box-Jenkins Modelling 1. Model construction, *Water resources research, Vol. 13, NO. 3, June, 1977.*

Jain S.K., B. Soni, P. V. Seethapathi, An optimization technique for water resources management, *journal of Civil engineering division, The Institution of Engineers (India), Vol. 69, Part-II, 1988.*

Karmouz, Mohammad, Mark H. Houck, Comparison of Stochastic and deterministic dynamic programming for reservoir operating rule generation, *Water Resources Bulletin, Vol-23, No.1, Feb, 1987.*

Lawrence, A.J., N. T. Kottegoda, Stochastic modelling of river flow time series, *Journal R. statistical Soc. A, 140,part 1,p.1, 1977.*

Little,J.D.C., The use of storage water in a hydroelectric system, *Operation Research,3, pp187-197, 1955*

Loke, E., Introduction to Artificial Neural Networks in urban Hydrology, *Draft #4, 13 February, 1997.*

Lorrai, M., G.M. Sechi, Neural nets for modelling rainfall runoff transformations, Water Resources Management, 9:299-313, 1995.

Mclead Angus Ian, Hipel, Keith William, William C. Lennox, Advances in Box-Jenkins Modelling 2. Applications, *Water resources research, Vol. 13, NO. 3, June, 1977.*

Minns, A.W., M.J. Hall, Artificial Neural Networks as rainfall runoff models, *Hydrological Sciences Journal, 41(3), June, 1996.*

Mohanty, B.R., Optimization study of Upper Indravati Project, Orissa, *M.E. Dissertation , Department of Hydrology, University of Roorkee, Roorkee, 1994.*

Neural network toolbox, For use with MATLAB: User's guide, *The mathworks Inc.*

O' Connell, P. E., R.T. clarke, adaptive hydrological forecasting - a review, *Hydrological Sciences Bulletin, 26,2(6).*

Project engineering and construction soundness assessment for UPPER Indravati Hydroelectric Project, *Phase-1, Vol-II, (march, 1994), BBSR, Orissa.*

Raman, H., N. Sunilkumar, Multivariate modelling of Water Resources time-series using Artificial Neural Networks, *Hydrological Sciences Journal, 40(2), April,1995.,*

Raman, H., V. Chandramauli, Deriving a general operating policy for reservoir using Neural Network, *Journal of water resources Planning and Management, Vol.122, no.5, pp.342-347, Sep/Oct,1996.*

Rao, Valluru, Hayagriva Rao, C++ Neural Networks and Fuzzy Logic, Second edition, *BPB Publications, 1997.*

Rumelhart, David, E., James L. mclelland and the PDP research group, Parallel Distributed Processing, Explorations in the Micro structure of Cognition, Vol.I:Foundations, *The MIT Press, Cambridge, Massssachusetts, London, England, 1986.*

Salas, J.D., J.W. Deulleur, V. Yevjevich, W.L. Lane , Applied modelling of Hydrologic Time Series, *Water Resources publication, Littleton, Colorado, USA, [1980].*

Sedimentation Assessment Study, Final report, Vol-II, Dept. of Energy, PA, UIHEP, Govt. of Orissa, 1995

Seethapathi, P.V., S.K. Jain, Optimum reservoir operation using dynamic programming, *User's mannual, report Um-17, National Institute of Hydrology, 1985-86*

Shapiro, C. Stuart, Encyclopedia of artificial intelligence, Second edition, *Vol - II, pp 1016-1057, John Wiley & Sons, Inc.*

Smith, Jason, Robert N. Eli, Neural Network models of rainfall-runoff process, *Journal of water resources Planning and Management, Vol.121, No.6, Nov/Dec,1996.*

Vermury,V.R., and R.D. Rogers (eds.), Artificial Neural Networks: Forecasting Time Series, *IEEE compu. Soc. Press, Los Atlantas, California, 1994.*

Yakowitz, Sidney, Dynamic programming applications in water resources,*Water Resources Research, 18(4), 673-696, Aug,1982.*

Zurada, M,Jacek, Introduction to Artificial Neural Systems, *Jaico Publishing house, 1996.*

PRINCIPAL PROJECT DATA

Reservoir :

Catchment area	2630 km^2
Catchment area of Indravati river	1153 km^2
Catchment area of Indravati river at Intake site	530 km^2
Area of full reservoir(FRL RL. 642.00m)	110 km^2
Storage - live	1485.50 Mm^3
- dead	314.50 Mm^3

Dams :

Indravati dam - Masonry Gravity Type.

Length overall		550 m
Non over flow sections		426 m
Spillway		129 m
Dam crest level	EL	645 m
Width		7.5 m
Maximum height of dam		45 m
Spillway crest level	EL	629.5m
Gates, radial no. width/height		7 -15.0 X 12.5 m
Capacity M.W.L. EL 643.00		11430 m^3 /s
Depletion sluices no - width/height		4 - 2.0 X 3.0 m
Discharge M.W.L. EL 643.00		555 m^3/s

Podagada dam - Homogeneous Earthfill

Length		462 m
Crest level	EL	646 m
Width		9.0 m
Parapet height		1.0 m
Maximum height		64 m

Depletion sluices, number - 1	
Diameter	1 - 6.2 m dia
flow, M.W.L. 643.00m	644 m^3 /s

Kapur dam - Homogeneous Earthfill

Length	537 m
Crest level	EL 646 m
Width	9.0 m
Parapet height	1.0 m
Maximum height	71 m

Muran dam - Masonry, Gravity

Length overall	494 m
Non over flow sections	403 m
Spillway	91 m
Dam crest level	EL 645 m
Width	7.5 m
Maximum height of dam	65 m
Spillway crest level	EL 629.5 m
Gates, radial no. width/height	5 -15.0 X 12.5 m
Capacity M.W.L. EL 643.00	8060 m^3 /s

Dikes - Homogeneous Earthfill

	Dike Number	Height (m)	Crest length (m)
left	1	30	553
	2	20	320
	3	15	680
	4	20	160
right	1	15	463
	2	20	146
	3	25	593
	4	15	535

Crest level	EL	646 m
Width		7.0 m
Parapet height		1.0 m

Link Channels	Length(m)	Bed EL(m)	Bedwidth(m)	Slope
Gunturkhal	1523	613.00	75	1:1
Kusumpadar	1550	620.00	23.5	1.5:1

Developed Head

Maximum Gross (FRL 642.00, TWL 263.00)	379 m
Minimum Gross (MDDL 625.00, TWL 267.00)	358 m

Water Ways

Head race Channel

Length	335 m
Width , Min.	37.5 m

Head race Tunnel - 7.0m ID

Intake, horizontal inlet type with trash racks

Gate size : no - width/height	1 - 5.75 / 8.00m
Tunnel - Length	3,934 m
design flow	200 M^3/sec
Lining, concrete	0.3 m

Surge Tank - Restricted orifice type

Diameter		20 m
Height		132 m
maximum surge levels	EL	670.9 m
minimum surge levels	EL	622.2 m
Lining, concrete		0.45 m

Pressure Tunnels - 2 - 5.25 m ID

Length	298 m
Lining thickness	0.3 m

Steel liner ASTM A 537 class 2 with stiffners

Liner plate thickness		16 mm

Penstocks: 4 - 3.50 m ID

length		790 m
steel shell ASTM A 537 class 2		
Shell plate thickness	max.	38 mm
	min.	22 mm

Tail race Channel :

Length		9 km
Width, bed	max.	20 m
	min.	18 m
Concrete lining thickness		0.15 m
Discharge, 4 units		200 m^3/s
Depth max. flow		4 m

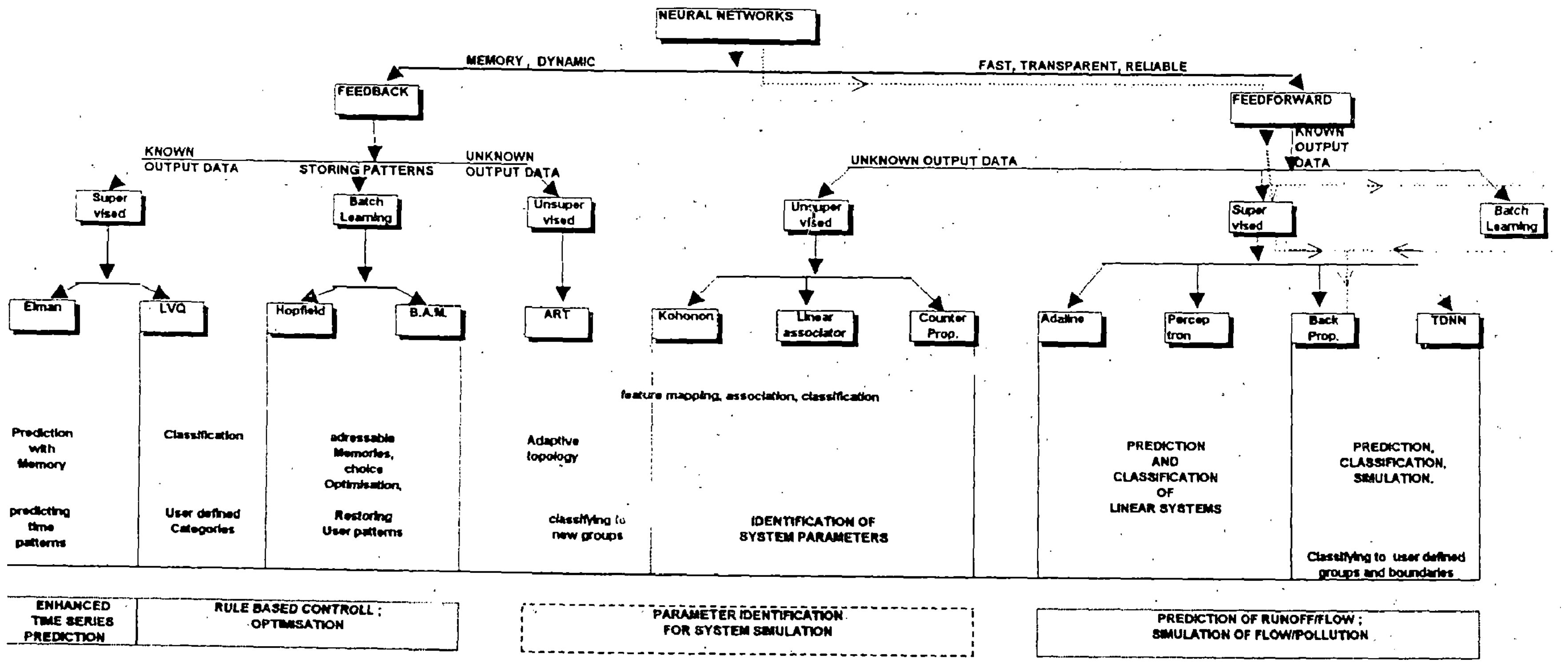
Power Generation

Installed Capacity	4 -	150 MW units
Average annual power(100 % load factor)		252 MW
Average annual generation		2.206 G.Wh.

Irrigation Works

Location - River Hati near Mangalpur		
Head-works : Barrage		
radial Gates :	6 - 12 X 6 m	
	4 - 6 X 8 m	
Length	EL	117.00 m
Crest level	EL	259.00 m
Max. Pond level	EL	265.00 m
Design discharge		64.3 m^3 /s
pondage between EL 265 to 260.4 m		604 Ham
Distribution System		
G.C.A.	1,35,700 Ha.	
C.C.A.	1,28,000 Ha.	
Annual Irrigation	1,85,800 ha.	

ANNEXURE - II



Note : The line shown dotted with arrow marks, shows the path for the currently adopted NN model in the classification tree.

CLASSIFICATION TREE OF ARTIFICIAL NEURAL NETWORKS IN HYDROLOGY

```
#include <stdio.h>
#include <iostream.h>
# include <stdlib.h>
# include <time.h>
/*      Programmed by        : Ashutosh Dash
        Date of Last Update : 10.11.97
The program is a customised database having the scope for accepting a
two-dimensional arrayed data file of any size, scaling down any particular
column/columns as per user's wish, showing the maximum and minimum value of
any column,seasonalising the entire data set into 12 output files, randomized
suffling of the patterns, after recording the sequnce of randomization in an
output file <ref.out>, and inter changing the columns as per user requirement
or else purging some of the columns.*/


static int array[400];
inline int guess (int n) {
      return 1+int((float(rand())/32768.0)*n);
}

void random(int n)
{
      time_t *t = new time_t;

      int seed = int (time(t)) % 10000;
      srand(seed);

      int x;
      array[0] = guess(n);
      short int z;
      for (int i=1; i<n; i++) {
            z = 1;
            while (1) {
                  x = guess(n);
                  for (int j=0; j<i; j++) {
                        if (x == array[j]) {
                              z=0;
                        }
                  }
                  if (z==0) {
                        z=1;
                        continue;
                  }
                  else {
                        array[i] = x;
                        break;
                  }
            }
      }
}
main(int c, char *v [])
{

      FILE *fpt1,*fpt2,*fpt3;
      int n,count=0,count1=0,m,p[12],col,i;
      float data[400][12],q[12],mm[12];
      char tag[9],tag1[9];
      void season(float [][12],int,int);

      if (c != 3) {
            cout << "\nERROR: Argument missing.\n\n";
            cout << "Usage: " << v[0] << " input_file output_file\n\n";
            return(1);
      }
```

```
	fpt1=fopen(v[1], "r");
	fpt2=fopen(v[2], "w");
	fpt3=fopen("ref.out","w");

	printf("enter the length of data set :");
	scanf("%d",&n);
	printf("\nenter the width of data set :");
	scanf("%d",&m);

	for(count=0;count<m;count++)mm[count] = -999;

	for(count=0;count<n;count++) {
		for(count1=0;count1<m;count1++)
		{	fscanf(fpt1,"%f",&data[count][count1]);
if (data[count][count1] > mm[count1]) mm[count1] = data[count][count1];
	 }}
	cout<<"\n\n Want to segregate data into 12 seasons(y/n) :";
	cin>>tag1;
	if(tag1[0]=='y')
	{ season(data,n,m);}
	else
		{

	printf("\n Enter the total no of columns you wish to change : ");
	scanf("%d",&col);
	printf("\n Total %d columns to change",col);
	for(i=0;i<col;i++){
		printf("\nenter the %d column no ",(i+1));
	scanf("%d",&p[i]);
	printf("\n maxm. value of the column %d : %f",p[i],mm[p[i]-1]);
	printf("\n\n Now enter the transformation factor : ");
	scanf("%f",&q[i]);
	}
	for(count1=0;count1<n;count1++)
		for(i=0;i<col;i++)
			data[count1][p[i]-1]/=q[i];

	cout<<" Want to randomize the output sequence (y/n) :";
	cin >> tag;
	if(tag[0]=='n')

	{
		for(count=0;count<n;count++){
			for(i=0;i<col;i++){
				fprintf(fpt2,"%f\t",data[count][p[i]-1]);
			}
			fprintf(fpt2,"\n");
		}
		fclose(fpt1);
		fclose(fpt2);
	}

	else	if(tag[0]=='y')
	{
		random(n-1);
		for(count=0;count<n;count++){
			for(i=0;i<col;i++){
				fprintf(fpt2,"%f\t",data[array[count]][p[i]-1]);
			}
			fprintf(fpt2,"\n");
		}
	for(i=0;i<n;i++) {
		fprintf(fpt3,"%d\t",array[i]);
		if ((i+1) % 10 == 0) fprintf(fpt3, "\n\n");
	}
```

```
			fclose(fpt1);
			fclose(fpt2);
			fclose(fpt3);

		}
}
}
void season( float data[400][12],int n,int col)
{
FILE  *fpt4,*fpt5,*fpt6,*fpt7,*fpt8,*fpt9,  *fpt10,  *fpt11,  *fpt12,  *fpt13
,*fpt14,*fpt15;
int i;
	fpt4=fopen("jan.dat","w");
	fpt5=fopen("feb.dat","w");
	fpt6=fopen("mar.dat","w");
	fpt7=fopen("apr.dat","w");
	fpt8=fopen("may.dat","w");
	fpt9=fopen("jun.dat","w");
	fpt10=fopen("jul.dat","w");
	fpt11=fopen("aug.dat","w");
	fpt12=fopen("sep.dat","w");
	fpt13=fopen("oct.dat","w");
	fpt14=fopen("nov.dat","w");
	fpt15=fopen("dec.dat","w");

		for(int count=0;count<n;count++){
if (data[count][1]==1.)
{	for(int i=0;i<col;i++)
				fprintf(fpt4,"%f\t",data[count][i]);
				fprintf(fpt4,"\n");}else
if (data[count][1]==2.)
{	for(i=0;i<col;i++)
				fprintf(fpt5,"%f\t",data[count][i]);
				fprintf(fpt5,"\n");}else
if (data[count][1]==3.)
{	for(i=0;i<col;i++)
				fprintf(fpt6,"%f\t",data[count][i]);
				fprintf(fpt6,"\n");}else
if (data[count][1]==4.)
{	for(i=0;i<col;i++)
				fprintf(fpt7,"%f\t",data[count][i]);
				fprintf(fpt7,"\n");}else
if (data[count][1]==5.)
{	for(i=0;i<col;i++)
				fprintf(fpt8,"%f\t",data[count][i]);
				fprintf(fpt8,"\n");}else
if (data[count][1]==6.)
{	for(i=0;i<col;i++)
				fprintf(fpt9,"%f\t",data[count][i]);
				fprintf(fpt9,"\n");}else
if (data[count][1]==7.)
{	for(i=0;i<col;i++)
				fprintf(fpt10,"%f\t",data[count][i]);
				fprintf(fpt10,"\n");}else
if (data[count][1]==8.)
{	for(i=0;i<col;i++)
				fprintf(fpt11,"%f\t",data[count][i]);
				fprintf(fpt11,"\n");}else
if (data[count][1]==9.)
{	for(i=0;i<col;i++)
				fprintf(fpt12,"%f\t",data[count][i]);
				fprintf(fpt12,"\n");}else
if (data[count][1]==10.)
{	for(i=0;i<col;i++)
				fprintf(fpt13,"%f\t",data[count][i]);
```

-(SOURCE CODE FOR THE DATA_MANIP1 PROGRAM DISCUSSED IN CHAPTER-7)-

```
# include <stdio.h>
# include <math.h>
float a[400],mean,sdv,min=+999,max=-999.;
int n,ldiff=0;
/*   Programmed by : Ashutosh Dash
     Date of last update : 28.9.97
Scope:  The program accepts a one dimensional array data file and
converts to multi column output after optionally standardising,
differencing, scaling the the entire data range into 0 and 1,
with scope for complete user interface. This program was useful
in preparing the data into input patterns for ANN modelling. */


main()
{
     FILE *fpt1,*fpt2;
     int count=0,l,lag[10],count1;
     void standard(void);
     fpt1=fopen("column.in","r");
     fpt2=fopen("column.out","w");
     fscanf(fpt1,"%d %d",&n,&l);
     printf("n and l values are %d %d",n,l);
     for(;count<l;count++)fscanf(fpt1,"%d",&lag[count]);
     for(count=0;count<n;count++)fscanf(fpt1,"%f",&a[count]);
     printf("\n Wish to standardise or difference the data
(y/n):");
     if(toupper(getchar()) =='Y')standard();
     printf("Wish to scale the data between 0 and 1 (y/n)");
     getchar();

     if(toupper(getchar()) =='Y'){
          for(count=ldiff;count<n;count++)
          {
               if(a[count] < min) min=a[count];
               if(a[count] >max) max=a[count];
          }
          printf("minimum and maximum values are %f %f",min,max);
          for(count=ldiff;count<n;count++)

               a[count]=(a[count]-min)/(max-min);
     }
     for(count=(ldiff+lag[0]);count<n;count++)
     {
          for(count1=0;count1<l;count1++)
               fprintf(fpt2,"%6.2f\t",a[count-lag[count1]]);
          fprintf(fpt2,"\n");
     }
     fclose(fpt1);
     fclose(fpt2);
```

```
}

void standard(void)
{
     float sum=0;
     int i=0;
     printf("\n  Wish to standardise (y/n)   :");
     getchar();
     if(toupper(getchar()) =='Y'){

          for(;i<n;i++) sum+=a[i];
          mean=sum/(float)n;
          for(i=0;i<n;i++) sdv+=pow((a[i]-mean),2.);
          sdv = pow((sdv/(float)n),0.5);
          printf("\n Mean and S.D. are %f %f ",mean,sdv);
          for (i=0;i<n;i++) a[i] =(a[i]-mean)/sdv ;
     }
     printf("\n  Wish to do differencing also (y/n)   :");
     getchar();
     if(toupper(getchar()) =='Y'){
          printf("\n Enter the lag for differencing : ");
          scanf("%d",&ldiff);
          for (i=n-1;i>=ldiff;i--) a[i] -= a[i-ldiff];
     }
     return;
}
```

ANNEXURE - V

```
C*******************************************************************
c        SIMULATION STUDY FOR INDRAVATI RESERVOIR
c        DATE OF LAST UPDATE : 10.11.97
c        Customised program with options for simulating four different
c        reservoir operation policies, namely, Multiple Linear Reression,
c        Multiple non-linear Regression, ANN based Model and
c        Standard OPerating Policy.
C*******************************************************************
common/elev/elev(20),area(20),stor(20),m,smax,smin
         common/rel/dem(12),ainf(30,12)
         common/ann/n_layer,amult_in,amult_out,layers(5),weights(5,10,10)
         dimension evap(30,12),s_init(30,12),def(30,12),el(30,12),
   1     spil(30,12),a(30,12),relc(30,12),ev(12)
   2     ,gen(30,12),temp1(30,12)
         integer fail(30,12),fail_countm
         real net_av
         CHARACTER*80 TITLE
         DTIM=30*24*3600
         amil = 10**6
C*****************************************
c        Elev (m), area (sq m), cap(cu m), relc(cumec), epd(cm)
     OPEN(1,FILE = ('simul.in'))
     OPEN(2,FILE = ('simul.out'))
         READ(1,1) TITLE
1        FORMAT(A)
         write(*,113)
113      format(10x,'^[[1m^[[5m** SIMULATION MODEL HAS FOLLOWING OPTIONS **^[[0m'//////
   1     4x,'1      MULTIPLE LINEAR REGRESSION MODEL '/
   2     4x,'2      MULTI VARIATE NON LINEAR REGRESSION MODEL '/
   3     4x,'3      ANN BASED MODEL '/
   4     4x,'4  OR ANY OTHER NUMBER FOR STANDARD OPERATING POLICY'////2x,
   1     '^[[1m^[[5mPlease enter your choice :^[[0m ')
         read(*,*)code
         READ(1,*)m,ny
         READ(1,*) smax,smin,init_month,s_init(1,init_month)
         read(1,*)  capins, twl
         read(1,*) (elev(I),area(I),stor(I),I=1,m)
         read(1,*) (ev(i), i=1,12)
          read(1,*) (dem(j),j=1,12)
          do j=1,12
            dem(j) =dem(j)*1.e06
          enddo
      read(1,*)((ainf(i,j),j=1,12),i=1,ny)
         do i=1,ny
         do j=1,12
         ainf(i,j)=ainf(i,j)*amil
         enddo
         enddo

C*****************************************
c        INITIALISATION OF VARIABLES
C*****************************************
         cum_gen=0
         cum_def=0
         cum_spil=0
         fail_countm=0
         fail_county=0
C*****************************************
c        WRITE INPUT DETAILS
c*****************************************
         WRITE(2,900) TITLE
900      FORMAT(//A//20X,'*** I N P U T  D A T A ***'///5X
   1     'All Data are in MKS Units')
    if(code.eq.1)then
         write(2,*)'OPTION : MULTIPLE LINEAR REGRESSION MODEL '
         else if(code.eq.2)then
         write(2,*)'OPTION : MULTI VARIATE NON LINEAR REGRESSION MODEL '
```

```
      else if(code.eq.3)then
           write(2,*)'OPTION : ANN BASED MODEL '

c          This program accepts one input pattern from user and
c          sends the data through the neural net and computes
c          the output.Inputs are to be compatible with the neural net.
c          Neurons in each layer and of course a compatible weights
c          matrix in < weights.dat> file are to be given.
c          date : 11th sept. 97
c          programmed by Ashutosh Dash
c          ************************************************************
           open(4,file='weights.dat')
           write(*,*)'Total no. of layers in the neural net :'
           read(*,*) n_layer
           write(*,*)'Enter the multiplication factor for the input patterns'
           read(*,*)amult_in
           write(*,*)'Enter the multiplication factor for the output'
           read(*,*)amult_out
           write(*,*)'neuron architecture in the neural net :'
           read(*,*)(layers(i),i=1,n_layer)
           write(*,*)'Weight matrix of neural net :'
           do k=1,n_layer-1
           do i=1,layers(k)
           read(4,*)ilop,(weights(k,i,j),j=1,layers(k+1))
           write(*,*)ilop,(weights(k,i,j),j=1,layers(k+1))
           enddo
           enddo
           endif
           write(2,901) ny
901        FORMAT(/5X'The number of years for analysis :'i3/
   1       5x,'The computational time interval : one month')
      write(2,902) SMAX/amil, SMIN/amil
902        FORMAT(/5x'Maximum storage capacity :'f11.3' million cum'/
   1  5x 'Minimum storage capacity :'f11.3' million cum')
           write(2, 907) capins, twl
907        format(/' Installed Capacity :'f6.0' MW'/' Tail Water'
   1  ' Level :'f8.3 ' m')
           write(2,903) (I,ELEV(I), AREA(I),STOR(I),I=1,m)
903        FORMAT(/10x,'Elevation - Area - Capacity' ,
   1       'Table'//
   2       2x,' S N     Elevation     Area       Capacity   '
   3       :/
   4       2x,'           ( m )      (sqm)       (cum)      '
   2       //(2x,i4,f14.3,2f14.1))
911        FORMAT(I5,i3,F9.1,3f8.1,i3.1x,f6.1,1x,f8.1,f7.1,f9.1,F9.1)
c  ************************************************************
c          STARTING THE SIMULATION
c  ************************************************************
           do  555 i=1,ny
            write(2,905)
905        FORMAT(//20x,' **** Results of Calculations ****'//
   1       ' YYYY-MM Init-Stor Inflow  Demand  Release  fail',
   2  ' spil   av_H    Pow_G  Evap   Fin-Stor'/
   3  '         (Mil Cum) (m cum) (Mcm)  (m cum) (m cum)',
   4  '  (m)     (MW)   (Mcm)   (Mcm)'/)
           if(i .eq. 1) then
           pj = init_month
           else
                     pj=1
           endif
           do mj=pj,12
           j=mj
           flag=0
           temp=0
           avst=s_init(i,J)
10         call area_elev(avst,a(i,j),el(i,j))
           evap(i,j)=a(i,j)*ev(j)/100.
           net_av = s_init(i,j)+ainf(i,j)-evap(i,j)-smin
           if(net_av .lt. 0) then
           temp =net_av
           net_av =0
```

```
        flag=1
        go to 19
        endif
c       calculate the model release  and fix objective

        rel=release(code,s_init(i,j),ainf(i,j),dem(j))
        if(rel .gt. dem(j)) then
        obj=rel
        else
        obj = dem(j)
        endif
        if(net_av .le. obj) relc(i,j) = net_av
        if(net_av .gt. obj) relc(i,j) = obj
        write(*,*)'dem rel obj relc',i,j,dem(j),rel,obj,relc(i,j)
        net_av = net_av - relc(i,J)
        if(net_av .gt. (smax-smin)) then
        relc(i,j) = relc(i,j)+net_av-(smax-smin)
        net_av =smax-smin
        endif
19      avst=(s_init(i,j)+(net_av+smin))/2.
        call area_elev(avst,a(i,j),havg)
c       if(abs(havg-el(i,j)) .ge. .1) go to 10
        def(i,j)=dem(j)-relc(i,j)
        if(def(i,j).lt.0) def(i,j)=0
        fail(i,j)=0
        if(def(i,j).gt.0)fail(i,j)=1

c**       Generation in kwh(energy),  kw(power)
          orel = relc(i,j)/2592000        ! rel in cumec, 30*86400=2592000
          gen(i,j) = orel*(havg-twl)*8.79          ! 9.8*0.92*0.975 = 8.79
          if(gen(i,j).gt.600000) then
           gen(i,j)  = 600000.0
        orel=gen(i,j)/(havg-twl)/8.79
        temp1(i,j) = relc(i,j)
        relc(i,j)=orel*2592000
        net_av=net_av+(temp1(i,j)-relc(i,j))+smin
        if(net_av .gt. smax)then
        spil(i,j) = (net_av-smax)
        net_av=smax
        endif
        else if(gen(i,j).le.600000)then
        if(flag.eq.1)net_av=temp
        net_av=net_av+smin
        endif
        if(j .eq. 12) then
        s_init(i+1,1) = net_av
        else
        s_init(i,J+1) = net_av
        endif
c       ******************************
c       COMPUTE THE CUMULATIVE QUANTITIES
c       ******************************
        cum_gen=cum_gen+gen(i,j)
        cum_def=cum_def+def(i,j)
        cum_spil=cum_spil+spil(i,j)
        fail_countm=fail_countm+fail(i,j)
        write(*,*)'fail count details',i,j,fail(i,j),fail_countm
        write(2,911)i,j,s_init(i,j)/amil,ainf(i,j)/amil,
   1 dem(j)/amil,relc(i,j)/amil,fail(i,j),spil(i,j)/amil,havg,
   2 gen(i,j)/1000.,evap(i,j)/amil, net_av/amil
          enddo
555     continue
        write(2,126)cum_gen/1000.,cum_def/1.e06,
   1    cum_spil/1.e06,fail_countm
126     format(2x,'cumulative generation :',f9.2/
   1    2x,'cumulative deficit : ',f9.2/
   2    2x,'cumulative spillage : ',f9.2/
   3    2x,'total months model failed :',i3/)
        stop
        end
```

```
      function release(code,stor,inflow,dem)
      common/ann/n_layer,amult_in,amult_out,layers(5),weights(5,10,10)
      dimension transit(5,10,10)
      real inflow,input(5,10,10)
      if(code .eq. 1)then
      release=-23.464+0.95*dem/1.e06+.086*inflow/1.e06
     1 +.022*stor/1.e06
       release=release*1.e06
      return
      else if(code .eq. 2) then
      release= (.011*(inflow/1.e06)**1.269 +
     1 .105*(dem/1.e06)**1.366
     1 +1.875*(stor/1.e06)**.461)*1.e06
      return

      else if (code.eq. 3)then

c     Initialise the input patterns to an array
      input(1,1,1) = stor/amult_in/1.e06
      input(1,1,2) = inflow/amult_in/1.e06
      input(1,1,3) = dem/amult_in/1.e06
      write(*,*)'stor inf dem values',(input(1,1,J),j=1,3)
c     starting matrix multiplication and sigmoid function loop
      do n=1,n_layer -1
      i=1
      do j=1,layers(n+1)
      transit(n,i,j)=0
      do k=1,layers(n)
      transit(n,i,j)=transit(n,i,j)+input(n,i,k)*weights(n,k,j)
      enddo
      input(n+1,i,j)=1./(1.+exp(-transit(n,i,j)))
      enddo
      enddo
      if(j.eq.2)then
      input(n_layer,1,1)=input(n_layer,1,1)*amult_out*1.e06
      release=input(n_layer,1,1)
      write(*,*)'output release is',release
      return
      else
      write(*,*)'value of j =',j
      write(*,*)'logical error ! Please check algorithm !! '
      return
      endif
      endif
      return
      end

      subroutine area_elev(storage1,a,el)
      common/elev/elev(20),area(20),stor(20),m,smax,smin
      if(storage1.gt.smax)then
      storage=smax
      else
      storage=storage1
      endif
      do i=1,m
c     write(*,*) storage,stor(1),stor(m)
      if(storage.lt.stor(1) .or. storage.gt.stor(m)) then
      write(*,22)
22    format(2x,'Storage goes beyond range  ERROR',/
     1 'PROGRAM TERMINATED')
      return
      else if (storage.gt.stor(i) .and. storage.lt.stor(i+1))then
      a=area(i)+((area(i+1)-area(i))/(stor(i+1)-stor(i)))*
     1 (storage-stor(i))
      el=elev(i)+((elev(i+1)-elev(i))/(stor(i+1)-stor(i)))*
     1 (storage-stor(i))
      return
      endif
      enddo
      end
```

<INPUT FILE> " SIMUL.IN" FOR THE SIMULATION MODEL

```
SIMULATION STUDIES OF INDRAVATI RESERVOIR

14 5
2300e+06 814.5e+06 1 2127.e+06
600.0 265.0
580      0.24e+06       0.00e+06
585      0.45e+06       1.70e+06
590      2.74e+06       11.28e+06
595      5.60e+06       40.00e+06
600      10.7e+06       74.16e+06
605      16.80e+06      140.00e+06
610      23.95e+06      237.69e+06
615      33.29e+06      380.29e+06
620      44.33e+06      570.54e+06
625      59.16e+06      827.15e+06
630      74.59e+06      1152.96e+06
635      89.97e+06      1562.60e+06
640      104.87e+06     2052.12e+06
645.5    122.80e+06     2710.00e+06

7.5 15 17.5 20.0 20.0 17.5 7.5 7.5 7.5 10.0 10.0 7.5

231.86 276.01 247.30 163.19 115.05 86.43 284.85 40.85 317.86 283.35 106.75 88.88
45.532799 31.449600 34.819199 36.288002 61.603203 59.616001 332.121613 278.553589 355.104004 149.990402
116.639999 77.673599
66.959999 58.060799 37.497601 44.063999 37.497601 72.576004 211.593597 348.192017 274.752014 155.347198
82.944000 42.854401
53.568001 60.480000 32.140800 20.736000 21.427200 334.368011 310.694397 741.916809 637.632019 125.884804
88.127998 69.638397
50.889603 72.575996 66.959999 64.800003 203.558411 300.671997 819.590393 723.168030 821.664001 425.865601
217.727997 101.779205
66.959999 74.995201 64.281601 51.840000 50.889603 31.104000 634.780823 1234.742432 1360.800049 1178.495972
225.503998 184.809601
```

THE OUTPUT FILE <SIMUL.OUT> FOR THE ANN OPTION IN A CURTAILED FORM.

```
SIMULATION STUDIES OF INDRAVATI RESERVOIR

        *** I N P U T   D A T A ***

  All Data are in MKS Units
 OPTION : ANN BASED MODEL

  The number of years for analysis :  5
  The computational time interval : one month

  Maximum storage capacity :  2300.000 million cum
  Minimum storage capacity :  814.500 million cum

 Installed Capacity : 600. MW
 Tail Water Level : 265.000 m

      Elevation - Area - CapacityTable

  S N   Elevation    Area       Capacity
        ( m )       (sqm)       (cum)

   1    580.000    240000.0        0.0
   2    585.000    450000.0    1700000.0
   3    590.000   2740000.0   11280000.0
   .
   .
        INTERMEDIATE LINES DELETED.....
   .
   .
```

14 645.500 122800000.0 2710000128.0

**** Results of Calculations ****

YYYY-MM		Init-Stor	Inflow	Demand	Release	fail	spil	av_H	Pow_G	Evap	Fin-Stor
		(Mil Cum)	(m cum)	(Mcm)	(m cum)		(m cum)	(m)	(MW)	(Mcm)	(Mcm)
1	1	2127.0	45.5	231.9	267.7	0	0.0	639.6	340.0	8.0	1896.8
1	2	1896.8	31.4	276.0	276.0	0	0.0	637.1	348.3	15.0	1637.3

.
.
.

INTERMEDIATE LINES DELETED.....

.
.
.
.

**** Results of Calculations****

YYYY-MM		Init-Stor	Inflow	Demand	Release	fail	spil	av_H	Pow_G	Evap	Fin-Stor
		(Mil Cum)	(m cum)	(Mcm)	(m cum)		(m cum)	(m)	(MW)	(Mcm)	(Mcm)
3	1	850.3	53.6	231.9	84.8	1	0.0	625.1	103.6	4.5	814.5
3	2	814.5	60.5	276.0	51.7	1	0.0	624.8	63.1	8.8	814.5
3	3	814.5	32.1	247.3	21.9	1	0.0	624.8	26.7	10.2	814.5
3	4	814.5	20.7	163.2	9.1	1	0.0	624.8	11.0	11.7	814.5
3	5	814.5	21.4	115.1	9.7	1	0.0	624.8	11.9	11.7	814.5
3	6	814.5	334.4	86.4	113.7	0	0.0	626.4	139.3	10.2	1025.0
3	7	1025.0	310.7	284.9	284.9	0	0.0	628.2	350.8	5.1	1045.7
3	8	1045.7	741.9	40.8	83.5	0	0.0	632.7	104.1	5.2	1698.9

.
.
.

INTERMEDIATE LINES DELETED....

.

5	10	2300.0	1178.5	283.4	469.2	0	698.1	642.1	600.0	11.2	2300.0
5	11	2300.0	225.5	106.8	214.3	0	0.0	642.1	274.1	11.2	2300.0
5	12	2300.0	184.8	88.9	176.4	0	0.0	642.1	225.6	8.4	2300.0

cumulative generation : 14952.63
cumulative deficit : 1395.91
cumulative spillage : 1852.08
total months model failed : 10