NONLINEAR MODEL PREDICTIVE CONTROL USING LEARNING MACHINES

Ph.D. THESIS

by

M. GERMIN NISHA

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE – 247 667 (INDIA)

MARCH, 2014

NONLINEAR MODEL PREDICTIVE CONTROL USING LEARNING MACHINES

A THESIS

Submitted in partial fulfilment of the

requirements for the award of the degree

of

DOCTOR OF PHILOSOPHY

in

ELECTRICAL ENGINEERING

by

M. GERMIN NISHA

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE – 247 667 (INDIA)

MARCH, 2014

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in this thesis entitled **"NONLINEAR MODEL PREDICTIVE CONTROL USING LEARNING MACHINES"** in partial fulfilment of the requirements for the award of the Degree of Doctor of Philosophy and submitted in the Department of Electrical Engineering of the Indian Institute of Technology Roorkee, Roorkee is an authentic record of my own work carried out during a period from December, 2010 to March, 2014 under the supervision of Dr. G. N. Pillai, Associate Professor, Department of Electrical Engineering, Indian Institute of Technology Roorkee, Roorkee.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other Institute.

(**M. GERMIN NISHA**)

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

(G. N. Pillai)
Supervisor

Date: _____

The Ph.D. Viva-Voce Examination of **Mrs. M. GERMIN NISHA**, Research Scholar, has been held on……..

Signature of Supervisor Chairman, SRC External Examiner

Head of the Department/Chairman, ODC

ABSTRACT

One among the advanced control algorithms that is receiving a great deal of concern in the process industries, chemical plant and oil refineries is predictive control. Predictive control is a prudent control which retains information about the past process variables and survey the current as well as the upcoming process variables. Internal model control, inferential control and model predictive control are some of the popular predictive controllers. Model predictive control (MPC), which works on the basis of receding horizon control has attracted the process control community due to its capability to handle constraints on process variables, nonlinearities and interactions among process variables, disturbances etc.

Model predictive controllers are model- based controllers which rely on dynamic models of the process. Process model plays a key role in model predictive controllers. The more accurate the model the more accurate is the controller. State space models, First Principle models, Hammerstein models, Volterra models etc were used widely to develop accurate dynamics of nonlinear processes which are effort demanding and time consuming. Then artificial neural network (ANN) models turned the attention of MPC users due to their ability to perfectly identify complex nonlinear relationships between dependent and independent variables with less effort. Several researchers have approximated nonlinear models by neural networks besides its lengthy training time, requirement of large training data, poor extrapolation, offset for multistep predictions in the presence of disturbances, over fitting with poor generalization etc. Another widely used machine learning technique introduced by Vapnik is deterministic sparse kernel technique named as support vector machine (SVM). Support vector regression (SVR) models are significant for its accuracy and sparse nature. Another existing machine learning technique introduced by Tipping is probabilistic sparse kernel learning technique called as relevance vector machines (RVM). Relevance vector regression (RVR) models which are much sparser reproduces the nonlinear dynamics accurately.

In this thesis, a novel neuro fuzzy technique, Extreme ANFIS is proposed and its significance in achieving accurate model is verified. Thus different machine learning techniques say ANN's, SVM's, RVM's and proposed novel neuro-fuzzy technique are employed to show their suitability to achieve accuracy and computational efficiency.

The other concern in model predictive controller is computational cost, as it does prediction and optimization at each sampling instant. This could be overcome by fast prediction and fast optimization techniques. Quasi-Newton methods are known for its fast

optimization as it skips Hessian matrix computations, but accuracy is less. Particle swarm optimization is an evolutionary algorithm which is meant for its success rate but has long processing time. Processing of conventional particle swarm optimization could be speeded up by particle swarm optimization with controllable random exploration velocity (PSO-CREV) technique. This PSO-CREV technique improves the intensity of exploration capability of conventional particle swarm optimization significantly by a time-varying bound of arbitrary search velocity to meet both the necessities of strong exploration skill and fast convergence with less number of iterations and less number of populations. In this thesis, PSO-CREV technique is adopted for its accuracy, simplicity and controllable fast computations.

A nonlinear model predictive control strategy which utilizes the above mentioned machine learning techniques and PSO-CREV optimization algorithm is applied to a single input single output (SISO) catalytic continuous stirred tank reactor (CSTR) process. An accurate reliable nonlinear model is first identified by the above mentioned machine learning techniques and then the optimization of control sequence is speeded up by PSO-CREV. An improved system performance is guaranteed by an accurate model and an efficient and fast optimization algorithm. Performance comparisons of MPC's using probabilistic sparse kernel learning technique called RVM's regression model, deterministic sparse kernel learning technique called Least squares support vector machines (LS-SVM) regression model, a proposed novel neuro-fuzzy based (Extreme adaptive neuro fuzzy inference system (ANFIS)) model and neural network based model are done on a CSTR process. Relevance vector regression model and Extreme ANFIS model shows good tracking performance with very less computation time which is much essential for real time control.

A nonlinear system with much faster dynamics is also considered for control. The control of photovoltaic (PV) array Maximum Power Point Tracker (MPPT) through Nonlinear Model Predictive Control (NMPC) strategy which uses Extreme ANFIS/ LS-SVM/ RVM regression model is proposed. Another Extreme ANFIS/ LS-SVM/ RVM model is employed to offer the reference Maximum Power Point (MPP) trajectory to the model predictive control system by predicting the maximum power point current and voltage of the nonlinear PV module at different operating conditions. The above control algorithm is speeded up by simplifying the optimization problem by Finite Control Set Model Predictive Control (FCS-MPC) technique. Thus an improved system performance is guaranteed by an accurate predictive model and simple control algorithm. The obtained

simulation results show the superiority of the proposed method compared to state space model based NMPC.

Control of highly nonlinear processes with interacting process variables is still a challenge in industries. Hence, a highly nonlinear binary distillation column process is considered for control to highlight the control accuracy and computational efficiency of NMPC strategy. An accurate reliable nonlinear model is first identified by the proposed novel neuro-fuzzy based (Extreme ANFIS) model and then the optimization of control sequence is speeded up by PSO-CREV. To compare the performance, MPC using probabilistic sparse kernel learning technique RVR with a RBF kernel, deterministic sparse kernel learning technique called LS-SVM regression model and ANN based model is done on a distillation column process. RVR based MPC and Extreme ANFIS based MPC again shows its significance in achieving good tracking performance with very less computational effort which is much essential for real time control applications.

Thus this thesis focused in incorporating accurate nonlinear model and reducing the computational cost related to nonlinear model predictive controller.

ACKNOWLEDGEMENT

I wish to affirm my life time gratitude and heartfelt thanks to my respected supervisor Dr. Gopinatha Pillai, Associate professor, Department of Electrical Engineering, IIT Roorkee for his proficient and sustained supervision, wide experience, exact suggestion which were the constant source of inspiration for the completion of this research work. I sincerely thank his humanistic and warm personal approach, which has led me to a smooth and steady research path. No words would suffice for his timely help and encouragements.

I am thankful to the Doctorial Research Committee chairman Dr. Barjeeve Tyagi, my internal subject expert Dr. Indra Gupta, Professor, Electrical Engineering department, IIT Roorkee, my external subject expert Dr. Sukavanam, Professor, Mathematics Department, IIT Roorkee for their valuable suggestions and encouragements.

I owe my special thanks to Dr. Promod Agarwal, Head Depatment of Electrical Engineering for providing excellent facilities and peaceful research environment for all students including me.

I am grateful to Prof. Vasantha for his encouragements with happy words, which helped me boot my morale.

I express my gratefulness to Dr. Vinod kumar, Dr. Pathey, Sr. Rajendra Prasad whose lectures gained me depth knowledge to complete my course work successfully. I also extend my thanks to Mr. C. M. Joshi and Mr. Rajendra Singh of Microprocessor lab for providing the laboratory comfortably accessible.

I wish to acknowledge the M.Tech students Mr. Alex, Mr. Varun Jain, Mr. Amit Kumar, Mr. Ankit Modi, Mr. Dipankar Battacharya, Mr. Pushpak Jagtap for spending their valuable time in constructive discussion.

I would like to thank my friends Dr. Srinivasan Alavandhar, Dr. Manju, Dr. Purushothaman, Dr. Thangaraj, Dr. Radha, Mr. Shajan, Mr. Ajay Shiv Sharma, Mr. Neeraj Gupta, Mr. Danie Roy, Mr. Franklin Frederick, Mr. Senthil, Mr. Ramesh, Mr. Prabu, Mr. Siva Chidambaram, Miss. Padma, Miss. Malarkodi, Mrs. Shermi for their helping hands in needs.

I have no words to express my thanks to my dear Loving husband Mr. M. John Robert Prince, Research Scholar, Civil department, IIT Roorkee, who stood behind me at every moment with care and affection. He has helped me in documentation also besides his research work. I feel Proud of my daughter J. Prisha and my son J. Jeffrey whose funny fights and laughter's provided me pleasant moments during the course of this work.

I gratefully acknowledge the moral support of my mother Mrs. I. Mary Retnam, my father Mr. V. Maria Francis, my brother Mr. M. Anish Nirmal, father- in -law Mr. M. Maria Innasi and mother -in -law Mrs. M. Gnana Selvam whose prayers and blessings led a smooth path throughout this research work. I am proud to humbly dedicate this research work to my Husband.

Above all I thank the Lord Almighty for his blessings and mercy to keep me healthy and make me able to complete this work.

Thank you all.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

Symbol	Description
A	State matrix
B	Input matrix
C	Output matrix
u	Manipulated variables
x	States of the process
y	Output of the process
Q	State weighting matrix
R	Input weighting matrix
$\hat{y}$	Predicted output
N_p	Prediction horizon
N_u	Control horizon
ref	Desired trajectory
k	Current sampling instant
e	Error
n_d	Delay node corresponding to plant input
d_d	Delay node corresponding to plant output
f_j	Activation function for the jth hidden neuron
b	Bias on the output node
b_j	Bias on the hidden nodes
hid	Number of hidden neurons
w_j	Weight between the jth hidden node and the output node
$w_{j,i}$	Weight connecting the ith input node with the jth hidden node
N_1	Minimum value of prediction horizon
N_2	Maximum value of prediction horizon
Δu	Control input change defined as u(n+j)-u(n+j-1)
λ	Control input weighting factor
d	Dimension of the solution space
M	Swarm size
r_1, r_2	Random numbers in the range [0,1]
c_1, c_2	Learning factors.
P_{id}	Best position recognized by particle I
P_{id}	Global best position recognized by particle i

M	Polynomial order
W	Weight vector
H	Hyperplane
γ	Regularization parameter
α_i	Lagrange multiplier
σ	Kernel width.
p(w|t)	Posterior probability
p(w)	Prior probability
p(t|w)	Likelihood function
σ^2	Variance
z	High dimensional feature space
φ	Basis function
A_i, B_i	Linguistic variables
a, b, c	Premise parameters
p, q, r,	Consequent parameters
η	Learning rate
$\xi_{id}^{(t)}$	Bounded random variable with continuous uniform distribution
$\overline{\xi}(n)$	Stochastic velocity with invariable value range and zero expectant
h	Liquid level in the reactor,
C_b	Concentration of product at the output of the process
C_{b1}	Concentration of the concentrated feed of the process
C_{b2}	Concentration of the diluted feed of the process
q_1	Concentrated feed flow rate Cb1
q_2	Diluted feed flow rate Cb2
q_0	Product flow rate at the output of the process
N	Total number of samples
I	Net output current of solar cell
I_{ph}	Photovoltaic current
I_{01},I_{02}	Diode reverse saturation currents
V	Terminal voltage
R_s	Series resistor
R_p	Parallel resistor
a_1,a_2	Diode constants
V_t	Thermal voltage

I_{SC}	Short-circuit current
V_{OC}	Open-circuit voltage
P	Power
i_{pv}	PV array current
v_c	Converter output voltage
v_{pv}	PV array voltage
J	Performance index
F	Feed rate
q_F	Fraction of liquid in feed
D and B	Distillate and bottom product flow rate
x_D	Distillate product composition
L	Reflux flow
x_B	Bottom product composition
V	Boilup flow
M_B	Liquid holdup on reboiler
M_D	Condenser holdup
M_i	Liquid holdup on theoretical tray *i*
N_F	Location of Feed tray from bottom
Q_F	Fraction liquid in feed
L_B	Liquid flow rate into reboiler
V_T	Vapour flow rate on top tray
X_B	Logarithmic bottom composition
Y_D	Logarithmic top composition
x_i	Liquid mole fraction of light component on stage i
y_i	Vapour mole fraction of light component on stage i
y_T	Vapour mole fraction of light component on top tray
Z_F	Mole fraction of light component in feed
x_D^{ref}	Desired value of distillate product composition
x_B^{ref}	Desired value of bottom product composition
N_T	Total number of trays

LIST OF ACRONYMS

LQR	Linear Quadratic Regulator
MPHC	Model Predictive Heuristic Control
MPC	Model Predictive Control
DMC	Dynamic Matrix Control
QDMC	Quadratic program Dynamic Matrix Control
IDCOM-M	Identification, configuration, simulation and control
SMOC	Shell Multivariable Optimizing Controller
DMC+	Dynamic matrix control plus
RMPC	Robust model predictive control technology
MIMO	Multi Input, Multi Output
PID	Proportional Integral Derivative
LMPC	Linear Model Predictive Control
NMPC	Nonlinear Model Predictive Control
ANN	Artificial Neural networks
FF	Feed-forward
RBF	Radial Basis Function
ERN	External Recurrent Networks
SVM	Support Vector Machines
SVR	Support Vector Regression
LS-SVM	Least squares support vector machines
RVM	Relevance vector machine
RVR	Relevance vector regression
SLFNs	Single-hidden layer feed-forward neural networks
ELM	Extreme learning machine
Extreme ANFIS	Extreme Adaptive neuro fuzzy inference system
FCS-MPC	Finite Control Set Model Predictive Control
CSTR	Catatytic Stirred Tank Reactor
ANFIS	Adaptive network based fuzzy inference system
HLA	Hybrid learning algorithm
BFGS	Broyden–Fletcher–Goldfarb–Shanno
PSO-CREV	Particle swarm optimization with controllable random exploration velocity
SISO	Single input single output
PV	Photovoltaic
MPPT	Maximum Power Point Tracker
PSO	Particle swarm optimization
CSA	Coupled Simulated Annealing
MLE	Maximum likelihood estimate

MAP	Maximum a posteriori
EM	Expectation Maximization
MHW	Moving horizon window
RMSE	Root mean square error
DC	Direct current
SMPS	Switched-mode power supply
IAE	Integral absolute error
LDA	Linear discriminant analysis
PCA	Principal components analysis
LSE	Least square error
BOS	Balance of system

Chapter 1
INTRODUCTION

This chapter describes the introduction to the research work. It starts with the basics of predictive control. Subsequently, the demanding tasks in nonlinear model predictive control are discussed with its solutions, which is the main subject of this thesis. Next, author's contribution and organisation of thesis are explained.

1.1 OVERVIEW

In the early 1960's, Kalman [1] developed the modern control concepts to determine an optimal linear control system by studying a Linear Quadratic Regulator (LQR) problem. The process to be controlled is described by a discrete time deterministic linear state space model as shown in equation (1.1).

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k \end{aligned} \tag{1.1}$$

The vectors A, B, C, *u*, *x* , *y* describes the state matrix, input matrix, output matrix, manipulated variables, states of the process, output of the process respectively. The quadratic objective function to be minimized is shown in equation (1.2) where Q is state weighting matrix and R is the input weighting matrix.

$$J = \sum_{J=1}^{\infty} \left[\left\| x_{k+\mathrm{j}} \right\|^2 Q + \left\| u_{k+\mathrm{j}} \right\|^2 R \right] \tag{1.2}$$

The solution to the above LQR problem resembles a propotional controller where the gain matrix K is the solution of matrix Ricatti equation (1.3)

$$u_k = -K x_k \tag{1.3}$$

The above algorithm was found to be stabilizable and detectable until the weight matrices Q and R are positive definite. Following this a dual control theory Linear Quadratic Gaussian controller came into practice which merges LQR and Kalman Filter [2]. But it was regarded as impractical due to its inability to meet constraints, process nonlinearities, and model uncertainty. This surroundings show the way for the intensification of model based control methodologies.

The exact origin of Model Predictive Control (MPC) took place in mid seventies to mid eighties. The first description on MPC, Model Predictive Heuristic Control (MPHC) was advocated via Richet et al., 1978 [3], followed by Cuttler and Ramaker 1980 [4] who developed Dynamic Matrix Control (DMC). These strategies made MPC more popular among industries. During the period of eighties to nineties, again new variants of MPC

techniques have been developed. These MPC algorithms include Quadratic program Dynamic Matrix Control (QDMC), identification, configuration, simulation and control (IDCOM-M), Shell Multivariable Optimizing Controller (SMOC), Dynamic matrix control plus (DMC+), Robust model predictive control technology (RMPC) etc. [5]. All the above linear MPC algorithms differ mainly in the process models used (step, impulse and state space) and their amendment to time varying models.

Model predictive control (MPC) is renowned as one of the advanced control technique found to be very successful in real time applications [5]. This acknowledgment is due to its ability to handle constraints forced on inputs and outputs of process, interactions between process variables, process nonlinearities, dead times, capability of controlling multi input, multi output (MIMO) nonlinear systems with signficant dead time and model uncertainties. The main advantage of MPC lies in its capability to optimize the current time slot, in relation with future timeslots.

Even though the proportional Integral derivative (PID) controller was widely used from 1890's, optimization of controller performance is beyond its scope. Currently model predictive control (MPC) has made the optimization and control of such complex problems more feasible. Incorporating accurate nonlinear model and fast online optimization techniques with less computational complexity is the main subject of this thesis.

1.2 RECEDING HORIZON CONTROL

It is the one which overcomes the drawbacks of fixed horizon control. A fixed horizon control leads to a control sequence which starts at some current time and concludes at some future time. Hence it leads to outdated fixed control choices and as the time given for objective function reduction is fixed the control may not be satisfactory. All the above draw backs could be overcome by receding horizon control principle.

In model predictive control, the prediction horizon keeps on moving forward and hence it is based on the principle of receding horizon control. The concept of receding horizon in MPC is shown in Fig.1.1. At current time instant K, the current plant state is sampled and a control strategy to minimize the difference between the desired set point trajectory and the process output predicted over the prediction horizon is calculated for a relatively short time horizon in future. Having calculated the future control strategy, only the initial step of the calculated control series is implemented to the actual process. At the subsequent sampling instant k+1, the output measurements necessary for prediction are restructured, the prediction horizon is moved one step ahead and the entire course of action

is repeated until the difference between the reference trajectory and the predicted plant's output are within the tolerable limit.

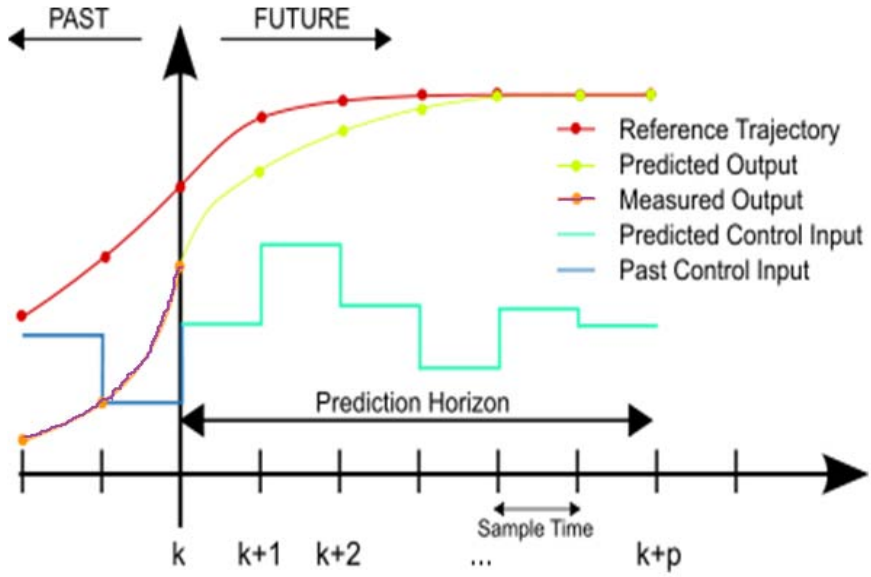

Fig. 1.1 The concept of receding horizon

1.3 LINEAR MODEL PREDICTIVE CONTROL

Linear MPC uses the linear dynamic model of the plant explicitly in order to control the upcoming plant behaviour. Thus plant model becomes the heart of MPC which is responsible to anticipate the effects of future manipulated variables, from the past history of measurements and control and to estimate the current state of the plant. The MPC algorithm determines an open loop series of manipulated variable at each control interval in order to optimize the upcoming plant behaviour online by incorporating constraints in the process variables. This major difference from other controllers makes MPC more attractive among industries.

The strategy of Linear MPC (LMPC) is demonstrated in Fig 1.2. It includes three important blocks, the actual plant to be controlled with output $y(k)$. The linear model of the actual plant has the predicted output $\hat{y}(k) = [\hat{y}(k+1)/k \ , \ldots \ \hat{y}(k+Np)/k]$ here, Np is the prediction horizon of MPC which dictates how far we wish the future to be predicted for. Next is the optimization block which provides the optimized control signal $u(k)=[u(k/k),\ldots u(k+ Nu -1 /k)]$ where Nu is the control horizon of MPC which dictates the number of control moves used to attain the upcoming control trajectory, subjected to the specified constraints that is required for the plant to achieve the desired trajectory $\text{ref}(k)=[ref1(k) \ldots . ref_{Np}(k)]$. Here k stands for the current sampling instant.

Thus at each sampling instant a sequence of manipulated variable $u(k)$ is calculated in order to lessen the difference involving the predicted output of the model and the desired

set point trajectory over the specified prediction horizon *Np*. The number of manipulated variable in the series is decided by the control horizon value *Nu* and only the first manipulated variable is implemented to the actual plant. This course is done in each sampling instant. Earlier linear MPC's were repeatedly used in practice which operates in a narrow operating region.

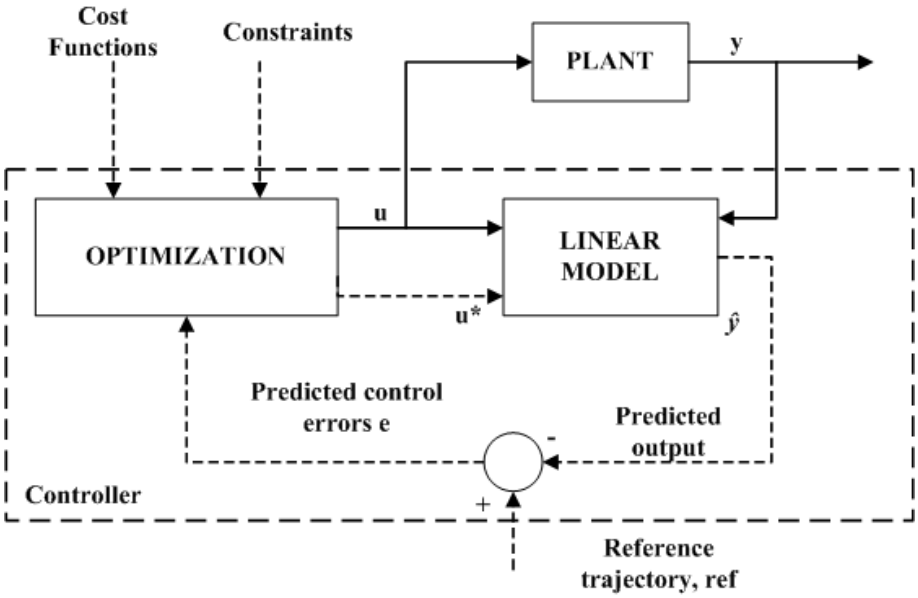

Fig. 1.2 The strategy of Linear MPC

But linear model predictive controllers fail to experiment the inevitable nonlinear behaviour of processes.

Linear MPC's are inadequate with the following limitations.

- Performances of Linear MPC'S are very poor for highly nonlinear processes operating over wide regions.
- They are not satisfactory when the process is subjected to strong regulator control problems and strong servo control problems.
- Linear MPC performance depends on the accuracy of the linear dynamic model.
- Most Linear MPC's uses dynamic step response model or impulse response model which has no validation technique to check if the collected data are sufficient to characterize the system dynamics.

These inadequacies coupled with increasingly stringent demands on throughput and product quality has encouraged the development of nonlinear model predictive control (NMPC) [6]. Hence the next generation MPC's started focussing on nonlinear models.

1.4 NONLINEAR MODEL PREDICTIVE CONTROL

Nonlinear MPC uses the nonlinear dynamic model of the plant explicitly to control the upcoming behaviour of the plant. Two challenging tasks in nonlinear model predictive

controller are acquiring an accurate nonlinear model and solving nonlinear optimization problem online. The strategy of nonlinear MPC is obtained by replacing linear model in Fig. 1.1 by nonlinear model.

The performance of nonlinear model predictive controller depends on model accuracy. For a highly tuned controller a very accurate model is necessary [7]. Thus precise nonlinear model is expected for better controlled performance.

1.4.1 Nonlinear system modeling

Artificial Neural networks (ANN) were widely believed for estimation of nonlinear system dynamics due to following reasons.

- Neuro-computing has its inbuilt ability to learn and approximate nonlinear functions.
- Online computation requirements are very less.
- Effortless configuration with less number of parameters
- Neural networks (NN) do not require an algorithm or rule development.
- Software development is relatively straight forward with just data file input output, Peripheral device interface, pre-processing and post processing.
- Peripheral device interface, pre-processing and post processing.

Most frequently used ANN topologies in nonlinear model predictive control techniques are, Feed-forward (FF) networks, Radial Basis Function (RBF), External Recurrent Networks (ERN).

Most of the researchers have adopted feed forward neural network for modeling due to its simple topology. Several scholars [8-10] have approximated nonlinear models by neural networks besides its following limitations.

- Lengthy training time.
- Requirement of large training data.
- Poor extrapolation.
- Offset for multistep predictions in the presence of disturbances.
- Over fitting with poor generalization etc.

Thus in spite of the existence of many nonlinear control approaches in theory, designing an appropriate controller for complex process is still confronted in practice [11].

Another learning method, sparse kernel learning is a nonlinear modeling method formerly projected in the machine learning area [12, 13]. Recently a novel kernel based deterministic nonlinear modeling method, Support Vector Machines (SVM) introduced by

Vapnik [14] has found its increasing applications in process modeling The guaranteed model accuracy, better generalization capability of SVR model are explicitly acknowledged by many researchers [15-20]. Thus by Support Vector Regression (SVR), the problem of over fitting can be avoided; generalization ability can be improved with better extrapolation capability with less number of training data and less training time. The complexity of developing an accurate model for highly nonlinear processes and the nonlinearities of its dynamics, make very attractive the use of SVM.

But, practical applications of SVMs are limited because of its requirement of larger number of kernels to approximate the optimal solutions. In least squares support vector machines (LS-SVM) the regularization parameter γ and the kernel width parameter σ are the two free parameters to be tuned to improve the generalization ability of predicted model. Thus the LS-SVM model is burdened with additional externally determined parameters, which is a time consuming task. Subsequently Tipping [21] introduced relevance vector machine (RVM) in 2000 which attracted much interest in the research community owing to its advantages over support vector machine. They are established on a Bayesian formulation which results in usage of less number of relevance vectors leading to much more sparse representation than support vector machine. Unlike in SVM framework where the basis functions must satisfy Mercer's kernel theorem, in the RVM case there is no restriction on the basis functions [14, 22]. Also, kernel width σ is the only parameter to be tuned in Relevance vector regression (RVR) model. Consequently the sparse RVR model could generalize better with very less computation time than SVM. The result given in [21] demonstrates the comparable generalization performance of RVM than SVM with intensely fewer kernel functions. Hence in this thesis the sparse kernel learning algorithms SVM and RVM are used for modeling of nonlinear complex processes to make use of the advantage of accurate prediction and sparse nature.

A novel machine learning algorithm which works on single-hidden layer feed-forward neural networks (SLFNs) is Extreme learning machine (ELM) [23].Conventional learning methods like neural networks and SVM's suffer drawbacks like slow learning speed and trivial learning variations for different applications of regression problems and two class and multi-class classification problems. Every parameter of conventional SLFNs requires tuning and thus there exists the reliance between parameters of hidden layer and output layer. Traditional learning techniques based on gradient descent methods are normally too slow due to inappropriate learning steps and have the problem of converging to local minima. The essence of ELM is that the hidden layer parameters cannot be

dependent of training samples, and these parameters need not be tuned. Recent research on ELM [24] has shown that ELM is better than neural networks because of faster learning speed and smaller generalization error.

A faster and novel neuro-fuzzy learning technique, Extreme Adaptive neuro-fuzzy inference system (Extreme ANFIS) is proposed in this thesis which combines the qualitative approach of fuzzy logic and adaptive capability of neural network. The structure of new extreme ANFIS algorithm is same as conventional ANFIS which works on hybrid learning algorithm. The significant features of proposed Extreme-ANFIS algorithm as compared to conventional hybrid learning algorithm are listed below:

- The time required to find gradient and to update premise parameters iteratively reduced in proposed Extreme ANFIS algorithm which resulted in significant reduction in overall learning time.
- The proposed algorithm is much simpler, faster and provides better generalization than conventional hybrid learning algorithm.
- The intuitive assumption of random premise parameters in the form of membership functions which are spread throughout the universe of discourse of input variable and local mapping ability of Sugeno type FIS in the form of rule base helps a lot in further improvement of learning speed.
- The proposed algorithm improves the flexibility of ANFIS architecture by eliminating differentiability constraint on membership function. In other words, the algorithm can also work with non differentiable membership functions. Also the different shapes of membership functions could be practiced easily within same universe of discourse of input.
- The reduction in iterative steps of learning allows ANFIS architecture to increase number of inputs and membership functions within required time constraints to improve accuracy while modeling the complex nonlinear systems.

1.4.2 Nonlinear optimization

In NMPC as the model is nonlinear the optimization problem is no longer quadratic and hence it imposes the optimization problem as nonlinear optimization problem. Hence, despite of accurate approximation of nonlinear dynamics of large dimension and highly nonlinear processes, it suffers from computational burden as model predictive controller does prediction and optimization at each sampling instant. The solution of such non convex optimization problem by evolutionary algorithms is very simple with less

computational complexity. Hence in this thesis a powerful evolutionary algorithm, particle swarm optimization with controllable random exploration velocity is applied for online optimization.

1.5 FINITE CONTROL SET NONLINEAR MODEL PREDICTIVE CONTROL

The Finite Control Set Model Predictive Control (FCS-MPC) is a method, in which the discrete character of power converters is utilized by which the optimization problem of MPC is omitted. In FCS-MPC method, by using a discrete model the performance of the system for every allowable actuation is predicted and the one which minimizes the predefined cost function is implemented for the subsequent sampling instant [25]. The key gain of FCS-MPC lies in the straight application of the control action to the power converter. The usage of prediction using RVR model or SVR model in FCS-MPC makes the controller faster to deal with real time applications with power converters.

1.6 LITERATURE REVIEW

The main stumbling block owing to the extension of LMPC to NMPC is the necessity of accurate nonlinear model and significant computation requisite during online optimization. Identification of a complex dynamic plant is a major concern in control theory [26]. This section presents a review of the most commonly used nonlinear process modeling techniques and online optimization techniques in NMPC. Widely used nonlinear process modeling techniques in NMPC are,

- First Principle model
- Empirical Models

1.6.1 First principle model

The first principle model otherwise called as white box model or mechanistic model is developed based on analysis of the system at fundamental level like the transient mass, energy and momentum relation to the system, state equations boundary conditions etc. Thus development of such fundamental dynamic models requires understanding of process fundamentals.

Patwardhan et al. [27] controlled a distillation column using NMPC at an operating condition where the process gain changed sign. The rigorous mechanistic model developed was not appropriate for use in on-line model predictive control calculations due to the memory limitations.

Chen et al. [28] described a novel NMPC scheme based on first principle model with guaranteed asymptotic closed loop stability. But this closed loop stability is proved

only under the assumption that no model/plant mismatch or disturbances are acting on the system and that the whole state vector can be measured.

Ricker et al. [29] developed a NMPC using the state variable formulation model with 26 states, 10 manipulated variables and 23 outputs. Developing such a model is difficult and which in turn makes the optimization problem of NMPC more complex.

Zheng et al. [30] proposed a novel model predictive control (MPC) algorithm for control of nonlinear multivariable systems, in which the author concluded that the number of manipulated variables directly affects the online computational demand of the algorithm.

Padtwardhan et.al [31] suggested that In NMPC, online solution to nonlinear programming will become simple if the model order is kept low. Hence, In order to keep the dimensionality of the nonlinear programming problem low the author has used Orthogonal Collocation method of order reduction to ease the computation.

Limitations of First Principle Model

- Development of first principles model is usually costly, time consuming and effort demanding.
- This method derives a model of very high order because of thorough modeling hence optimization problem in NMPC becomes a complex task [32].

1.6.2 Empirical models

The empirical model otherwise called as black box model is the one which relies exclusively on the system data obtained and necessitates no understanding of fundamental physical facts of the system. The data collected from the process should be able to extract the process characteristics accurately. The following are the commonly used empirical models in NMPC.

- Hammerstein model [33].
- Volterra model [34].
- Collocation model [35].
- Artificial Neural Networks [36].

1.6.2.1 Hammerstein model, Volterra model and Collocation model

Nonlinear model predictive controllers using Hammerstein model, Volterra model, Wiener model and Collocation model are discussed in many literatures. Fruzzetti et al. [33] developed a nonlinear Hammerstein model for a chemical plant. The Hammerstein model

developed is composed of a linear active element in succession to a nonlinear stationary element. The closed loop system performance on the developed Hammerstein model has shown significant improvement than the linear model even with constraints. Wiener model could be obtained by interchanging linear and nonlinear block. Maner et al. [34] describes the model predictive control using a Volterra model. In this paper, the author describes the capability of second-order Volterra models to hold asymmetric variation in output due to the symmetric variation in input. Jang et al. [35] explained NMPC based on collocation modeling.

Limitations

- Developing Volterra model for chemical plants from experimental data is practically due to the large requirements of data and experimentation time [37].
- Time required for Hammerstein model identification is more than parametric identification methods [38].

1.6.2.2 Neural networks for nonlinear modeling

The importance of nonlinear system identification and control using ANN has improved broadly during the past decade. A neural networks performance is highly dependent on its structure [39]. This is basically due to the verified superiority of various neural network architectures say Feed Forward neural network, Radial basis function (RBF) neural network, Elman neural network etc. in random non-linear mapping. Researchers [40, 41] have reviewed many papers on the process modeling and control using artificial neural networks which explores its significance. Arumugasamy et al. [42] discussed the better set point tracking and disturbance rejection of many neural network based NMPC with less computation expense. It also ensures that the neural network based NMPC works better in noisy environment when compared to the PI controllers. The satisfactory classification performance of neural networks using back propagation and radial basis function algorithm are discussed [43]. These significances turned the attention of MPC users to model their nonlinear systems using ANN. This section reviews a numbers of literatures on the integration of different neural network architectures into MPC schemes with their advantages and limitations.

(i) Feed forward Neural Network

Feed forward ANN allows signals to pass through in one direction without any feedback arrangement. The basic topology of feed forward neural network is shown in Fig.1.3.

Feed forward ANN are straight forward networks that correlate inputs with outputs and such an organisation is also referred to as bottom-up or top-down. They are extensively used in pattern recognition, system identification etc.

Georgieva et al. [44] controlled the feed flow rate of sugar syrup in a sugar crystallization process using a FF neural network based NMPC. Simulation results conveyed the smooth behaviour of the control actions and satisfactory set point tracking performance.

The review paper [45] expressed the versatility of Neural networks in its capability of being included in a variety of nonlinear control techniques and approaches, also the paper highlighted the sufficiency and capability of Multilayered feed forward neural network with sigmoid or hyperbolic transfer functions in most of the applications for performing systems identification and controls, besides the presence of several other kinds of topologies and activation functions.

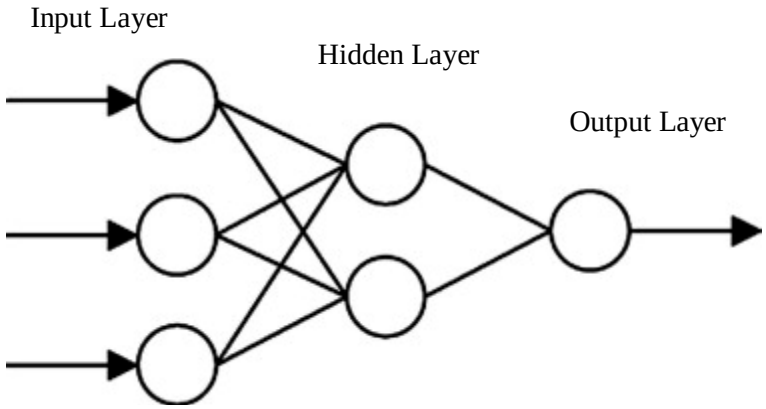

Fig. 1.3 Feed forward neural network

Gallaf et al. [46] modeled a nonlinear liquid level system with interaction using feed forward neural network and controlled using NMPC. Then the superior set point tracking performance of FF neural network based NMPC in comparison with conventional statistical identification method based NMPC is illustrated with necessary simulations.

Rankovic et al. [47] controlled a nonlinear system using a feed forward neural network based NMPC and digital recurrent network based NMPC. The FF neural network is trained using the standard back propagation algorithm and the digital recurrent network is trained using a dynamic back propagation algorithm. The simulation results showed the suitability of feed forward NN and digital recurrent network for the identification of complex nonlinear dynamics with satisfactory results.

Chidrawar et al. [48] controlled three different processes using multilayer feed forward network based NMPC. Simulation results demonstrate the consequence of neural networks on Generalized Predictive Control.

Kittisupakorn et al. [49] modeled a nonlinear steel pickling process by a multi-layer feed forward NN and incorporated it in model predictive control scheme to control hydrochloric acid concentrations. The neural network based NMPC developed gave better control performance with very less oscillations than the Proportional integral controller with less integral absolute error values even in the presence of disturbances.

Literature [42] evaluated multilayer feed forward neural network as the better network when compared to RBF and Elman NN. The better dynamic modeling capability of feed forward neural network made the performance of neural network based NMPC much better than linear MPC which were widely used in the MPC framework. It also guaranteed feasibility of the reference tracking in the presence of input constraints.

The simulation and experiment results showed the steady state offset of multistep model predictive controller using feed forward neural network models with longer prediction horizon in the presence of disturbance [50].

Advantages

- Less complicated network
- Better extrapolation property compared to Radial basis function and Elman neural networks [42]

Limitations

- Training time is lengthy than Radial basis function neural network [42].
- Multistep Model predictive control using feed forward neural network produces offset with the presence of disturbance [50].

(ii) Radial Basis Function network

Fig. 1.4 shows the basic structure of RBF network with p number of input layers with inputs $x_1....x_p$ respectively. The middle layer is the hidden layer with N neurons having radial basis function φ and weights $w_1... w_N$ associated with it. $F(x)$ is the output of RBF network.

The RBF, which was first introduced by Powell, was first used in neural network by Broomhead and Lowe. It is a type of neural network which is designed as a curve fitting problem in a high dimensional feature space. RBF network is trained in such a way to find a multidimensional function by measuring in a statistical sense that fits the training data as

best as possible, which is being measured in some statistical sense. Here the hidden unit forms a set of random basis function for the input vector which is called as the radial basis function.

Nonlinear model predictive controllers with radial basis functions as the plant model are discussed in [51]. Simulation results reveal the prospective ability of NMPC to model and control the catalytic stirred tank reactor (CSTR) process with better performance over the conventional proportional integral derivative controller.

Literature [52] presents a generalized MPC based on RBF neural network model for a class of time delayed nonlinear system. Initially RBF-NN modeling is done on the time delayed nonlinear system and then the developed model is embedded into predictive control algorithm. The RBF-NN can model nonlinear systems with unknown time delay. It can generate accurate control signals even in the presence of noise or fluctuation of the parameters of the plant. The efficacy of the proposed controller is confirmed in the simulation which explores the good adaptation and robustness capability of the network.

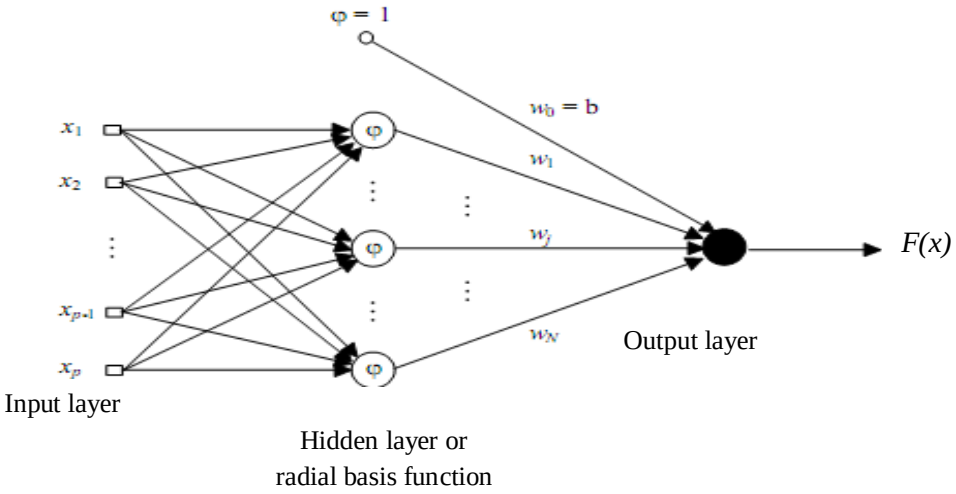

Fig. 1.4 Structure of RBF neural network

A radial basis function network using linear, cubic, thin-plate-spline, multiquadratic and inverse multiquadratic basis functions are presented [53]. The RBF network is trained to represent discrete-time nonlinear dynamic systems and the results are compared. The predictive accuracy of inverse multiquadratic function was relatively poor when compared to other basis functions. The results also indicate that the performance of the method depends very much on the different systems that were used.

Advantages

- Radial basis function has shortest training time [42].
- Radial basis function networks do not suffer from local minima problem [54].

Limitations

- Extrapolation far from training data is usually dangerous and unjustified in RBF networks [54].
- RBF are more sensitive to the curse of dimensionality [54].

(iii) Elman neural network

Elman Neural Network, a semi-recursive neural network using the Back propagation algorithm is a special case of architecture employed by Jeff Elman. These neural algorithms have a drawback of slow convergence rate [55]. Fig. 1.5 shows an Elman network which is a simple recurrent NN with input layer *x*, hidden layer *y*, output layer *z* and context unit *u*. The inputs to the network are $in_1....in_k$. The corresponding weights between input layer and hidden layer *are* $w_{x1y1}....\ w_{xkyl}$ and the corresponding weights between the hidden layer and output layer are $w_{y1z1}....\ w_{ylzn}$. The hidden layer *y* is connected to the context unit *u* with a fixed weight of one. At each time step, the input is transmitted in a standard feed-forward fashion, and then the back propagation learning rule is applied.

The permanent back connections make the context units always to maintain a duplicate of the earlier values of the hidden units. Thus the response of the Elman network depends both on current and past network inputs. Hence Elman network could perform sequence-prediction which is beyond the scope of a standard multilayer perceptron.

In [56] an Elman ANN model is used to replicate the dynamics of imperfectly mixed bioreactor and a feed-forward ANN is used for control purpose. They investigate the appropriateness of an ANN for optimization and control.

Declerq et al. [57] analyzed all the three different neural networks feed forward NN, the RBF based NN and the Elman neural network in designing a predictor for a dynamic non-linear system. Those neural network predictors were then used in MPC control algorithm. They validated the models using time validation and found that the feed forward neural net predicted the underlying non linearity of the system repeatedly better than the other two networks.

Limitations

- Training time is lengthy when compared to Radial Basis Functions [42].
- Poor extrapolation property when compared to feed forward neural network [42].

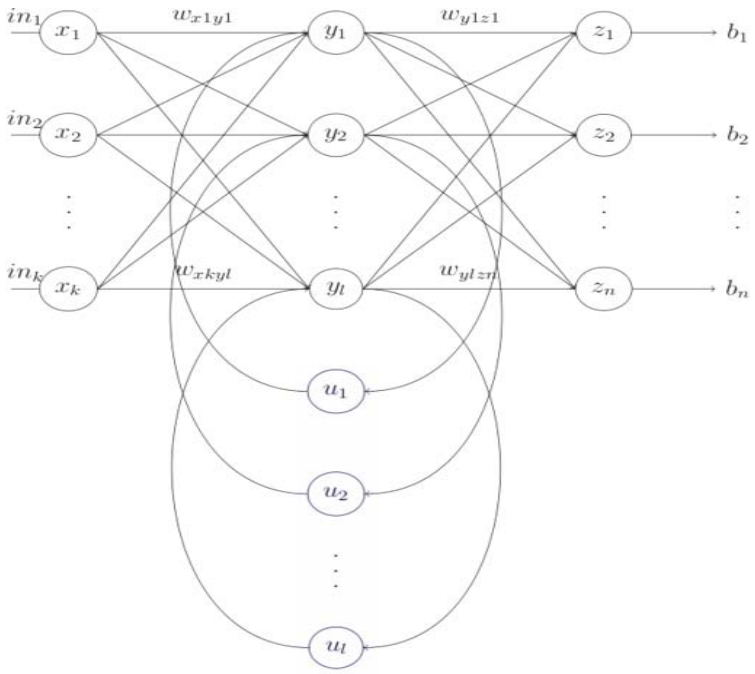

Fig. 1.5 Elman neural network

1.6.3 Support Vector Machines for nonlinear modeling

The novel SVM algorithm was invented by Vladimir Vapnik in 1995 [58]. A support vector machine is a new concept for analysis of data in a set of associated supervised machine learning methods, recognizing patterns, classification and function approximation analysis. Support Vector learning depends on effortless ideas which initiated in statistical learning theory [14]. They are fast replacing neural network as the tool of choice for classification and regression tasks, primarily due to their ability to generalize well on unseen data. SVM's are characterized by usage of kernels, nonexistence of local minima and sparse solution with less number of support vectors. Although SVM's are being used mainly for classification tasks, in recent times SVM's have been fruitfully applied to solve regression problems [59]. The guaranteed model accuracy and better generalization capability of SVR models are explicitly acknowledged by many researchers [15- 20].

- The neural network model could not perform better outside the training data range, also the NN model over fits data with poor generalization. The SVR model extrapolates much better than that of the NN model with good generalization also, the SVR modeling method can be implemented without difficulty. Therefore, SVR is a more suitable when compared to other data-driven empirical modeling methods [11**].**

- An upright SVR model can be attained using less training data than NN methods [60].
- One of the specific advantages of SVM is sparseness of the solution. Which means SVM's solution depends on the support vectors and not on the whole data set [61]. SVM's are successfully applied in identifying faults in induction motors [62].
- The three significant controller performance characteristics of robustness, set-point tracking, and also ensuring the preferred feature in the presence of unmeasured disturbances makes SVR a substitute control method for nonlinear chemical processes [11].

Least squares support vector machines, first proposed by Suykens and Vandewalle are least square versions of support vector machines, In LS-SVM version, the solution is made simple by resolving a set of linear equations as an alternative to a convex quadratic programming problem for conventional SVMs.

1.6.4 Relevance Vector Machine

The research community is attracted towards Relevance vector machines for classification and regression due to its advantages. RVM are based on a Bayesian formulation of a linear model, which makes its representation sparser. This sparseness nature makes RVM to generalize better and provide outputs at low computational cost [63]. Nonlinear system identification using RVM is successfully discussed in many literatures [64- 69] which highlights its significance.

- The number of relevance vectors is very less than that of support vectors which makes the RVM approximated function significantly sparser than the SVM decision function [63].
- Modification of regularization parameter is not needed in RVM as in SVM training [63].
- The algorithm for RVM seems to be simple with the requirement of more memory and computations [22].

1.6.5 Neuro-Fuzzy techniques

Neuro-fuzzy techniques have captivated increasing attention of researchers in different areas due to the growing requirement of intelligent systems [70]. Satisfactory performance of Fuzzy logic based controller is discussed in [71, 72].

Fuzzy logic systems do not incorporate any learning, while neural networks, a black box approach, do not possess mechanisms for explicit knowledge representation. By

combining neural networks and fuzzy logic, advantages of both these approaches can be incorporated in neuro-fuzzy systems. Adaptive network based fuzzy inference system (ANFIS) is a hybrid intelligent system which merges the fuzzy logic's qualitative approach and neural network's adaptive capabilities towards better performance [73]. A hybrid learning algorithm is used by ANFIS to identify parameters of Sugeno-type [74] fuzzy inference systems. Many applications of ANFIS are reported in literature [75]. ANFIS has strong computational complexity restrictions because of hybrid learning algorithm (HLA). A faster learning technique 'Extreme ANFIS' is proposed in this thesis. Extreme learning machine [23] is a novel algorithm which works on single-hidden layer feed-forward neural networks. Extreme ANFIS network reduces the computation complexity of the ANFIS by eliminating the hybrid learning algorithm and avoids the randomness of the ELM networks by incorporating explicit knowledge representation using fuzzy membership functions.

1.6.6 Online Nonlinear Optimization Techniques

Finding feasible solution for a constrained nonlinear equation is a very challenging problem. Solution for such complex nonlinear system requires high computational efforts [76]. MPC is forced to do optimization at each sampling instant online. Hence computational complexity and guarantee for convergence are the two important tasks to be considered while selecting the optimization techniques.

1.6.6.1 Nonlinear Programming

Nonlinear programming is the process of cracking a method of collectively termed constraints both with equalities and inequalities, over a set of mysterious real variables, along with a cost function to be maximized or minimized, where few constraints or the cost function are nonlinear. Different methods of nonlinear programming applied for nonlinear optimization in MPC are Newton-Raphson method [77], Levenberg Marquardt [78]. In the above techniques computation of Jacobian and hessian are unavoidable which increases the computational cost.

The online optimization in MPC at each sampling instant is carried out by Newton-Raphson method in [77]. It involves the calculation of Hessian but as the number of iterations required for convergence is less real time applications become feasible.

In [78] the optimization performances of Levenberg Marquardt method of optimization and Newton-Raphson method of optimization in MPC algorithm are compared and concluded that Levenberg Marquardt method performs better with minimum oscillations around the set points with stability.

Neural network based MPC using Quasi Newton method of online optimization is discussed in [79]. The Newton's method approximated by Broyden–Fletcher–Goldfarb–Shanno (BFGS) method is called as Quasi Newton's method of optimization. Here the calculation of Hessian is omitted making the algorithm simpler with fewer computations. But the accuracy of Quasi Newton method is less due to approximations. Standard direct methods, such as Newton's method, are impractical for large-scale problems because of their high linear algebra costs and large memory requirements [80]. Xue Cheng Xi et al. [81] obtained the control sequence by dynamic programming in which selection of sub problems and ordering are tough tasks.

1.6.6.2 Evolutionary algorithms

One of the promising research fields is evolutionary techniques which utilize nature or social systems. It became popular because of its ability and versatility to optimize complex problems [82]. The solution of non convex nonlinear optimization problem by evolutionary algorithms is simple since it is a derivative free method. The computational complexity in evolutionary algorithm is a prohibiting factor. Chen Yue-hua et al. [83] has optimized the performance index by genetic algorithm which has more computational effort when compared with particle swarm optimization.

Emad et al. [84] has compared five evolutionary-based search methods in his paper. Genetic algorithm, Memetic algorithm, Particle swarm optimization, Ant colony optimization, Shuffled frog leaping optimization algorithms were considered and their comparative results were presented. The particle swarm optimization (PSO) method dominates other algorithms both in success rate and solution quality but only second best in time consumption.

Particle swarm optimization method confirmed its capability to treat very difficult optimization and search problems [85]. This algorithm is an attractive tool due to its simplicity and high performance, it has been proven to be a powerful challenger to other evolutionary algorithms [86, 87] and been extensively used in numerous optimization processes [88, 89]. The performance of PSO-based feedback controller is robust and optimal [90]. It is a computationally efficient method since it is a derivative free method. Xin Chen et al. [91, 92] has authenticated a novel method of optimization, particle swarm optimization with controllable random exploration velocity (PSO-CREV), for its computational efficiency and improved performance than conventional particle swarm optimization with guaranteed convergence.

1.6.7 Finite Control Set Model Predictive control

The Finite Control Set Model Predictive Control is a method, in which the optimization problem can be simplified by utilizing the discontinuous nature of power converters. In this technique the system behaviour is predicted only for those possible on/off switching states and the one with minimum error is implemented. Many successful applications of FCS-MPC method for power converters and drives are discussed in articles [93- 100].

The FCS-MPC of PV system for maximum power point tracking using state space model is discussed in [101]. The faster dynamics of PV system and power converters are administered successfully by FCS-MPC principle since the burden of online optimization is omitted in FCS-MPC principle.

1.7 AUTHOR'S CONTRIBUTIONS

The author's contributions in the area of research are summarized as follows-

1. This research focused in achieving accurate nonlinear models by adopting different machine learning techniques say; Feed forward neural network, Support vector machines, Relevance vector machines and novel Extreme adaptive neuro-fuzzy inference systems. Their suitability in achieving accurate predictions is verified.
2. The fast accurate convergence of performance function in MPC is a great challenge in practice. This is made simpler by adopting PSO-CREV optimization technique which optimizes the performance function accurately with very less computational time.
3. Nonlinear model predictive control of system with faster dynamics is made feasible by incorporating FCS-MPC principle. The control of photovoltaic array Maximum Power Point Tracker through Nonlinear Model Predictive Control strategy using fast predicting model and FCS-MPC technique is successfully done.
4. The Feed forward neural network based NMPC is demonstrated for a nonlinear system. The nonlinear process considered for study is the Duffing's equation. The online nonlinear optimization involved in NMPC is performed using PSO. The performance of the above developed NMPC is compared with Linear MPC. The simulation results conveys the significance of NMPC
5. Different empirical models are developed for a single input single output (SISO) process. Catalytic stirred tank reactor is the SISO process considered for analysis. The performances of corresponding model based NMPC's are compared. The developed Extreme ANFIS PSO-CREV-MPC, SVM-PSO-CREV-MPC and RVM-

PSO-CREV-MPC for SISO process are compared with NN based MPC to show their significance.

6. Then the control of system with faster dynamics using model predictive control is focused. A photovoltaic (PV) array with power converter is considered as the fast dynamic system. The control of photovoltaic array Maximum Power Point Tracker (MPPT) is done by Finite Control Set Nonlinear Model Predictive Control strategy by utilizing Extreme ANFIS/ SVM/ RVM regression models. The accurate prediction nature of Extreme ANFIS/ SVM/ RVM model made the performance better.
7. Different modeling techniques are incorporated to develop NMPC for a Multi Input Multi Output process. Binary distillation column is the MIMO process considered for analysis. The developed Extreme ANFIS-PSO-CREV-MPC, LS-SVM-PSO-CREV-MPC and RVM-PSO-CREV-MPC for MIMO distillation column processes are compared with NN based MPC to show their significance.

1.8 ORGANISATION OF THE THESIS

This thesis is organized into seven chapters. A brief summary of each chapter is given below.

Chapter 1 describes the history of model predictive control. Then the demanding tasks in nonlinear model predictive control are discussed with its solutions, which is the main subject of this thesis. The author's contribution and organization of thesis are also explained in this chapter.

In Chapter 2, Neural Network (NN) based model predictive control is explained in detail. The Feed forward neural network based NMPC with particle swarm optimization is developed for a well known nonlinear system, Duffing's equation. The performance of the above developed NMPC is compared with Linear MPC with necessary simulations.

In Chapter 3, the basics of SVM based nonlinear model predictive control, RVM based nonlinear model predictive control and the proposed neuro-fuzzy techniques are discussed with their advantages.

Chapter 4 analyzes the importance of Relevance Vector Machine based NMPC, Support Vector Machine based NMPC and Extreme ANFIS based NMPC in comparison with Neural Network based NMPC by simulating a SISO CSTR process. A Fast and guaranteed optimization algorithm PSO-CREV is used for online optimization which minimizes the complexity in computation.

Chapter 5 describes the model predictive control of a system with faster dynamics. The study system is a photovoltaic array with power converter. The importance of RVM/ SVM model in developing MPP tracker, and the control of entire photovoltaic system by Finite control set model predictive control to maintain the maximum power are validated by simulation results.

Chapter 6 analyzes a highly nonlinear, interacting MIMO distillation column process whose simulation results convey the importance of Relevance Vector Machine based NMPC, Support Vector Machine based NMPC and Extreme ANFIS based NMPC in comparison with Neural Network based NMPC. A fast and guaranteed optimization algorithm PSO-CREV is used for online optimization which minimizes the complexity in computation.

Chapter 7 concludes the thesis by highlighting the significant contribution of the thesis with the scope for further research in the area of NMPC.

Chapter 2
MODEL PREDICTIVE CONTROL USING NEURAL NETWORKS

This chapter describes model predictive control using neural networks. It starts with some background on neural networks capability in nonlinear system modeling. Then, the theory behind the basic principle of neural network based MPC is discussed more in detail. The performance comparisons of linear MPC, nonlinear MPC, NN based MPC using line search method of optimization and NN based MPC using particle swarm optimization are shown.

2.1 INTRODUCTION

A common control loop feedback mechanism, PID controller is extensively used in industrial control systems from 1890's. But controlling systems with higher order dynamics and large time delays and constraints are beyond its scope. In some cases, complex constrained formulations are transformed to simple unconstrained optimization formulations after removing the set of constraints and inserting them into the cost function [102]. Optimization and control of such dynamic processes became viable after the surfacing of Model Predictive controller in 1980's. MPC technology is found to be applied in broad areas including chemicals, food processing, automotive, and aerospace applications [5]. Original Model predictive controllers were developed using linear models and hence the optimization problems are convex which could be solved analytically. But due to the severe nonlinearity of several chemical and industrial processes incorporation of nonlinear model is inescapable for better controlled performance. Therefore, a modeling technique with the ability to learn and represent the nonlinear behaviour is needed [6].

The modern era of artificial neural network started with the revolutionary work of McCulloch and Pitts in the year 1943. Artificial Neural networks purge the need to develop an explicit model of a process by developing model even for unidentified parts of the process with noise. Neural networks have the better capability to virtually map any sets of data. System identification using neural networks have been demonstrated in many published works with their promising results [8- 10], [103- 108].

After reviewing 100 relevant papers, Mohamed Azlan Hussain [45] has concluded that multilayer feed forward with sigmoid or hyperbolic activation functions is commonly used in applications than other network topologies and activation functions. This proves their adequacy and potentiality for performing process identification and controls for ample range of problems. The prediction accuracy of Feed forward neural networks is much better than external recurrent network but, they produce offset for multistep ahead prediction in the presence of disturbance. The external recurrent neural network produces

zero offset in the presence of disturbance [50]. But training phase of these recurrent neural networks is not easy for large number of inputs.

System modeling using dynamic recurrent neural networks is discussed in many literatures [109-112]. Artificial neural network has its unique advantage in the area of incipient faults detection [113]. Errors in neural network output depends on the architecture also hence selection of appropriate neural network plays vital role before modeling.

The feed forward neural networks are generally used for system identification in process control due to following significance. It has a fixed computation time and reasonable computation speed as a result of the parallel structure [114]. It can learn from noisy and incomplete data [115, 116]. FF neural network could generalize better than RBF and recurrent networks to circumstances not taught to network previously but the output of neural networks is usually associated with small error, since it finds a general approximation to a solution.

When incorporating the nonlinear neural network model in MPC, the optimization problem of Neural network based NMPC becomes non convex which could not be solved analytically. The selection of minimization algorithm plays an important role in reducing computational complexity of nonlinear MPC's. In MPC, as the future control signal is calculated based on online optimization at each sampling instant, it is burdened with the calculation of Jacobean and Hessian in nonlinear programming. This computation cost disables the nonlinear model predictive controller from making response in time [91]. As model predictive controller does prediction and optimization at each sampling instant, it is computationally expensive. Hence selecting a suitable optimization algorithm is another criterion.

Some researchers have used nonlinear programming methods based on approximated Hessian. In [117] the most popular formula known as Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm to approximate the inverse Hessian is used. Quasi Newton is easier to implement as exact Hessian is not required and it further ensures fast convergence. But the set point tracking performance is not remarkable.

Introduction of this chapter is followed by the theory behind the basic principle of neural network based MPC. Then simulations are carried out to compare the performances of linear MPC, nonlinear MPC, NN based MPC using line search method of optimization and NN based MPC using particle swarm optimization.

2.2 PRINCIPLE OF NEURAL NETWORK BASED MPC

The basic arrangement of NN based predictive controller is shown in Fig. 2.1. It contains three important components, the process under control with output *y*, a NN that approximates the process with predicted output $\hat{y}$ and the optimization algorithm which is framed based on cost function and constraints which determines the input *u* required for the plant to achieve the desired performance. Reference trajectory is represented by *ref*. The general principle of NN based MPC is also explained by the flow diagram shown in Fig. 2.2.

At every sampling instant a series of control input is calculated based on past process measurements. Number of control inputs in the sequence is decided by the control horizon N_u which dictates the number of control moves used to attain the future control trajectory. The control inputs are calculated in such a way as to minimize the performance function, which is the difference between models predicted output and the desired set point trajectory over the specified prediction horizon N_p which dictates how far we wish the future to be predicted for, and only the initial control input in the series is used by the actual plant. This procedure is repeated at every sampling instant with new prediction of the same length.

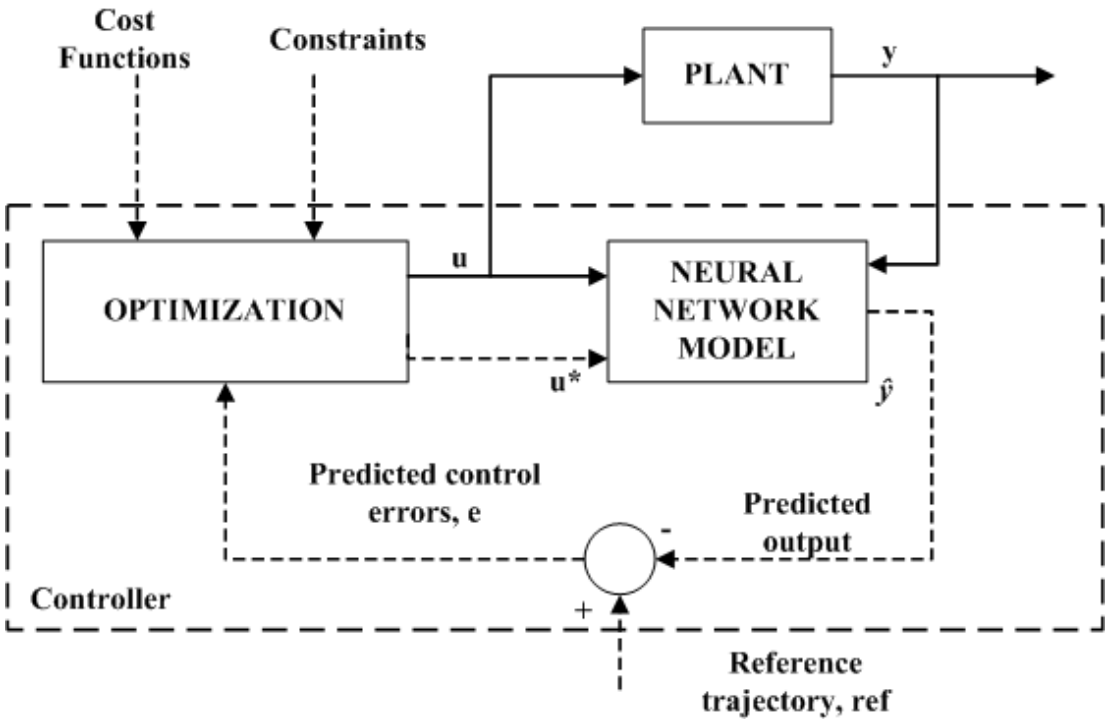

Fig. 2.1 Basic Structure of neural network based model predictive controller

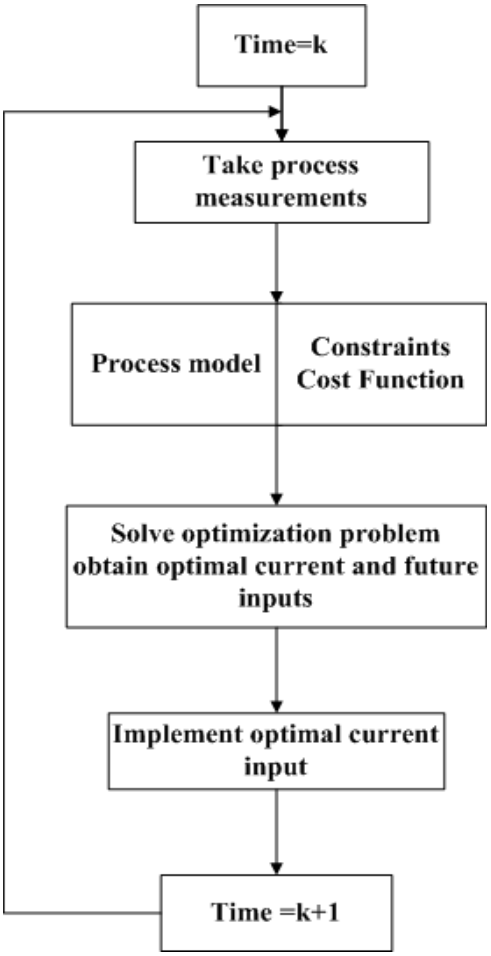

Fig. 2.2 Flow chart of MPC algorithm

2.3 TRAINING OF NEURAL NETWORK

The offline training of neural network model before embedding in MPC is shown in Fig. 2.3. The experimental input, output data is collected from the plant to be modeled. This input, output data should articulate the dynamics of the plant to be controlled. Same input *u(k)* is provided to both the plant and the NN model. The neural network has another input which is responsible for capturing the dynamics of the plant and maintaining system stability. This input either comes from the actual plant's output, *y(k)*, or the NN's output, $\hat{y}(k)$.

Training a neural network is nothing but adjusting the weights associated with each input until the desired output is reached. The difference between the NN's output signal, $\hat{y}(k)$, and the plant's output signal, *y(k)* is calculated and is then used to revise the weights of the network after passing through appropriate learning function. This procedure is recurrently done until the deviation of the output of NN model from the actual process is reduced to allowable level.

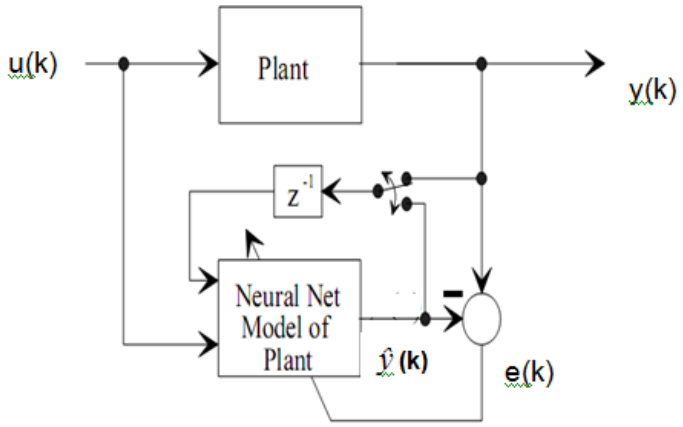

Fig. 2.3 Neural network training

Many different training functions are available in Matlab neural network toolbox, for updating weights in neural networks. They are Batch training with weight and bias learning rules, BFGS quasi-Newton backpropagation, BFGS quasi-Newton backpropagation for use with NN model reference adaptive controller, Bayesian regularization, Batch unsupervised weight/bias training, Cyclical order incremental update, Powell-Beale conjugate backpropagation, Fletcher-Powell conjugate gradient backpropagation, Polak-Ribiére conjugate gradient backpropagation, Gradient descent backpropagation, Gradient descent with adaptive learning rule backpropagation, Gradient descent with momentum backpropagation, Gradient descent with momentum and adaptive learning rule backpropagation, Levenberg- Marquardt backpropagation, One step secant backpropagation,Random order incremental training with learning functions, Resilient backpropagation, Sequential order incremental training with learning functions, Scaled conjugate gradient backpropagation. Among them Levenberg- Marquardt minimization algorithm is the better one which converges faster with robustness [118- 120].

Before using the trained network for prediction, validation of the network is necessary to check over fitting and generalization. In this test, an unseen input signal which is beyond the input data provided for training the ANN is used to compare the non-linear plant output and the trained ANN output. Hence, it is a time validation test with an error index equation (2.1) to quantify it and to check the accuracy of the approximated model with N number of validation test samples.

$$error_{index} = \sqrt{\frac{\sum^{N}(\hat{y}(k) - y(k))^2}{\sum_{k=1}^{N} y(k)^2}} \tag{2.1}$$

2.4 PREDICTION USING FEED FORWARD NEURAL NETWORK

A model is developed to generate system predictions. The neural networks describe the nonlinear models explicitly by their powerful function approximation properties [121]. Controllers based on neural network models offers an attractive property of robustness due to necessary information contained in the model [42].

One step ahead prediction of a Multilayer feed forward neural network structure comprising of five hidden neurons with sigmoid activation function and a single output neuron with linear activation function is shown in Fig. 2.4. The input's for this network are *u(k+1),y(k)* and their analogous delay nodes *u(k),…..u(k-n_d)* and *y(k) … y(k-d_d)* where *k* is the current sampling instant. The number of delay nodes corresponding to input, *u* and output, *y* is determined by n_d and d_d respectively. The output of the NN $\hat{y}(k+1)$ is the first predicted output. The network has one hidden layer were the number of hidden neurons and the activation function associated with each is selected by the user based on the plant under control.

Multistep ahead prediction is done by repeating the single step ahead prediction recursively by feeding back the predicted output at each step, as input to the neural network.

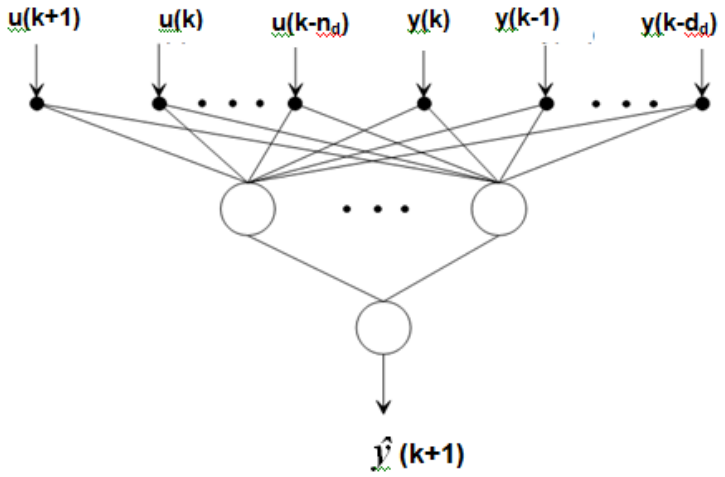

Fig. 2.4 Multilayer feed forward neural network for single step ahead prediction

The equation for n step ahead prediction is given below,

$$\hat{y}(k+n) = \sum_{j=1}^{hid} \{w_j f_j(net_j(k+n))\} + b \qquad (2.2)$$

Where,

$$net_j(k+n) = \sum_{i=0}^{n_d} w_{j,i+1} \begin{Bmatrix} u(k+n-i) & ,n-N_u < i \\ u(k+N_u) & ,n-N_u \geq i \end{Bmatrix}$$

$$+ \sum_{i=1}^{\min(n,d_d)} \left(w_{j,n_d+i+1} \hat{y}(k+n-i) \right) + \sum_{i=n+1}^{d_d} \left(w_{j,n_d+i+1} y(k+n-i) \right) + b_j \qquad (2.3)$$

k - current sampling instant,

$\hat{y}(k+n)$ - n step ahead predicted output of neural network at k^{th} sampling instant,

f_j - activation function for the j^{th} hidden neuron

b, b_j - bias on the output and hidden nodes respectively,

hid - number of hidden neurons,

w_j - weight between the j^{th} hidden node and the output node,

$w_{j,I}$ - weight connecting the i^{th} input node with the j^{th} hidden node

The first term in equation (2.3) stands for the values of control input u used in Fig. 2.4, in which the condition $k\text{-}N_u<i$ holds the previous future values of u and the other condition set all the other control inputs to $u(k+N_u)$. The second summation of equation (2.3) stands for the feedback part which is fed back after each step prediction. The very last summation stands for the past values of y.

A two step ahead prediction of a plant with 2 delay nodes corresponding with input u and 3 delay nodes corresponding with plant output y is illustrated in Fig. 2.5.

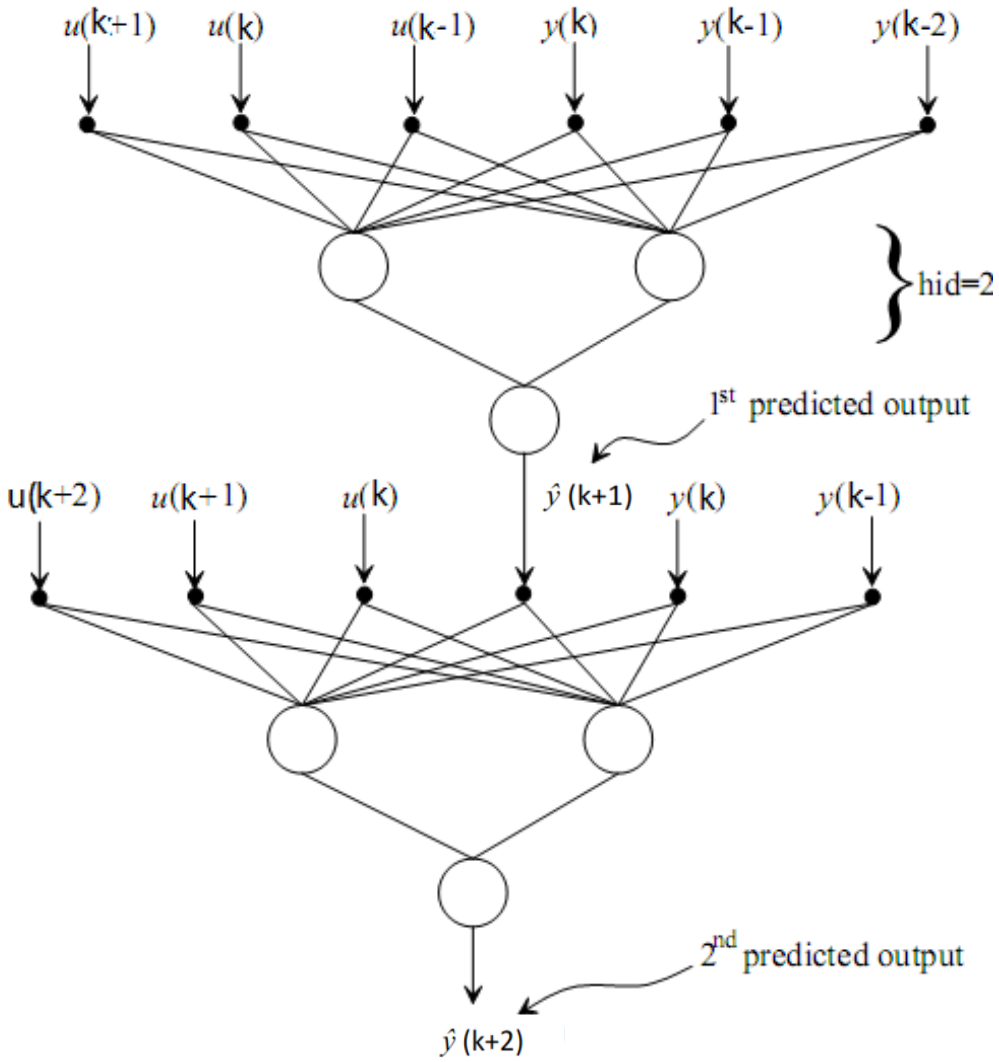

Fig. 2.5 Two step ahead prediction using feed forward neural network

2.5 COST FUNCTION FORMULATION

In the MPC dynamic optimization problem the cost function to be minimized over the prediction horizon at every sampling instant is formulated as in equation (2.4)

$$J = \sum_{j=N_1}^{N_2} [ref(k+j) - \hat{y}(k+j)]^2 + \sum_{j=1}^{N_u} \lambda [\Delta u(k+j)]^2 \qquad (2.4)$$

N_1 - minimum value of prediction horizon

N_2 - maximum value of prediction horizon

N_u - control horizon

ref - reference trajectory

$\hat{y}$ - output predicted by the neural network

Δu - control input change defined as *u(n+j)-u(n+j-1)*

λ - Control input weighting factor.

The first term in the cost function above minimizes the mean square error between the estimated output of the plant's model and set point signal and the second term minimizes the magnitude of change in the control signal used by the plant. As the value of λ increases the change in control signal will be smooth and slow.

In Model predictive controller this cost function is minimized at each sampling instant and the control signal within a specified constraint is generated. This control signal will let the plant's output to track the desired reference trajectory. N_2, N_u and λ are the tuning parameters in the cost function for better minimization. This cost function formulated above could be minimized by either neural network function approximator or nonlinear programming or any evolutionary algorithms which are derivative free.

2.6 MINIMIZATION OF PERFORMANCE FUNCTION BY NEURAL NETWORK

Akesson et al. [121] has minimized the cost function of the neural network based NMPC by another neural network approximator. The corresponding neural network model is trained offline so that the future cost over a prediction horizon is reduced. Here the online computational burden at each sampling instant is minimized; since part of the computation is done offline while training the neural network and making it ready for approximation. But the computational complexity to create the training data is more tiresome as every training data point involves solving the MPC optimization problem. Another drawback is that for some choice of cost functions, minimization may not be accurate. Hopfield neural network based optimization in grid applications is discussed in [122].

2.7 PERFORMANCE COMPARISON OF LINEAR MPC AND NEURAL NETWORK BASED MPC

A nonlinear system does not satisfy the superposition principle, or in other words, the nonlinear system provides output which is not directly relative to its input. This section presents the controlled output of a well known nonlinear system, Duffing's equation in equation (2.5) which describes the relation between mass, stiffening spring and damper with a non minimum phase zero.

$$\ddot{y}(t) + \dot{y}(t) + y(t) + y^3(t) = 2u(t) - \dot{u}(t) \tag{2.5}$$

The Matlab simulink model of the above duffing's equation is shown in Fig. 2.6.

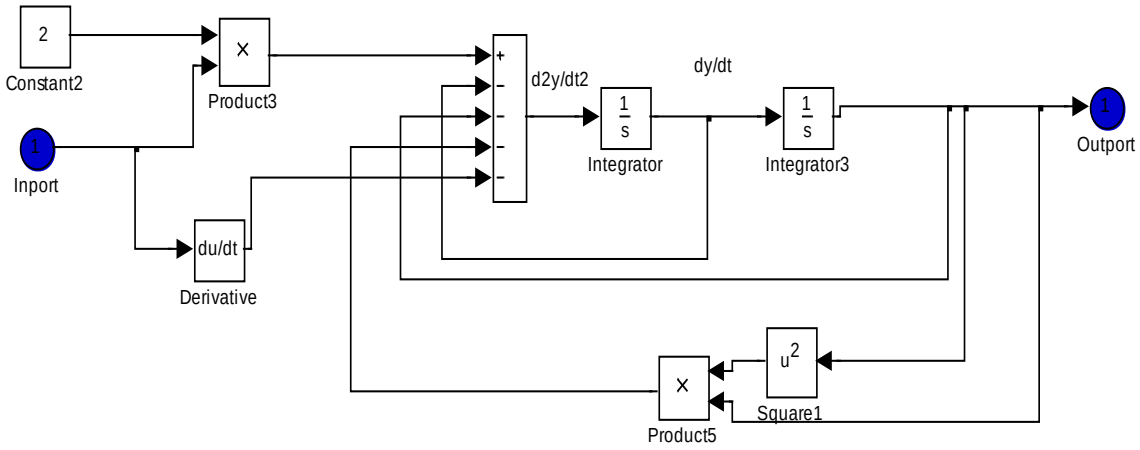

Fig. 2.6 Simulink model of Duffing's equation

This section describes the better capability of Neural Network based NMPC in comparison with linear MPC.

2.7.1 Linear MPC

A Linear time invariant state space form of the duffing's equation is used as the plant model. Then by Taylor's series method of linearization the equilibrium points are identified.

At Equilibrium point (0,0) At Equilibrium point (1,0) At Equilibrium point (-1,0)

$$A=\begin{bmatrix}0 & 1\\ -1 & -1\end{bmatrix} \qquad A=\begin{bmatrix}0 & 1\\ -4 & -1\end{bmatrix} \qquad A=\begin{bmatrix}0 & 1\\ 2 & -1\end{bmatrix}$$

$$B=\begin{bmatrix}0\\ 2\end{bmatrix} \qquad B=\begin{bmatrix}0\\ 2\end{bmatrix} \qquad B=\begin{bmatrix}0\\ 2\end{bmatrix}$$

$$C=\begin{bmatrix}1 & 0\end{bmatrix} \qquad C=\begin{bmatrix}1 & 0\end{bmatrix} \qquad C=\begin{bmatrix}1 & 0\end{bmatrix}$$

The prediction horizon N_p, control horizon N_c and control input weighting factor λ are set to is set to17, 2, 0.05 respectively. A quadratic program solver is used as the minimization routine for the cost function to compute the control signal. The process input is constrained to ±1 and output of the process is constrained to ±1.5.

2.7.2 Neural network based NMPC

A double layered feed forward NN comprising of five hidden neurons with sigmoid activation function and a single output neuron with linear activation function is used. The feed forward NN model of the duffing's equation is approximated as the nonlinear plant model. The identified multilayer feed forward neural network has 4 neurons in the hidden layer with two delayed plant inputs and plant outputs. The prediction horizon N_p, control horizon N_c and control input weighting factor λ are set to is set to17, 2, 0.05 respectively.

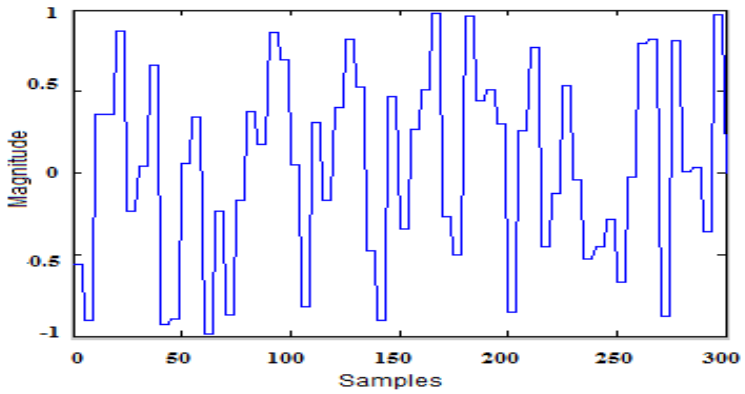

Fig. 2.7 Random signal used as input to the plant

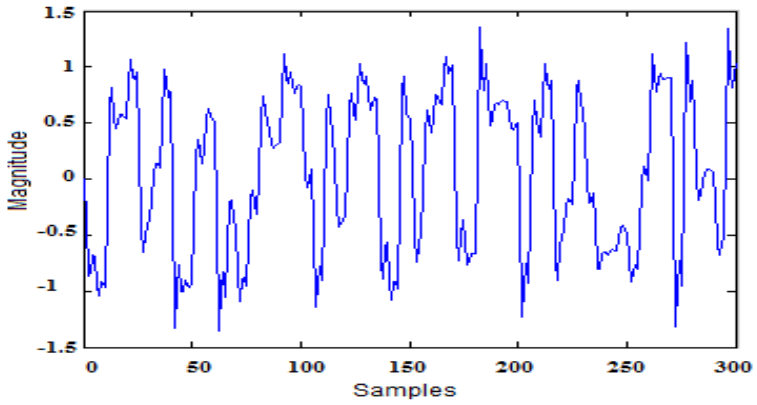

Fig. 2.8 Response of plant for random signal input

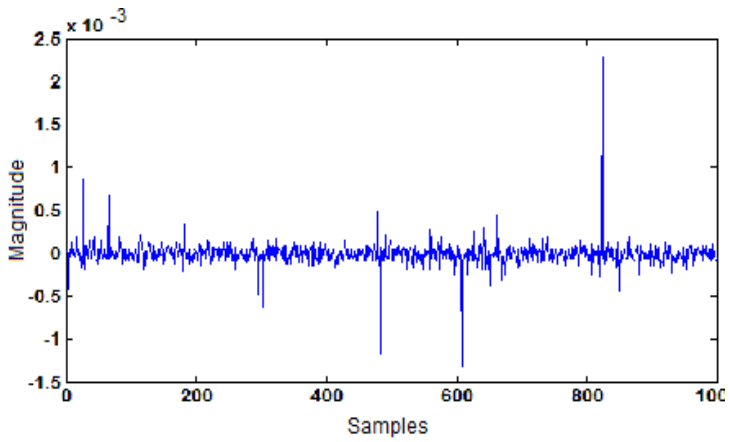

Fig. 2.9 Neural Network Prediction error

Training data is obtained by providing random signal as input to the plant and recording its corresponding outputs. The data used for training the neural network is given in Fig. 2.7 & Fig. 2.8. The neural networks offline training is performed using the same random signal through Levenberg-Marquardt learning algorithm with a learning rate of 0.01. Fig. 2.9 shows the deviation between the trained NN models predicted output and actual plant output for random input signals.

2.7.3 Simulation Results

The above trained NN model is then adopted as the nonlinear aproximated model in MPC. The simulation results of the tracking performance of LMPC and NMPC for duffing's equation are revealed in Fig. 2.10 and Fig. 2.11 correspondingly.

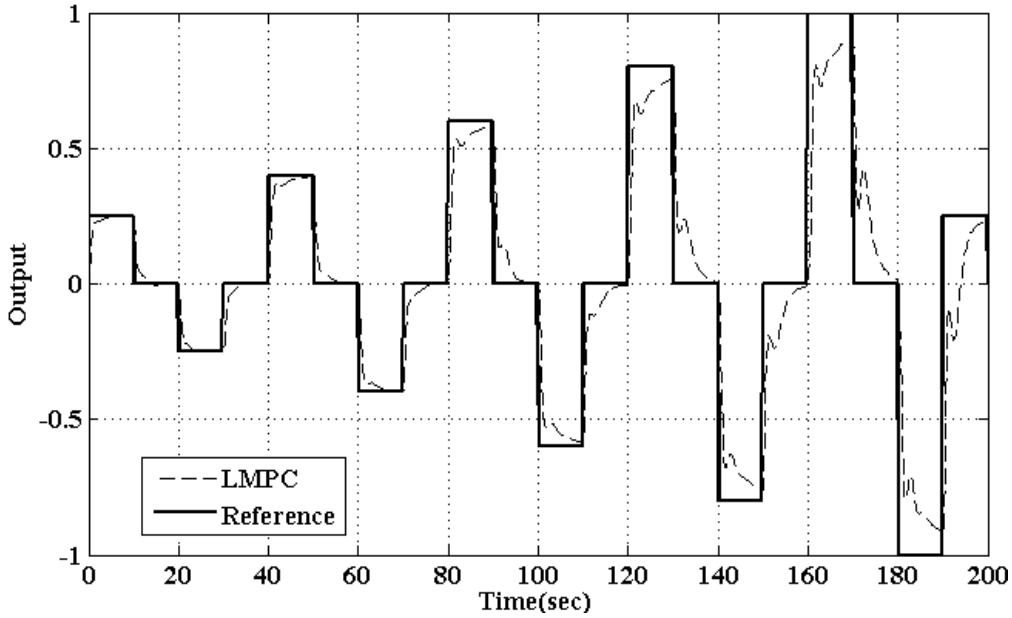

Fig. 2.10 Set point tracking performance of LMPC

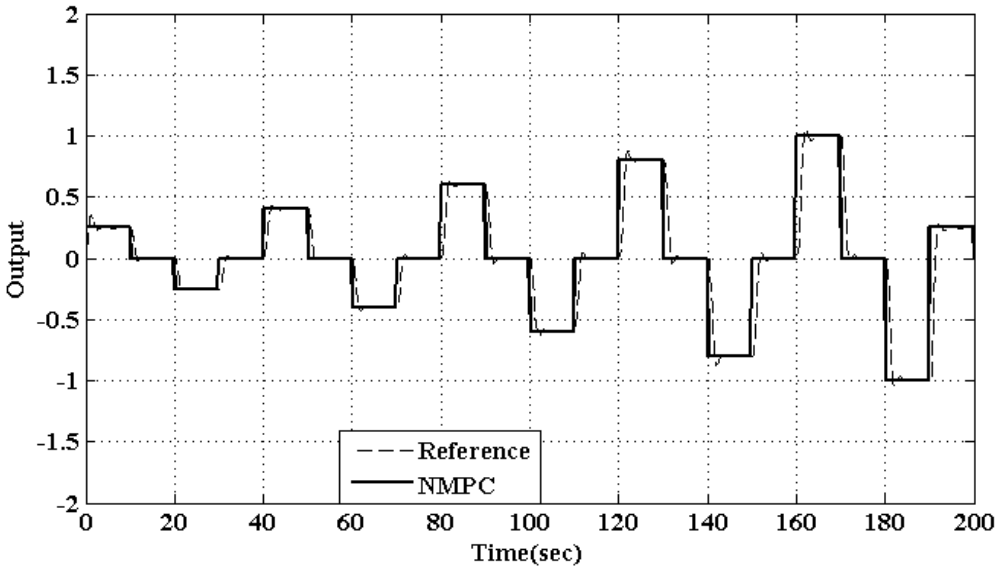

Fig. 2.11 Set point tracking performance of NMPC

From the simulation results it is clear that LMPC is not suitable for strong set point change problem over wide operating region. Thus neural network based NMPC dominates LMPC with smooth setpoint tracking performance even for strong set point changes.

2.8 PERFORMANCE COMPARISON OF NN BASED MPC WITH DIFFERENT ONLINE OPTIMIZATION TECHNIQUES

Nonlinear MPC does prediction and optimization at each sampling instant. Hence practical implementation of Nonlinear MPC is a challenging task. In this section the set point tracking performance of NN model based NMPC using line search routine based on backtracking technique and neural network based NMPC using particle swarm optimization are presented for the above mentioned duffings equation.

2.8.1 Particle Swarm Optimization

An ever-present and natural process that forms an essential part of our daily life is optimization. Basically, it is an art of choosing the finest choice among the available choices. Optimization techniques are being expansively used in the field of engineering.

After getting encouraged by the foraging performance of birds, American psychologist Kennedy and electrical engineer Eberhart introduced the particle swarm optimization algorithm [87]. The mechanism of particle swarm optimization is motivated from the complex social activities shown by the natural species like flock of birds, school of fish and even crowd of human beings [123]. It is an arbitrary optimization algorithm which is developed based on collaboration and competition among the particles in the swarm. PSO dependance on swarms cleverness is characterized by its universality and global optimization. The PSO algorithm is simple to put into practice and has been proven to be very competitive for solving diverse global optimization [124].

To make improvement in exploration and progress, Shi and Eberhart established inertia weight based on actual PSO [125]. As a first step, It initializes the swarm in possible solution space and velocity space with initial position and velocity, based on its distinct search principle. Each particle is a possible key of the optimization problem whose appropriateness value is dependent on the cost function. The particle velocity make a decision of the direction and distance of movement through the solution space. Generally, particles go after the current best solution and look for the ultimate finest solution which keeps on changing in each generation. In each generation, particles will go behind two extremums, one is the finest solution found by the individuals called pbest and the further is the finest solution found by any particle in their environs. Here, the global best value,

gbest is used, where every particle is associated to and able to attain information from every other particle in the swarm [126].

Basic steps involved in PSO algorithm are,

1. Initialization of population.
2. Evaluation of fitness of individual particles.
3. Modification of velocities based on pbest found by the individual and global best found by the neighbourhood.
4. Termination after attaining the required iterations.

Let us consider a feature space of n dimensional where the group $X=[X_1,...,X_2,\ldots,X_m]$ comprises m particles. The position and velocity of i^{th} individual particles is $X_i=[x_{i1},x_{i2},\ldots x_{in}]T$ and $V_i=[v_{i1},v_{i2},\ldots v_{in}]^T$ respectively and the best finest position the swarm has found individually, p_{best} is $P_i=[P_{i1},P_{i2},\ldots,P_{in}]^T$. The best location found by the swarm globally g_{best} is $P_g=[p_{g1},p_{g2},\ldots p_{gn}]^T$.

Now the velocity and position update equations for the particle X_i is given in (2.5) and (2.6) respectively.

$$v_{id}^{(t+1)} = \omega v_{id}^{(t)} + c_1 r_1 (P_{id}^{(t)} - X_{id}^{(t)}) + c_2 r_2 (P_{gd}^{(t)} - X_{id}^{(t)}) \quad (2.5)$$

$$x_{id}^{(t+1)} = x_{id}^{(t)} + v_{id}^{(t+1)} \quad (2.6)$$

Where,

$d=1,2,\ldots,n$	-	dimension of the solution space
$i=1,2,\ldots m$	-	particle
m	-	swarm size,
t	-	iteration counter,
w	-	inertia weights
r_1, r_2	-	random numbers in the range [0,1],
c_1, c_2	-	learning factors.
P_{id}	-	best position recognized by particle i
P_{id}	-	global best position recognized by particle i

Learning factors c_1 and c_2 ranges from [0, 4] but usually their value is taken to be 2. The basic flow chart for implementing PSO is given in Fig. 2.12.

Fig. 2.13 initially shows the random arrangement of particles in two dimensional optimization. Then as the number of itteration increases, the particles are moving closer to each other. This phenomenon repeats until most of the particles are coming to an unique optimum position.

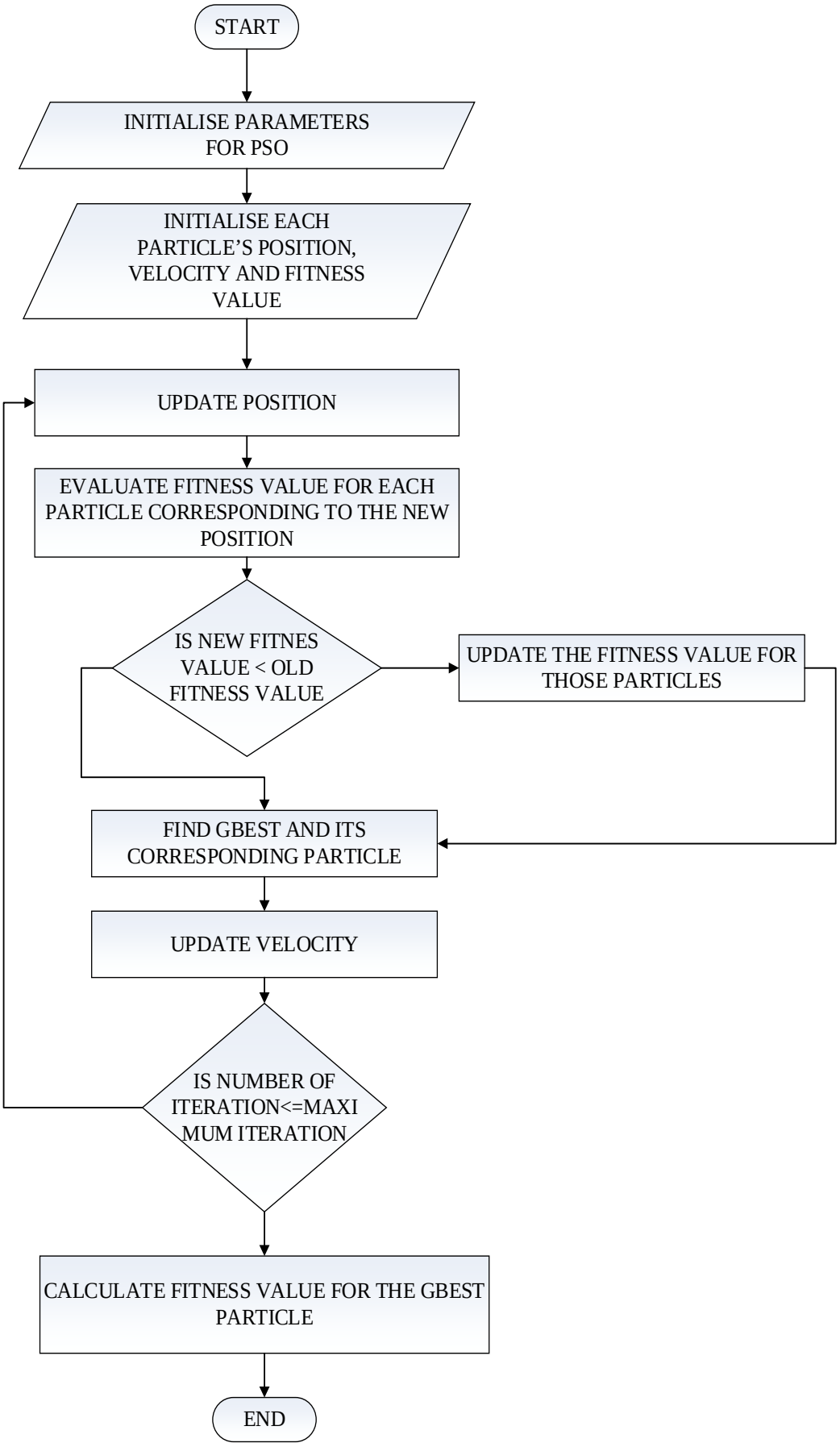

Fig.2.12 Flow chart of Particle Swarm Optimization algorithm

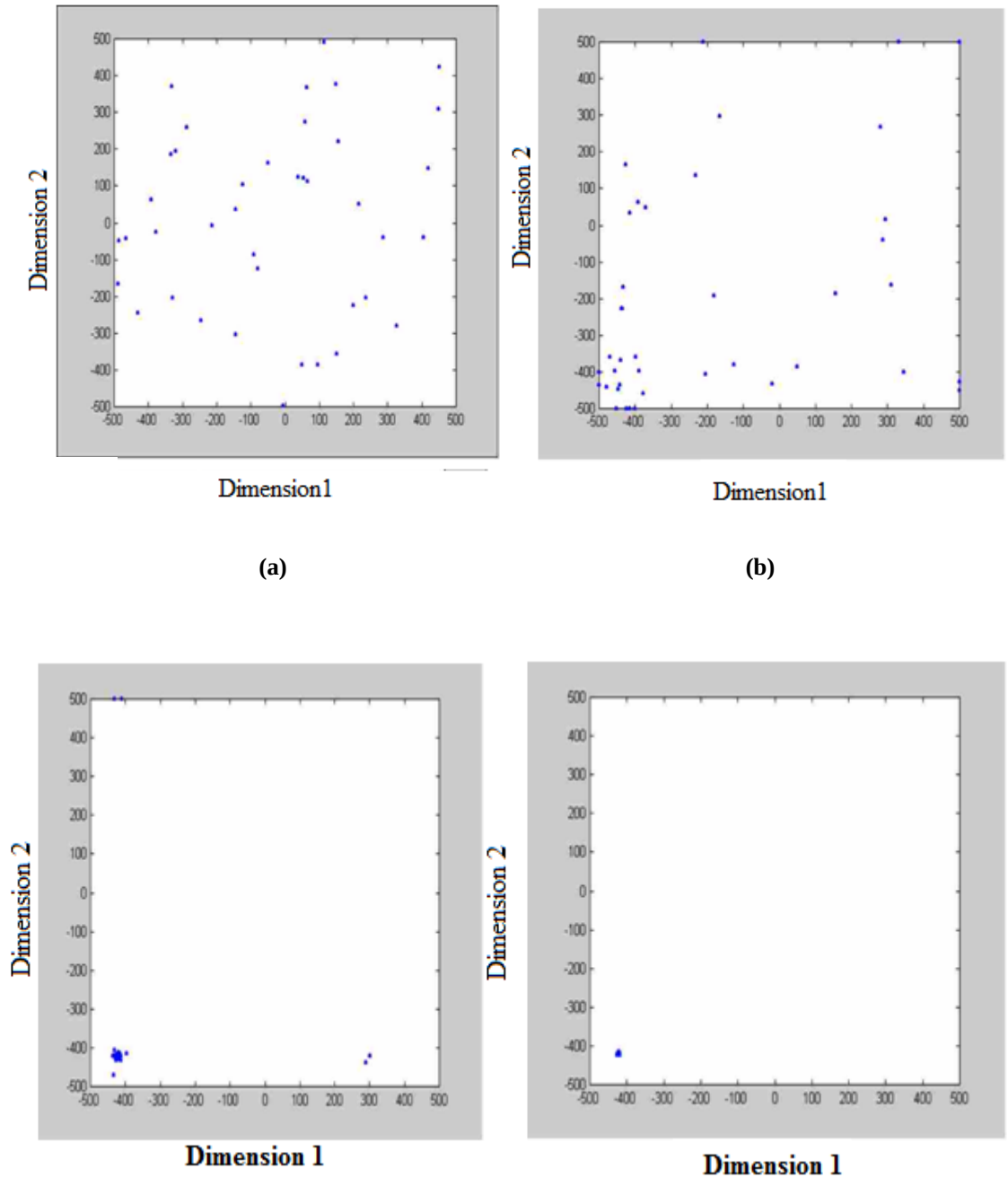

Fig. 2.13 (a) Initial position of the particles. (b) Position of the particles after 10 iterations. (c) Position of the particles after 100 iterations. (d) Position of the particles after 500 iterations.

2.8.2 Backtracking technique for optimization

Backtracking is a technique used to decide the distance to move in a particular search direction in an unconstrained method of optimization. This particular direction decides the minimization of the formulated cost function sufficiently. This is done numerically by a parameter α which gives an enough decrease in the objective function. Hence selection of α needs importance and is based on Armijo-Goldstein condition [127].

The backtracking algorithm actually reduces the value of α with rate parameter τ until the condition of Armijo-Goldstein is satisfied. It has the following steps.

(i) The iteration counter j is set to zero and the initial guess to the parameter α is $\alpha>0$ and rate parameter $\tau \in (0,1)$.

(ii) Repeat the following steps Until α^j satisfies the Armijo-Goldstein condition.

$\alpha^{j+1} = \tau\alpha^j$

$j = j+1$

(iii) Update the value of α by the above value.

2.8.3 Simulation Results

The set point tracking performance comparison of NN based MPC using line search method of optimization using backtracking technique and NN based MPC using PSO are presented in this section. The set point tracking performances of NN- MPC-backtracking and NN-MPC-PSO are shown in Fig. 2.14 and Fig. 2.15 correspondingly.

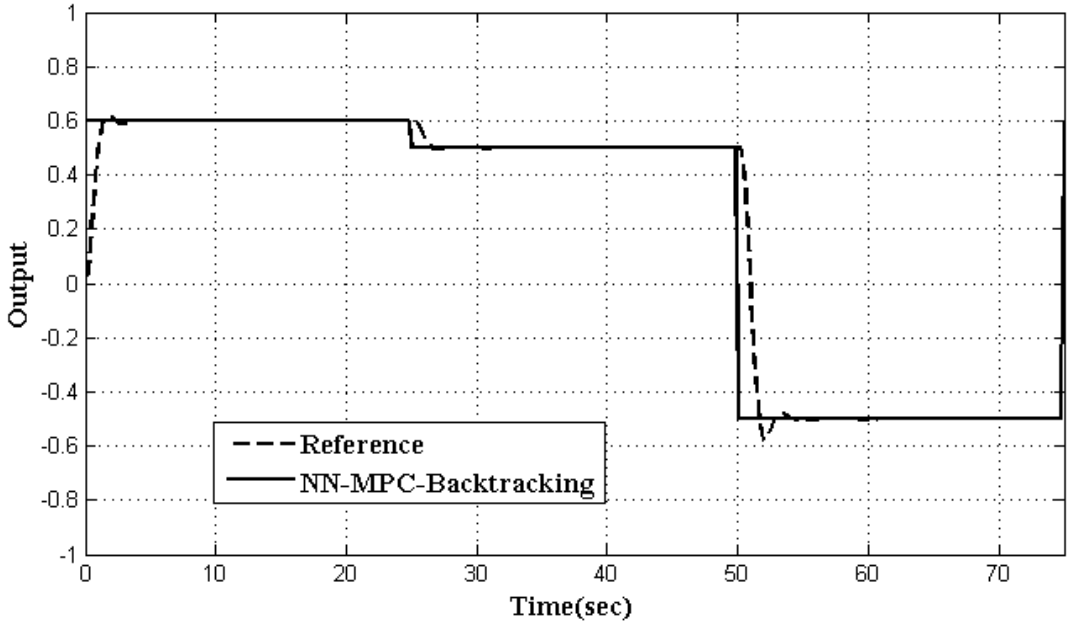

Fig. 2.14 Set point tracking performance of NN- MPC-backtracking

The performance of NN based MPC using PSO is better than NN based MPC using line search method of optimization with faster settling time. The little overshoots and undershoots in neural network based MPC using line search method of optimization is eliminated in neural network based MPC using PSO which exhibits its significance. Also as PSO is derivative free method, the computational expense of NN-MPC-PSO in less than NN-MPC-Backtracking as shown in Table 2.1.

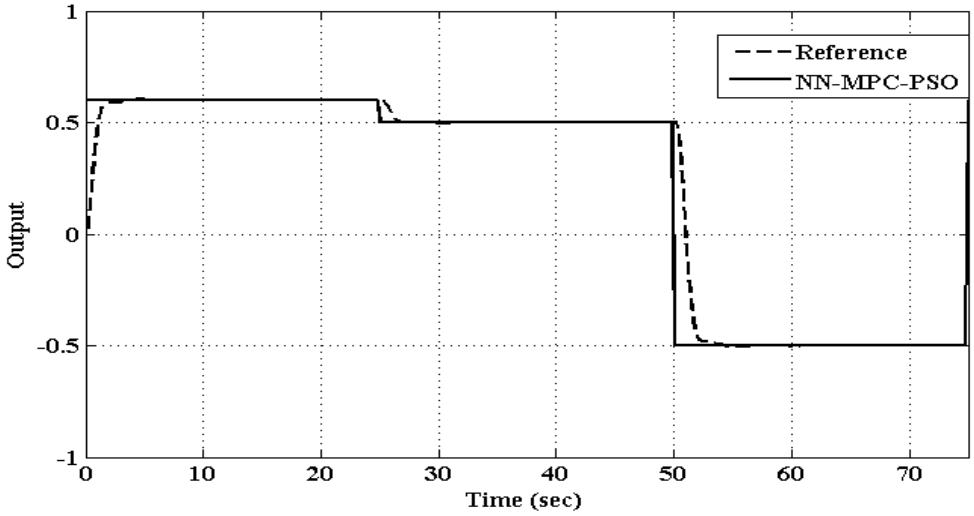

Fig. 2.15 Set point tracking performance of NN- MPC-PSO

Table 2.1 Comparison of computational cost

Optimization algorithm	Relative computational time
PSO	1
Line search with backtracking technique	1.7

2.9 CONCLUSION

The model predictive control using neural networks is described in this chapter with a short review on neural networks capability in nonlinear system modeling. The performance comparisons of linear MPC, nonlinear MPC, NN based MPC using line search method of optimization and NN based MPC using particle swarm optimization are done using MATLAB. The developed simulation results convey the importance of NN models in MPC and the better performance of PSO algorithm than conventional line search algorithm with relatively less computational time.

Chapter 3

MODEL PREDICTIVE CONTROL USING SUPPORT VECTOR MACHINES, RELEVANCE VECTOR MACHINES AND NEURO-FUZZY TECHNIQUES

This chapter describes model predictive control using Support vector machines and Relevance vector machines and neuro-fuzzy techniques. It starts with some background on kernel methods of system identification followed by basic principles of LS-SVM regression, RVM regression and neuro-fuzzy techniques. The significance of accurate and sparse model in MPC applications are explained by a benchmark example.

3.1 INTRODUCTION

One of the scientific disciplines in artificial intelligence is machine learning, which deals with algorithms that could develop performance based on empirical data collected from the process under study. The collected data should illustrate the relation between observed variables of the process. A learner captures the characteristics of their unknown underlying probability distribution with the help of data. The difficulty in machine learning lies in integrating all possible behaviour of a process by a set of experimental training records. The learner which becomes skilled from those data is expected to generalize the process behaviour from the training examples and to produce a useful output for unseen data.

Machine learning algorithms can be classified under different taxonomy as supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning, transduction etc. Supervised method understands the process behaviour from the input signal and supervisory output signal of training data. Thus the supervised learning algorithm has the capability to analyze the training data and to conclude them either as classification function for discrete data or as regression function for continuous data. For predicting the exact output for every valid input value the learning method should include the capability of generalization. The models incorporated in MPC's in this thesis are limited to supervised learning.

The following steps are to be performed to solve a supervised learning problem

1. Training data has to be collected. The approximated functions input features has to be determined. The accurateness of the approximated function depends powerfully on the illustration of input entity.
2. Hence the features in the input vector should be enough to predict the output accurately without creating the curse of dimensionality issue.

3. The configuration of the approximated function and analogous learning method has to be decided.
4. Training has to be performed on the selected algorithm. The trained function could be validated by optimizing performance or cross-validation on a validation set which is a subset of training set.
5. The precision of the trained function has to be evaluated by a test data set which is beyond the training set.

Kernel techniques are a group of novel methods for pattern analysis where support vector machines and relevance vector machines are vital elements. Kernel methods find the solution after explicitly mapping the data into the new high dimensional kernel Hilbert space; here the number of features of the data decides the number of coordinates. This method of finding the relation in the feature space by the help of kernel function by merely calculating the inner products among the images of all data pairs is called kernel trick. This method of finding the relation is very simple than the computation of coordinates of data in the high dimensional space. Support vector machine, Relevance vector machine, Fisher's linear discriminant analysis (LDA), Gaussian processes, principal components analysis (PCA), canonical correlation analysis, ridge regression, spectral clustering, linear adaptive filters are some of the algorithms operating with kernels.

Commonly used kernel functions are Linear Kernel, Polynomial Kernel, Gaussian Kernel, Circular Kernel, Bessel Kernel, Bayesian Kernel, Wavelet Kernel etc. Selecting a suitable kernel plays importance. Kernel function has to be selected based on the problem at hand [128]. In this thesis Gaussian kernel function is selected while using RVM and SVM for modeling.

One among the commonly used curve fitting methods is least square approach of polynomial curve fitting**.** Consider a training dataset $\mathbf{x} = (x_1, x_{2,\ldots,} x_N)^T$, which consists of N number of elements such that each element is a surveillance of *x* and let for each surveillance of *x* there is a corresponding value t. This dataset will be known as target dataset which can be written as $\mathbf{t} = (t_1, t_{2,\ldots,} t_N)^T$. Here, the target dataset is synthetically generated from the $\sin(2\pi x)$ function with some random Gaussian noise added to it. The regression problem of the above function is discussed below. The input training dataset **x** is generated by selecting values of x_n , for *n*=1,…, N, spread out uniformly in the range [0,1]. Real time dataset generation can be captured by adding Gaussian noise to the target dataset.

The objective is to predict the target variable $\hat{t}$ for some unknown value of $\hat{x}$ with the help of training dataset. For such prediction, the datasets must be fitted by a nonlinear polynomial function of the form shown in equation (3.1)

$$y(x, \mathbf{w}) = w_0 + w_1 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j \tag{3.1}$$

Where

M - polynomial order.

w - vector representing the coefficients of the polynomial ie.,

$\mathrm{w} = (w_0, \ldots, w_M)$

The polynomial $y(x,\mathrm{w})$ is a nonlinear function with respect to x but it is linear with respect to the coefficients of x which are the unknown parameters. The coefficients of x is to be selected in such a way to diminish the error among $y(\mathrm{x},\mathrm{w})$ and the training dataset. The error function measured here is the summation of the squared error between the predictions $y(x_n,\mathrm{w})$ for each data point x_n and the corresponding target values t_n:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 \tag{3.2}$$

The minimization routine of (3.2) chooses the value of w in such a way that makes $E(\mathbf{w})$ as small as possible. The derivatives of the $E(\mathrm{w})$ with respect to w will be a linear function since $E(\mathrm{w})$ function is a quadratic function of w**.** Next step is the selection of the value of M which is the order of the polynomial. As exposed in Fig. 3.1 the fitting of data is poor if the value of M is 0 or 1 and the polynomial fitting is the best one function $\sin(2\pi x)$, if the value of M is 3, the polynomial gives the best fit to the functions$\sin(2\pi x)$. Increasing the order of the polynomial to 9 makes the polynomial curve to pass through all the data points giving an outstanding fitting of the data with $E(\mathrm{w})=0$. Even though M=9 gives outstanding fitting of the data, it suffers from generalization problem, i.e., the prediction for unseen data is very poor since the fitted polynomial violently oscillates at different points. Hence, with respect to the best generalization, M=3 polynomial gives the best data fit. Fig. 3.1 gives the polynomial plots for the values of M.

A novel powerful deterministic methodology for evaluation of nonlinear function and nonlinear classification based on kernel methods is Support Vector Machine which avoids over fitting problem and improves generalization ability with less number of training data and less training time. In LS-SVM the regularization parameter γ and the kernel width parameter σ should be tuned to improve the generalization ability of predicted model. This SVM method has better generalization capability. The overall precision of the

SVM method is better than the overall precision of the evolutionary fuzzy rules [129]. Combinations of SVM models and MPC approaches are also discussed in [17- 20].

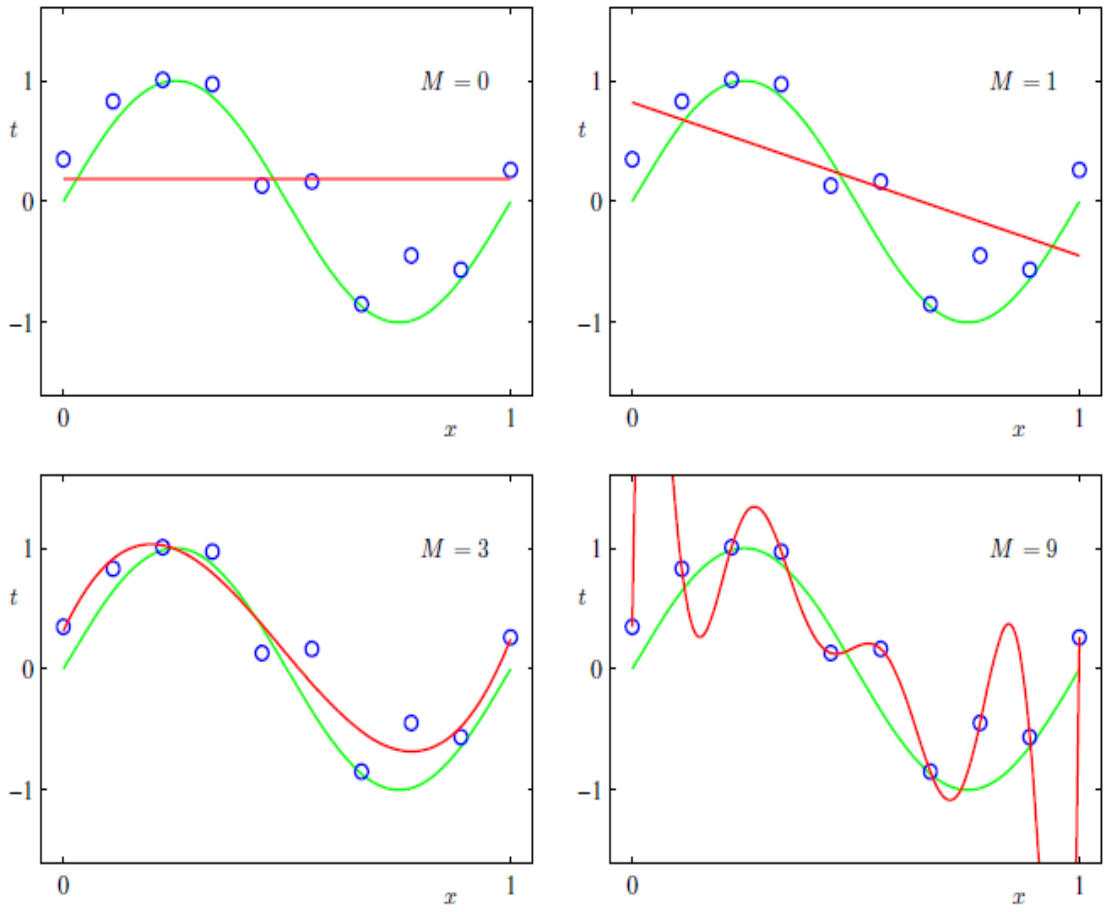

Fig. 3.1 Polynomial plots for the values of M (red colour curve)

RVM is an advanced probabilistic machine learning technique for classification and regression that uses Bayesian inference to get sparing solutions. Relevance vector machines results in usage of less number of relevance vectors leading to much more sparse representation than Support Vector Machine. The functional form of RVM and SVM are same. In the case of RVM no constraint is put on the basis functions whereas SVM is forced to satisfy the Mercer's kernel theorem [15, 22]. Also, RVM modeling is made simple by having kernel width σ as the only tuning parameter. The RVR model is much sparser than SVM model as the relevance vectors are very less than support vectors. As a result sparse RVR model could perform better with better generalization and less computation time than SVM.

In Fig. 3.2 the one in the right side seems to be over-fitting and the one in the left shows seems to be an ideal fit. But without prior knowledge about the approximated function the above decision cannot be done.

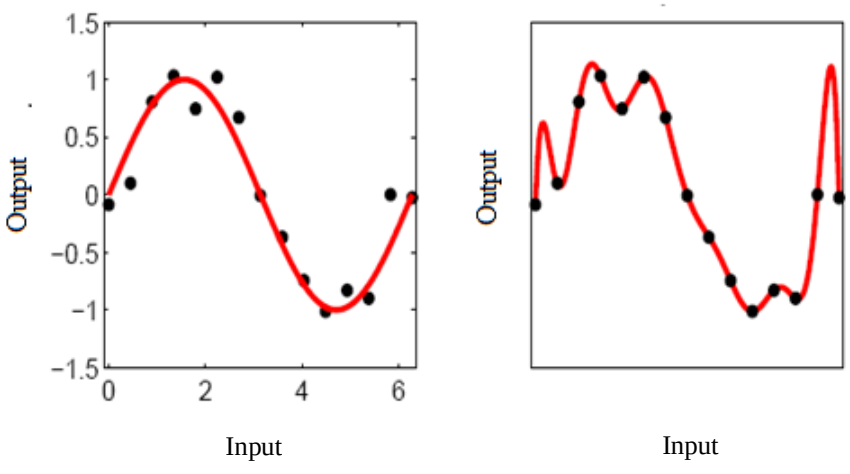

Fig. 3.2 (a) Ideal fitting (b) Over fitting

The only way for a meaningful judgment is by imposing a priori based on previous knowledge about the function. This could be done commonly by regularization. In relevance vector regression, in order to avoid these over fitting difficulty, restraints are established on the weight parameters, which describe the specific preferred assets of the approximated function. The Bayesian method offers constraints on the weight parameters by considering the parameters as arbitrary variables, for which prior probabilistic distributions are initiated. Combination of relevance vector regression model and MPC are less reported in literatures.

Another universal estimator, Adaptive neuro-fuzzy inference system is a class of NN which depends on Takagi–Sugeno fuzzy inference system. As it puts together the principles of both NN and fuzzy logic, it gains the strength of both the techniques in a single structure. The inference mechanism of ANFIS is based on several membership functions with typical fuzzy If-Then rule which describes the dynamics of the nonlinear system.

A new neuro-fuzzy learning technique, Extreme ANFIS is proposed in this chapter which depends on the concept of Extreme Learning Machine [23]. The proposed algorithm

has minimized the learning time of ANFIS architecture which is very simple and eliminate drawbacks of gradient based Hybrid Learning Algorithm.

Next section of this chapter describes basic theory behind Support vector machines, Relevance vector machines and neuro-fuzzy techniques. The significance of accurate, sparse model and proposed novel neuro-fuzzy technique in MPC applications are explained by benchmark examples which concludes this chapter.

3.2 SUPPORT VECTOR MACHINES

Support Vector learning is based on effortless thoughts which initiated in statistical learning theory. It is a class of supervised learning algorithms first introduced by Vapnik [14]. SVM's are fast replacing neural network as the tool of choice for classification and regression tasks, primarily due to their ability to generalize well on unseen data. SVM's are exemplified by usage of kernels, nonexistence of local minima, and sparse nature of the solution. Although SVM's are being used mainly for classification tasks, newly SVM's have been effectively applied to solve regression problems [59]. SVM performs better than neural network model due to its better generalization capability [130].

SVMs build a hyperplane which divides examples such that examples of one class are all on one side of the hyperplane, and examples of the other class on the other side.

Consider input data of the form (x_i, y_i) where the vectors x_i are in a dot product space H, and y_i are the class labels. Formally, any hyperplane in H is defined as

$$\left\{x \in H \middle| \langle w, x \rangle + b = 0\right\} \quad w \in H,\ b \in R \tag{3.3}$$

where *w* is a vector orthogonal to the hyperplane and $\langle\ \rangle$ represents the dot product and b is the bias. In an SVM, the idea is to find the hyperplane that maximizes the minimum distance (called the margin) from any training data point (Fig. 3.3). The following constraint problem describes the optimal hyperplane:

$$\min_{w \in H, b \in R} \tau(w) = \frac{1}{2}\|w\|^2 \tag{3.4}$$

subject to $y_i\left(\langle x_i, w \rangle + b\right) \geq 1$

for i = 1, 2, …, m where m is the number of training examples.

The above problem can be solved by introducing the Lagrange multipliers ($\alpha_i \geq 0 (i = 1,..., m)$) and maximizing the following dual problem

$$\max_{\alpha \in R} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \tag{3.5}$$

subject to, $\alpha_i \geq 0, i = 1,...., m,$

$$\text{and } \sum_{i=1}^{m} \alpha_i y_i = 0 \tag{3.6}$$

The patterns x_i which correspond to non-zero Lagrange coefficients are called support vectors. The resultant decision function has the following form

$$y(x) = \text{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i \langle x_i, x \rangle + b \right) \tag{3.7}$$

Thus the optimal margin hyperplane is represented as a linear combination of training points. Consequently, the decision function for classifying points with respect to the hyperplane only involves dot products between points. Furthermore, the algorithm that finds a separating hyperplane in the feature space can be stated entirely in terms of vectors in the input space and dot products in the feature space.

When the samples are not linearly separable, a kernel function is used to transforms the data to a higher dimensional space where it is linearly separable and then applies the hyperplane. The kernel function gives the dot product of the two examples in the higher dimensional space without actually transforming them into that space. This notion, dubbed the kernel trick, allows us to perform the transformation for purposes of classification to large dimensional spaces. In the nonlinear case, resultant decision function has the following form

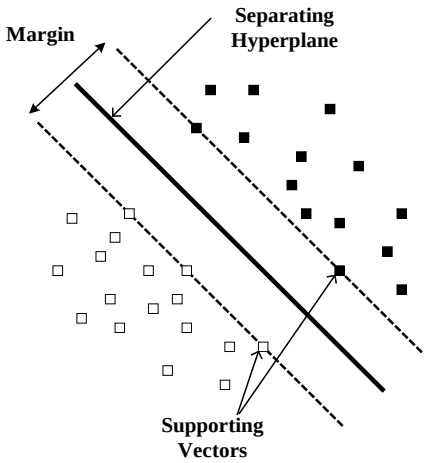

Fig. 3.3 Maximum margin and optimal hyperplane

Support vector machines analyze data and recognize pattern by supervised learning models using kernel methods for both classification and regression analysis. In SVM the data points are separated by selecting the best hyper plane focusing the target as largest separation or margin. In other words, the separating hyper plane is selected so that the

distance from it to the nearest data point on both the sides is maximum. The data point touching this maximum margin hyperplane is called as support vector.

$$y(x) = \text{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i K(x_i, x) + b\right) \tag{3.8}$$

where the kernel function $K(x_i, x) = \langle \phi(x_i), \phi(x) \rangle$ and $\phi(x)$ is the nonlinear map from original space to the high dimensional space. Two of the most commonly used kernel functions are polynomial functions and Gaussian radial basis functions and are given by

Polynomial kernel: $K(x, x') = \left[1 + x^T X'\right]^k \quad k = 2,3$ (3.9)

Radial basis kernel: $K(x, x') = \exp\left[-\frac{1}{2}\|x - x'\|^2\right] \Big/ \sigma$ (3.10)

where σ is the spread of the Gaussian function.

The maximum margin allows the SVM to select among multiple candidate hyperplanes; however, for many data sets, the SVM may not be able to find any separating hyperplane at all, either because the kernel function is inappropriate for the training data or because the data contains mislabeled examples. The latter problem can be addressed by using a 'soft margin' that accepts some misclassifications of the training examples. In this case introducing slack variables ξ_i and error penalty *C*, the optimal hyperplane can be found by solving the following new quadratic optimization problem [131]

$$\text{minimize } \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} \xi_i \tag{3.11}$$

$$\text{subject to } y_i\left(\langle x_i, w \rangle + b\right) \geq 1 - \xi_i \tag{3.12}$$

$$\xi_i \geq 0, \quad \text{for all } i$$

The value of *C* is set by the user and larger value of *C* leads to a larger penalty for an error. The problem can be solved by the dual formulation as in the linear separable case and a decision boundary of the form given in equation (3.8) can be obtained. The value of α_i is different from the separable case and is given by $0 \leq C \leq \alpha_i$.

3.2.1 Least Squares Support Vector Machines

Least squares support vector machines are least square versions of SVM's, first proposed by Suykens and Vandewalle. In this version, the solution is made simpler by resolving a set of linear equations as an alternative to the convex quadratic programming problem for classical SVMs. The significant dissimilarity with classical SVMs is the equality constraints and the sum squared error term [132]. The inequality constraints with the slack variable in equation (3.12) are replaced by the equality constraints with an error

variable E_i in equation (3.7) and the squared loss function is included in the objective function equation (3.6). The above modifications make it a simple linear problem.

Consider a particular training set of M regression data points $\{(x_i, y_i)\}_{i=1}^{M}$, where $x_i \in R^M$ is the input data to the real plant and $y_i \in R$ is the output data of the real plant.. In high dimensional feature space Z , LS-SVM model is,

$$y(x) = w^T \varphi(x) + b \qquad \text{where } w \in Z, b \in R \tag{3.13}$$

In the above nonlinear function estimation model, the weight vector w and the bias term b are the two parameters to be identified. $\varphi(.)$, maps the input data into a high dimensional feature space Z. In least squares version of SVM the optimization problem formulated is as follows.

$$\min_{w,e} J(w,e) = \frac{1}{2} w^T w + \frac{\gamma}{2} \sum_{i=1}^{M} E_i^2, \qquad \gamma > 0 \tag{3.14}$$

This is subjected to the following constraints

$$y_i = w^T \varphi(x_i) + b + E_i, \qquad i = 1,2,3.....,M \tag{3.15}$$

Where,

γ - Regularization parameter.

E_i - Deviation of actual output from predicted output of the i[th] data.

The Lagrange function for equation (3.6) is

$$L = \frac{1}{2} w^T w + \frac{\gamma}{2} \sum_{i=1}^{M} E_i^2 - \sum_{i=1}^{M} \alpha_i [w^T \varphi(x_i) + b + E_i - y_i] \tag{3.16}$$

where α_i -Lagrange multiplier.

According to Karush–Kuhn–Tucker conditions,

$$\left.\begin{aligned} &\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{M} \alpha_i \varphi(x_i) \\ &\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{M} \alpha_i = 0 \\ &\frac{\partial L}{\partial \alpha_i} = 0 \Rightarrow w^T \varphi(x_i) + b + E_i = y \\ &\frac{\partial L}{\partial E_i} = 0 \Rightarrow \alpha_i = \gamma E_i, i = 1,2,....,M \end{aligned}\right\} \tag{3.17}$$

All the above equations in Equation (3.9) are first transformed into a matrix form and then substituting the values of E and w results in the following matrix equation,

$$\begin{bmatrix} 0 & \vec{1}_M \\ \vec{1}_M & \varphi(x_i)^T \varphi(x_j) + \gamma^{-1} I_M \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \tag{3.18}$$

where $y = [y_1 ..... y_M]^T, \vec{1}_M = [1 ..... 1]^T, \alpha = [\alpha_1 ..... \alpha_M]^T$

I_M is an *MXM* identity matrix and $\varphi(x_i)^T \varphi(x_j) = K(x_i, x_j)$, *I, j=1, 2... M is* any kernel function satisfying the Mercer condition [133].

The parameters α and b can be obtained as a Solution of Equation (3.9) and hence LS-SVM predicted model of the given dataset is as follows,

$$\hat{y}(x) = \sum_{i=1}^{M} \alpha_i K(x_i, x) + b \tag{3.19}$$

In the present work Radial basis function (RBF) in Equation (3.12) is selected as kernel function because of its ability to decrease computational complication of the training process and to develop generalization capability of LS-SVM.

$$k(x_{i,} x) = \exp\{-\|x_i - x\|^2 / 2\sigma^2\} \tag{3.20}$$

where σ is the kernel width.

Thus in LS-SVM in order to obtain the predicted model there are two free parameters to be tuned i.e., regularization parameter γ and kernel width parameter σ. In LS-SVM these are the two parameters which decide the generalization ability of predicted model. Hence optimization of these parameters plays a momentous role. These parameters are tuned using the algorithm of Coupled Simulated Annealing (CSA) and simplex method [134]. Initially the global optimization technique Coupled Simulated Annealing determines suitable values for those parameters and then further optimization is done by simplex method to get finely tuned values of those parameters to achieve accurate prediction.

3.2.2.1 Principle of coupled Simulated Annealing

It is a global optimization algorithm which bears a resemblance to the annealing process carried out in material physics. This optimization technique was proposed by Kirkpatrick. The ideas of Boltzmann [135] and Metropolis et al. [136] influenced Kirkpatrick et al. [137]. Kirkpatrick et al. repeated the approach of Metropolis for all temperatures on the annealing agenda till the arrival of thermal equilibrium. [138].

Following are the steps involved in Simulated Annealing algorithm.

1. The upper bound and initial values of the SVM kernel function parameters γ and σ are set and fed to the support vector machine model and the predicted error is

obtained from the state of the system, which furnish the primary state of the system.

2. Then random values are selected to the two kernel function parameters and the new system state is forecasted.
3. The latest state is acknowledged or discarded based on the following equations.

   Accepted the new state if

   $$Energy_{(new\ state)} > Energy_{(old\ state)} \ and \ \ p < P\ ,\ 0 \le p \le 1$$

   or

   Reject the new state otherwise i.e., $Energy_{(new\ state)} \le Energy_{(old\ state)}$. Here, p is the arbitrary number to decide the recognition of new state and P is the possibility of accommodating the latest state.
4. In case of non acceptance of new state go to step 2 and repeat step 2 and 3 until the acceptance of new state.
5. After acceptance of a new system state, the temperature is reduced. The temperature reduction is obtained by the following equation,

   $$New\ Temperature = (Current\ Temperature) \times \rho$$

   Where, $0 < \rho < 1$

   Stop the algorithm when the predetermined temperature is reached. Otherwise go to step 2.

After this initial optimization by CSA algorithm the SVM kernel function parameter values are again optimized by simplex method to get fine tuned SVM kernel function parameters.

Simplex search method is a popular algorithm for linear programming. The Simplex method is a method that precedes from one extreme point of the feasible region of a linear programming problem to another extreme point, in such a way as to continually increase (or decrease) the importance of the objective function unless optimality is attained.

The well known function sinc(x) = sin(x)/x is chosen to demonstrate the support vector regression. SVM is applied to some simple synthetic datasets of sinc function and SVM regression is shown. Gaussian function is the Kernel function chosen for the SVM model. The target dataset of this regression problem is assumed to have additive Gaussian noise components.

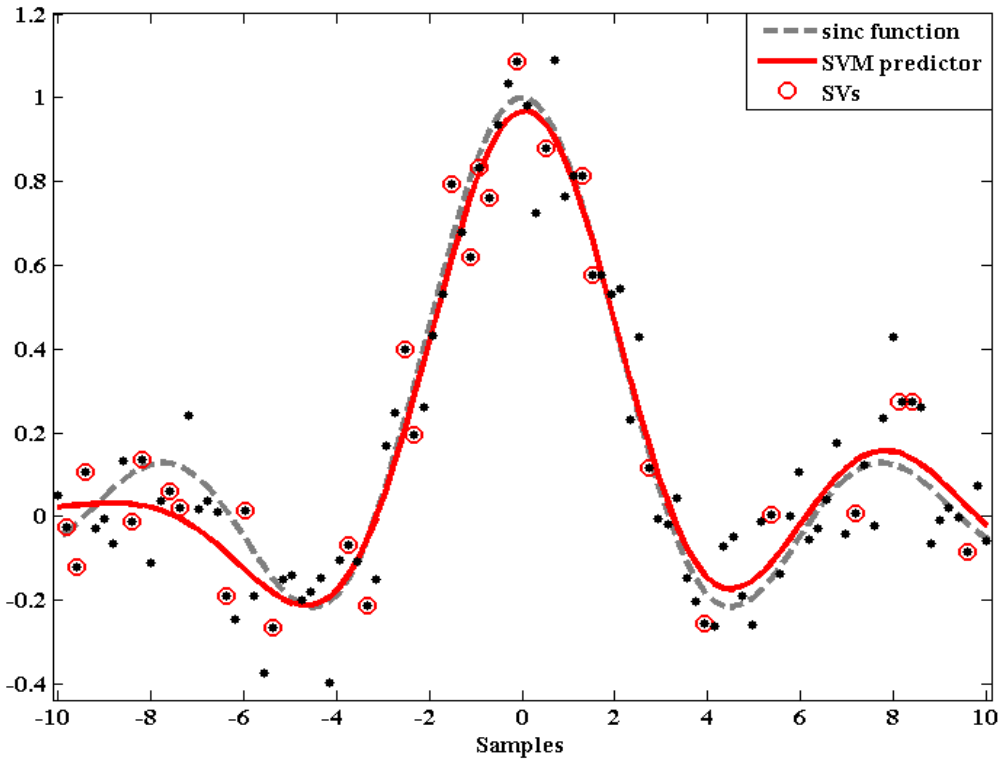

Fig. 3.4 SVM regression of 'sinc' function

In Fig. 3.4, the actual 'sinc' function is shown by dashed line and the noisy version of the 'sinc' function is shown by black dots. The objective is to predict the 'sinc' function with minimum amount of error and minimum number of support vectors. For this sinc function example 100 input vectors (100 black dots) have been used as dataset but the prediction is done by only 29 vectors which are known as support vector machine predictor. The RMSE value between the predicted output and the actual output is 0.0491.

3.3 RELEVANCE VECTOR MACHINE AND PROBABILITY THEORY

A novel probabilistic sparse kernel learning technique used for nonlinear classification/ regression is relevance vector machine. To represent ambiguity in an experiment, probability theory can be used. Probability theory is based on rational coherent inferences and it also gives significance to the common sense. Bayesian perspective is involved in addressing the amount of uncertainty involved in selecting the suitable model parameters, the weight vector w. The prior probability p(w), captures the former assumptions about the model parameters. Given the weight vector w, the targets of the observed data $\mathrm{t} = (t_1, t_{2,\ldots,} t_N)^T$ can be captured into the conditional probability p(t|w) and it is also known as the likelihood function because for different values of w, it represents

the quantity of likeliness of the observed data set . Hence, according to the Baye's theorem below,

$$p(\mathbf{w} \mid \mathbf{t}) = \frac{p(\mathbf{t} \mid \mathbf{w}) p(\mathbf{w})}{p(\mathbf{t})} \tag{3.21}$$

where,

$p(w|t)$ - posterior probability

$p(w)$ - prior probability

$p(t|w)$ - likelihood function

Posterior probability is the uncertainty in *w* after the data has been observed. In equation (3.13) the posterior probability is directly proportional to the prior and the likelihood function since the denominator of (3.13) is only acting as a normalizing constant.

The likelihood function plays a significant role in Bayesian and maximum likelihood estimate (MLE). Still, its role of play in both the methods is rather different. In MLE, *w* is estimated accordingly to maximizes the likelihood function i.e., maximum probability is for getting the observed target data *t*. But, in case of the Bayesian approach, the uncertainty of *w* is expressed by posterior probability. The main benefit of the Bayesian method is the presence of prior assumptions or knowledge. But the difficulty is that the prior is chosen based on the mathematical convenience rather than the prior belief and hence if the chosen prior is poor than Bayesian techniques gives poor results.

3.3.1 Bayesian inference and Relevance vector machine

Bayesian inference is a technique of statistical inference that uses few kinds of interpretation to compute the probability to decide if the hypothesis may be true. The usage of bayes theorem in the evaluation process has given the name "Bayesian". This Bayes theorem was deduced by Thomas Bayes. Practically, Bayesian inference leads to the usage of prior probability to judge the probability of another assertion. In other words the posterior probability of the assertion comes as the mixture of prior probability and likelihood of the evidence.

The Bayes method involves the following steps.

(i) A probabilistic model is developed. The model is defined in such a way to express qualitative aspects of our knowledge with some unknown parameters. The prior probability distribution for these unknown parameters is also specified.

(ii) Data is collected.

(iii) From the observed data, the posterior probability distribution for the parameters is calculated. This posterior distribution is used to make prediction or decision.

The posterior distribution for the parameters of the model which is the combination of the prior distribution and the likelihood for the parameter is completed by means of Bayes' Rule:

$$P(parameters/data) = \frac{p(parameters)\ p(data/parameters)}{p(data)} \tag{3.22}$$

The above equation shows,

$$P(parameters/data) \propto p(parameters)\ p(data/parameters) \tag{3.23}$$

The schematical representation of the above equation is,

$$posterior \propto prior \times likelihood \tag{3.24}$$

Prediction for unknown new data is obtained by integrating with respect to the posterior:

$$p(new\ data/data) \propto \int_{parameters} p(new\ data/parameters)\ P(parameters/data) \tag{3.25}$$

This Bayesian inference is applied in the context of machine learning using relevance vector machine. The task of Bayesian inference in machine learning technique is to estimate P(t/x) with some particular model depending on input-output pairs [22, 139].

3.3.2 Sparse Bayesian learning for regression using RVM

RVM is a probabilistic model with similar functional form as that of Support Vector Machine. The prediction accuracy of RVM is comparable to that of SVM with very less number of kernel functions [140]. RVM is based on Bayesians approach in which a prior is introduced over the model weights and each weight is administrated by one hyper parameter. The most probable value of each hyper parameter is iteratively evaluated from the data. The model is sparser since the posterior distributions of a few percentages of the weights are made zero.

Consider a given training set of M regression data points $\{(x_m, y_m)\}_{m=1}^{M}$, where $x_m \in R^M$ is the input data to the actual plant and $y_m \in R$ is the output data of the actual plant and is assumed to contain Gaussian noise ε with mean 0 and variance σ^2. In high dimensional feature space z, the outputs of an extended linear model can be articulated as a linear combination of the response of a set of *M* basis functions, as follows:

$$y(x,w)=\sum_{m=1}^{M} w_m \varphi_m(x) + w_0 + \varepsilon = w^T \varphi + \varepsilon \tag{3.26}$$

Now the predicted output $\hat{y}$ of the true value y is,

$$\hat{y}(x,w)=\sum_{m=1}^{M} w_m \varphi_m(x) + w_0 = w^T \varphi \quad \text{where } w \in Z \tag{3.27}$$

In the above nonlinear function estimation model, w_m is the weight vector and $\varphi_m(.)$ is an arbitrary basis function (or kernel). In the present work RBF is used as the kernel function because of its ability to reduce computational complexity of the training process. The vector form of $w=[w_0, w_1, ...w_M]^T$ and the responses of all kernel function $\varphi(x)=[\varphi_1(x).....\varphi_M(x)]^T$ maps the input data into a high dimensional feature space *z*.

Hence the obtained error signal could be stated as

$$\varepsilon_m = y_m - \hat{y}_m = N(0,\sigma^2) \tag{3.28}$$

The objective of relevance vector regression is to find the finest value of w such that $\hat{y}(x,w)$ makes fine estimations for unknown input data. For the RVM model in (3.27) let $\alpha=[\alpha_0, \alpha_1, ...\alpha_M]^T$ be the vector of *M* independent hyperparameters, each associated with one model weight or kernel function.

The Gaussian prior distributions of the RVM framework are chosen as shown below,

$$p(w_m/\alpha_m)=\prod_{m=1}^{M}(\alpha_m/2\pi)^{1/2}\exp\{-\alpha_m {w_m}^2/2\} \tag{3.29}$$

Here α_m is the hyperparameter that governs each weight w_m

The likelihood function of independent training targets $y=y_m, m=1....M$ can be stated as,

$$p(y/w,\sigma^2)=\frac{\exp\{-\left\|(y-\hat{y})^2\right\|/(2\sigma^2)\}}{\sqrt{(2\pi\sigma^2)^M}} \tag{3.30}$$

The above likelihood function is enhanced by the prior in (3.29) defined over each weight to trim down the complication involved in the model and to avoid over fitting.

Now using Bayes rule, the posterior distribution over model weights could be calculated as follows,

$$p(w/y,\alpha,\sigma^2)=\frac{p(y/w,\sigma^2)\,p(w/\alpha)}{p(y/\alpha,\sigma^2)} \tag{3.31}$$

The posterior distribution in (3.23) is a Gaussian distribution function,

$$p(w/y,\alpha,\sigma^2) = N(\mu,\sigma^2) \tag{3.32}$$

Whose covariance and mean are respectively given by,

$$\Sigma = (\sigma^{-2}\varphi^T\varphi + A)^{-1} \tag{3.33}$$

$$\mu = \sigma^{-2}\Sigma\varphi^T y \tag{3.34}$$

With $A = diag(\alpha)$

Marginalization of the likelihood over the target data is given by (3.30) and could be obtained by combining out the weights to acquire the marginal likelihood for the hyperparameters.

$$p(y/\alpha,\sigma^2) = \int p(y/w,\sigma^2)\,p(w/\alpha)dw = N(0,C) \tag{3.35}$$

Here the covariance is given by $C = \sigma^2 I + \varphi A^{-1}\varphi^T$

In (3.33) and (3.34) the only unknown variables are the hyperparameters α. The hyperparameters involved in the formulation are predicted using the structure of type II maximum likelihood [138].

$$\log p(y/\alpha,\sigma^2) = -1/2(M\log 2\pi + \log|C| + y^T C^{-1} y) \tag{3.36}$$

Logarithm is included in (3.36) to reduce computational complexity. Maximization of the logarithmic marginal likelihood in (3.36) over α leads to the most probable value α_{MP} which provides the maximum a posteriori (MAP) estimate of the weights.

The ambiguity about best possible values of the weights, given by (3.31), is used to convey ambiguity about the predictions made by the model, i.e., given an input x^*, the probability distribution of the consequent output y^* is known through its predictive distribution

$$p(y^*/x^*,\hat{\alpha},\hat{\sigma}^2) = \int p(y^*/x^*,w,\hat{\sigma}^2)\,p(w/y,\hat{\alpha},\hat{\sigma}^2)dw \tag{3.37}$$

which has the Gaussian form

$$p(y^*/x^*,\hat{\alpha},\hat{\sigma}^2) = N(Y^*,\sigma^{*2}) \tag{3.38}$$

The mean and variance of the predicted model are respectively,

$$Y^* = \varphi^T(x^*)\mu \qquad \text{and} \qquad \sigma^{*2} = \hat{\sigma}^2 + \varphi^T(x^*)\Sigma\varphi(x^*) \tag{3.39}$$

Maximizing the logarithmic marginal likelihood in (3.36) leads the optimal values of many of the hyperparameters α_m typically infinite , yielding a posterior distribution in (3.31) of the corresponding weights w_m that tends to be a delta function peaked to zero. Thus the corresponding weights are deleted from the model along with its accompanying

kernel function. Hence very few data points corresponding to nonzero weights build the RVM model and are called the relevance vectors. This results in very good sparseness of RVM model than SVM model. Thus the computation time for prediction using RVM model is reduced significantly.

3.3.3 Training of RVM network

The value of weight vector w and precision parameter β which is given by the inverse of the variance of the Gaussian conditional distribution by maximum likelihood, are found using the training dataset. Also α is the hyperparameter vector with dimension N+1. Considerably, each element of the hyperparameter vector α is committed independently with each element of the weight vector w.

Steps followed for training the RVM network are as follows:

1. Appropriate kernel function is chosen for the dataset and the basis function φ is designed by using the kernel function.
2. Suitable convergence criteria is selected for α and β For example, $\delta = \sum_{i=1} \alpha_i^{n+1} - \alpha_i^{n}$, such that re- estimation will halt when $\delta < \delta_{Thresh}$
3. Practically, infinity could be defined by choosing a very big value. A threshold value α_{Thresh} has been chosen for α_i. This α_{Thresh} value will be considered as infinity upon completing it.
4. Initialize α and β.
5. Estimate mean, $m = \beta \sum \phi^T t$ and covariance, $\sum = (A + \beta \phi^T \phi)^{-1}$. Here, A=diag($\alpha_i$)
6. Revise $\alpha_i = \frac{\gamma_i}{m_i^2}$ and $\beta = \frac{N - \sum_i \gamma_i}{|| t - \phi m ||^2}$. Here $\gamma_i = 1 - \alpha_i \sum_{ii}$, Here $\sum_{ii}$ is the i^{th} diagonal component of the posterior covariance $\sum$ and N is the total number of training data.
7. When $\alpha_i > \alpha_{Thresh}$, prune the α_i and matching basis functions.
8. Steps (5) to (7) are repeated, until the convergence criteria are satisfied.

Being using RVM network for process modeling, the architecture of the RVM network must be given significance. The delayed time input structure provides input for RVM network. The inputs to the RVM network includes signals u(n) and y(n-1), with their corresponding delayed version, u(n-1), ..., u(n-n_d), and y(n-2), ..., y(n-d_d). The parameters n_d and d_d represents the number of delay corresponding to the input and output of the

process to be modeled respectively. After the satisfaction of the convergence criteria sparseness is automatically achieved because while training, the posterior distributions of several of the weight values in the weight vector are strictly peaked around zero. The training vectors corresponding to the exceptional nonzero weights are known as the relevance vectors. Thus the output equation of the RVM network for an unknown time delayed input *x(n)* is

$$\hat{y}(x, w) = w^T \varphi(x(n)) \tag{3.40}$$

where,

$$x(n) = [u(n),.....u(n-n_d), y(n-1)....y(n-d_d)]$$

The structure given by (3.40) is sufficient to model any non-linear process. The plant output stored in vector form is the target dataset for the RVM. The regressor vector which is the input vector combining the target vector is used for training the plant model. The training stops when the stopping criteria either Maximum number of iterations or $\alpha_i = \alpha_{Thresh}$ is fulfilled. After training the RVM network, the trimmed basis function and the trimmed weight vector gives the model of the non-linear process.

Before using the trained RVM model for control purpose, validation of the network is required to check over fitting and generalization. In this test, an unseen input signal which is beyond the input signal used for training the RVM is used to compare the non-linear plant output and the trained RVM output. Hence, it is a time validation test with an error index equation (3.41) to quantify it and to check the accuracy of the approximated RVM model with N number of validation test data.

$$error_{index} = \sqrt{\frac{\sum_{k=1}^{N} (\hat{y}(k) - y(k))^2}{\sum_{k=1}^{N} y(k)^2}} \tag{3.41}$$

$\hat{y}(k)$ - RVM model prediction

$y(k)$ - actual plant output

N - number of validation test samples

3.3.4 Predictions for new data

The RVM network based predictive control algorithm can predict multistep ahead output of the process by using the process model for a random input from the present instance, *n*, to future instance, *n+k*. This is done by shifting equations (3.40) with respect to time by *k*, which becomes,

$$\hat{y}_n(n+k) = w^T \varphi(x(n+k)) \tag{3.42}$$

where,

$$x(n+k) = [u(n+k),.....u(n+k-n_d), \hat{y}(n+k-1)....\hat{y}(n+k-\min(k,d_d)), y(n-1),...y(n-\max((d_d-k),0))]$$

Thus prediction for new data point on the posterior distribution is obtained at convergence of the hyper parameter estimation procedure, trained on the maximizing values $\alpha_{MP}, \sigma^2_{MP}$. The predictive distribution for unknown data x_* can be determined by the following expression.

$$p\left(t_* / t, \alpha_{MP}, \sigma^2{}_{MP}\right) = \int p(t_* / w, \sigma^2{}_{MP})\, p(w / t, \alpha_{MP}, \sigma^2{}_{MP}) dw \tag{3.43}$$

Both the terms in the integral of equation (3.33) are Gaussian and hence the expression for predictive distribution can be expressed in terms of Gaussian as

$$p\left(t_* / t, \alpha_{MP}, \sigma^2{}_{MP}\right) = N(t_* / y_*, \sigma_*{}^2) \tag{3.44}$$

The predictive mean is by instinct $y(x_*; \mu)$. The predicted noise on the data and the noise due to prediction uncertainty together comprise the predictive variance.

Now, RVM is applied to some simple synthetic datasets and RVM regression is shown. In the RVM model, the Kernel function is the Gaussian function. Generally, the regression problems have their target dataset and in real time they will always be affected by some additive Gaussian noise component.

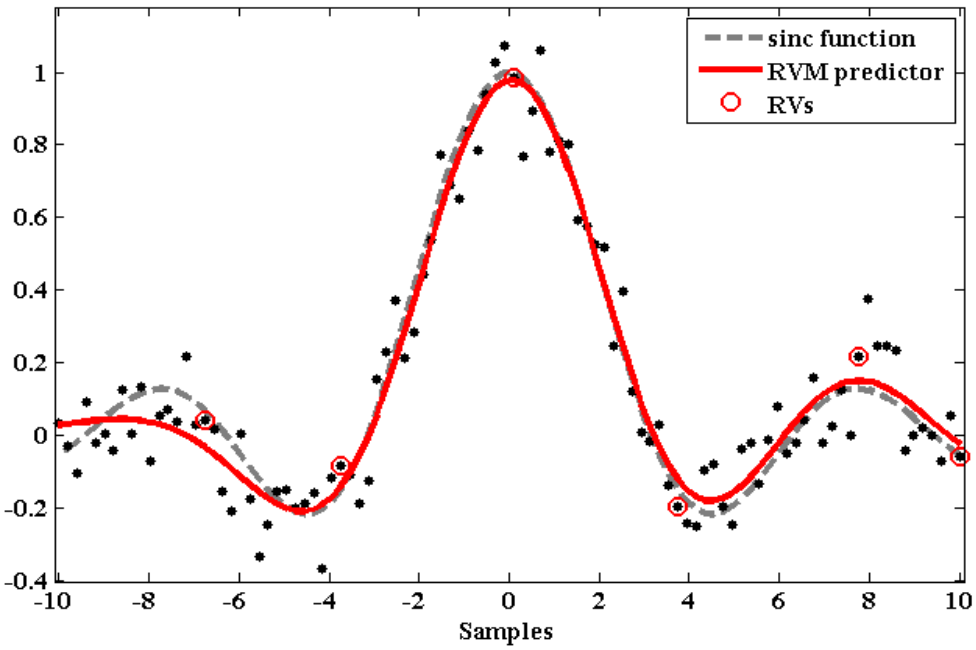

Fig. 3.5 RVM regression of 'sinc' function

In Fig. 3.5, the actual 'sinc' function is shown by dashed line and the noisy version of the 'sinc' function is shown by black dots. The objective is to predict the 'sinc' function with minimum amount of error and minimum number of relevance vectors. For this sinc function example 100 input vectors (100 black dots) have been used as dataset but the prediction is done by only 6 vectors which are known as relevant vector machine predictor. The RMSE value between the predicted output and the actual output is 0.042. Thus RVM has better prediction accuracy with much sparse model than SVM explained in previous section.

3.4 COMPARISON OF SVM AND RVM

Support Vector Machines differ from Relevance Vector Machines in different aspects as tabulated in Table 3.1.

Table 3.1. Difference between SVM and RVM

Sl. No.	Support Vector Machines	Relevance vector machines
1	SVM model is not as sparse as RVM model, as the number of support vectors necessity linearly develop with the number of training data.	RVM is much sparser than SVM, because practically while modeling in RVM several weights are peaked around zero, which in turn reduces the number of relevance vectors.
2	Predictions using SVM model are not probabilistic, instead deterministic.	Predictions using RVM model are probabilistic.
3	In SVM it is necessary to approximate a trade-off parameter 'C' by cross-validation procedure which is computationally expensive.	No such parameter to be estimated in RVM
4	In SVM the kernel function must satisfy Mercer's condition.	There is no such restriction on the basis function in the case of RVM.
5	In LS-SVM the regularization parameter γ and the kernel width parameter σ are the two parameters to be tuned to improve the generalization ability of predicted model	Kernel width σ is the only parameter to be tuned in Relevance vector regression (RVR) model.
6	The number of basis function used in SVM is more than RVM.	RVM model typically utilizes fewer kernel functions than SVM

3.4.1 Prediction accuracy of Support vector regression model

This section describes the comparative study of the prediction accuracy of LSSVM model and Neural Network model. The benchmark example under study is the Duffing's equation representing the relationship between a mass, damper and a stiffening spring as in equation (3.45)

$$\ddot{y}(t) + \dot{y}(t) + y(t) + y^3(t) = 2u(t) - \dot{u}(t) \tag{3.45}$$

A sequence of 300 samples is used to train the SVR model offline using the leave one out method. Leave one out method is one in which at each iteration, one leaves out one point, andfits the model on the other data points. The performance of the model is estimated based on the point left out. This procedure is repeated for each data point. Finally, all the different estimates of the performance are combined (default by computing the mean). The assumption is made that the input data is distributed independent and identically over the input space [134]. The two kernel parameters, kernel width (σ) and regularization parameter (γ) are chosen to be 1 and 100 respectively.

In the case of multilayer feed forward neural network, for offline training a sequence of 1000 samples are used and is done through Levenberg-Marquardt learning algorithm. After training the Support Vector Regression (SVR) model and Neural Network (NN) model, they are tested with some 100 unseen random inputs and there corresponding prediction errors are calculated. The comparative graph of prediction errors of SVR model and NN model are revealed in Fig 3.6.

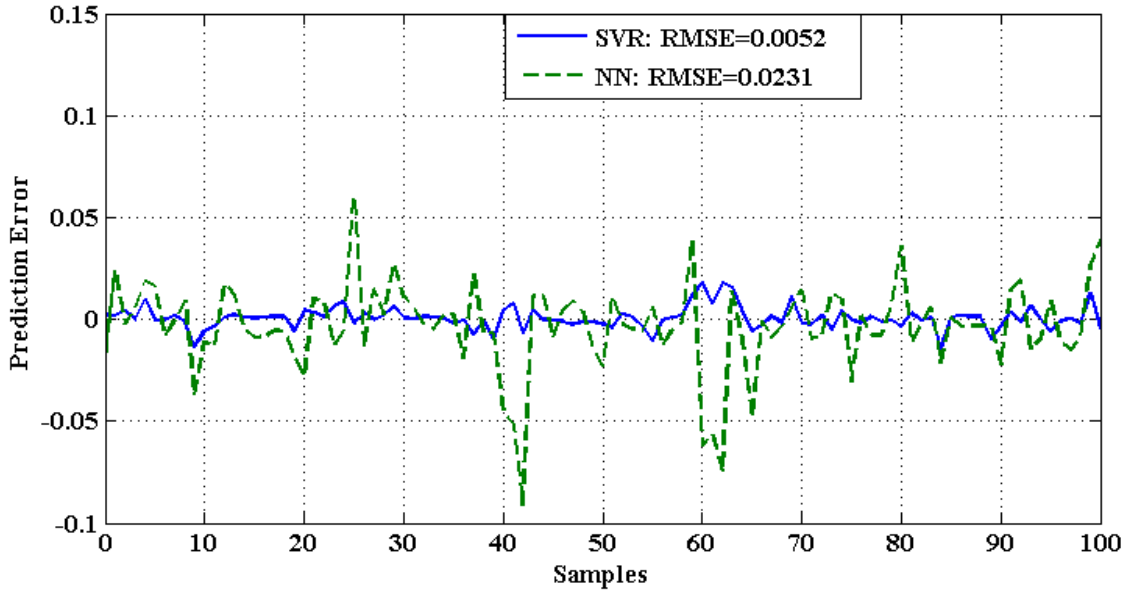

Fig. 3.6 Comparison of prediction errors of SVR and NN model

From Fig. 3.6 it is obvious that the prediction error for SVR model is very less when compared to NN model. The root mean square error (RMSE) value is 0.0052 and 0.0231 for SVR model and NN model respectively. This highlights the better prediction accuracy of SVR model.

3.4.2 Prediction accuracy of Relevance vector regression model

This section describes the comparative study of the prediction accuracy of RVR model and Neural Network model. The same benchmark example in the previous section is taken for analysis.

A sequence of 300 samples is used to train the sparse Bayesian RVR model offline. Hyper parameter estimation is carried out by Expectation Maximization (EM) updates on the objective function [22]. For this RVR model RBF kernel is used with the width parameter estimated automatically by the learning procedure [22] which improves generalization ability and reduces computational complexity of the training process. Thus, unlike in LS-SVM there is no necessity for computationally expensive determination of regularization parameter by cross validation technique. Also in the RVR model confidence intervals, likelihood values and posterior probabilities could be explicitly encoded easily.

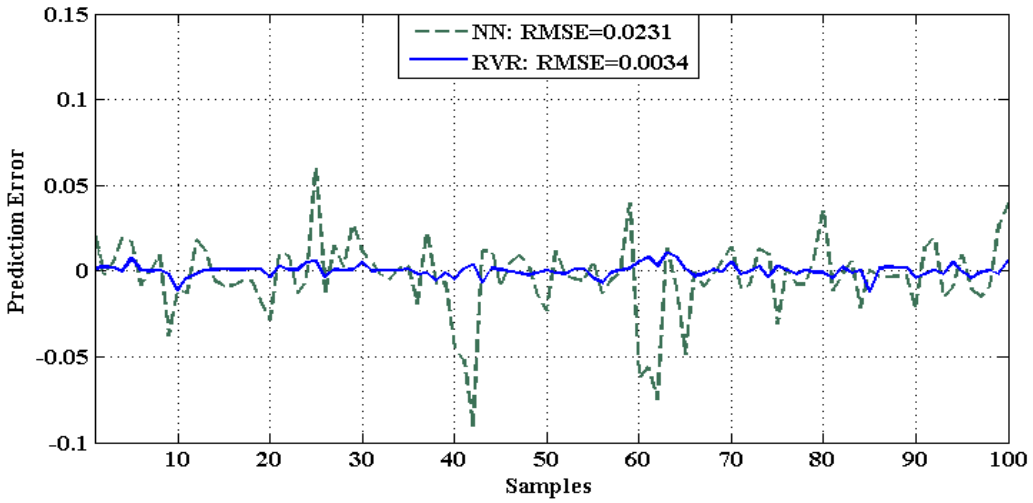

Fig. 3.7 Comparison of prediction errors of RVR and NN model

The comparative graph of prediction errors of RVR model and NN model are revealed in Fig 3.7. This figure make clear that the prediction error for RVR regression model is very less when compared to NN model. The root mean square error value is 0.0034 and 0.0231 for RVR model and NN model respectively. Also the prediction accuracy of RVR model is little better than SVR model. This highlights the better prediction accuracy of RVR model compared to SVR model and NN model.

Thus the better prediction accuracy of RVR and SVR model when compared to NN model are well understood from the above example.

3.4.3 Significance of accurate and sparse model in MPC

The model accuracy is a key task in order to provide an efficient and adequate control action. J. A. Rossiter [7] has explained the importance of model in MPC. In order to control a process very accurately a very accurate model is necessary. Thus model accuracy is the one which determines the controller's accuracy. A model is the one which provides useful predictions about the real world. If the models prediction is accurate then the control signal generated to control the actual process will also be accurate. So an accurate model is the heart of MPC.

In MPC as prediction and optimization are done at each sampling instant, care should be taken to decrease the time consumption for both prediction and optimization. A sparse model could predict much faster than a dense model [142]. Accordingly RVR model which is much sparser than SVR model and SVR model which is much sparser than NN model performs suitably in MPC applications.

Table 3.2. Significance of model accuracy and sparseness in MPC

Model	Number of samples	Prediction error RMSE	Number of support vectors/ Relevance vectors	Relative prediction time
RVM	100	0.0034	5	1
SVM	100	0.0052	41	2.3
Neural network	100	0.0231	-	4.9

The same benchmark example in the previous section is considered and the following data are tabulated in Table 3.2. The relative prediction time of RVR model is much less than SVR model and NN model which elucidate the importance of model sparseness. The RVR model is made much sparser than SVR model by the usage of less number of relevance vectors than support vectors.

3.5 NEURO-FUZZY TECHNIQUES

Neuro-fuzzy which was proposed by J.S.Jang [73] is the combinations of ANN and fuzzy logic. Fuzzy logic solves the complex nonlinear problems by utilizing the experience

or expert's knowledge described as the fuzzy rule base [143]. Fuzzy controllers which omit the requirement of exact mathematical model of the process to be controlled are well appropriate to nonlinear time-variant systems [144]. ANN and fuzzy logic techniques are effectively applied in motor drive systems [145, 146]. Neuro-fuzzy hybridization technique results in a better intelligent system that works together with the human-like reasoning style of fuzzy systems which incorporates fuzzy sets IF -THEN fuzzy rules and the learning structure of neural networks. The neuro-fuzzy technique could be divided under two areas: linguistic fuzzy modeling which focuses on interpretability, mainly the Mamdani model; and precise fuzzy modeling which focuses on accuracy, mainly the Takagi-Sugeno-Kang model.

3.5.1 Conventional ANFIS

ANFIS is a neuro-fuzzy technique, which is based on hybrid learning algorithm. It maps first order Sugeno fuzzy inference system in multilayer feed-forward adaptive neural network to enhance performance such as fast and accurate learning by fine tuning of membership function parameters and by analysing both linguistic and numerical knowledge. The first order Sugeno fuzzy model, its inference mechanism and defuzzification process is shown in Fig. 3.8. The typical fuzzy If-Then rule set is used to describe ANFIS architecture. It is expressed as follow:

Rule 1: If x is A_1 and y is B_1, then $f_1 = p_1\, x + q_1\, y + r_1$

Rule 2: If x is A_2 and y is B_2, then $f_2 = p_1\, x + q_2\, y + r_2$

Here, x and y are the crisp inputs, A_i, B_i are linguistic variables. The five layer ANFIS architecture explained by Jang [63] is shown in Fig. 3.9. The purpose of each layer is described as follow:

1) Layer-1: Every node i in this layer represents fuzzy membership function as node function with an adaptive parameters.

$$O_i^1 = \mu_{A_i}(x), \qquad i = 1,2 \tag{3.46}$$

where x is input value, O_i^1 is membership value of fuzzy variable A_i. a_i, b_i, c_i are the adaptive parameters commonly known as premise parameters.

2) Layer-2: Each node in this layer is fixed node which acts like product operation as in Sugeno fuzzy model.

$$O_i^2 = w_i = \mu_{A_i}(x) \times \mu_{B_i}(y), \qquad i = 1,2 \tag{3.47}$$

Fig. 3.8 First order Sugeno Fuzzy inference mechanism

3) Layer-3: layer contains fixed nodes, which calculates normalized firing strength, $\overline{W}_i$ as follows

$$\overline{w}_i = \frac{w_i}{w_1 + w_2}, \qquad i = 1,2 \tag{3.48}$$

4) Layer-4: Each node i, in this layer is an adaptive node with a node function given as

$$O_i^4 = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i) \tag{3.49}$$

where $\overline{W_i}$ is the output of layer-3 and $\{p_i, q_i, r_i\}$ are adaptive consequent parameters.

5) Layer-5: It is fixed single node that computes overall output as summation of all incoming signals from layer-4.

$$O_i^5 = \sum_i w f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{3.50}$$

The premise and consequent parameters are updated to minimise error by various learning algorithms. The most commonly used algorithm is hybrid learning algorithm [73]. This algorithm combines the gradient based method (i.e. back propagation) and least square error (LSE) to identify parameters. The hybrid learning algorithm (HLA) consists of two passes explained as follows:

Forward pass: In the forward pass of the HLA, node outputs pass forward up to layer-4 by assuming some premise parameters and the consequent parameters are recognized by the least-square error method. As the premise parameter values are fixed, the linear combination of the consequent parameters provides the overall output,

$$f = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_1}{w_1 + w_2} f_2$$

$$= \overline{w_1} f_1 + \overline{w_2} f_2$$

$$= (\overline{w_1} x) p_1 + (\overline{w_1} y) q_1 + (\overline{w_1}) r_1$$

$$+ (\overline{w_2} x) p_2 + (\overline{w_2} y) q_2 + (\overline{w_2}) r_2 \quad (3.51)$$

which is linear in the consequent parameters p_1, q_1, r_1, p_2, q_2 and r_2

$$f = XZ \quad (3.52)$$

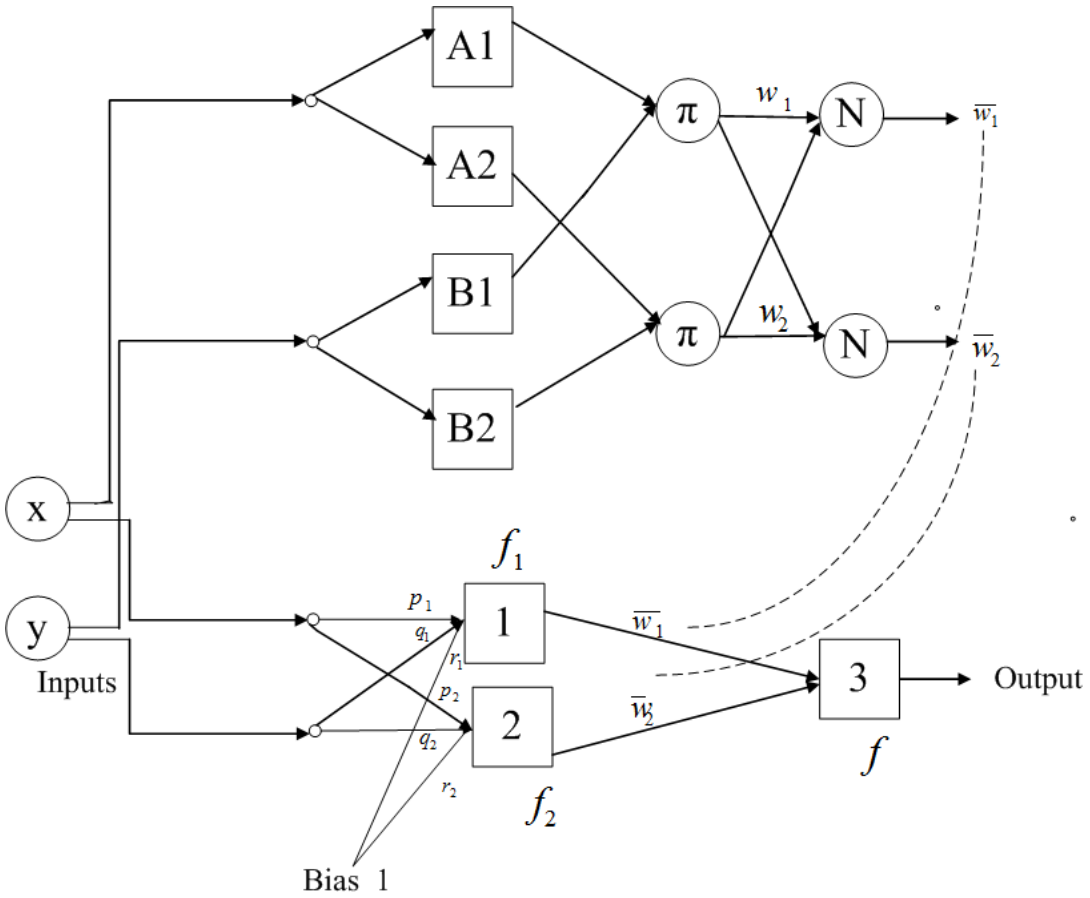

Fig. 3.9 ANFIS architecture

If X matrix is invertible then,

$$Z = X^{-1} f \quad (3.53)$$

Otherwise solution of Z could be obtained using pseudo-inverse method.

$$Z = (X^T X)^{-1} X^T f \quad (3.54)$$

b) Backward pass: In backward pass, the error signals propagate backward and the premise parameters are renewed with the help of gradient descent method by keeping consequent parameters fixed. Parameter updating rule is given as

$$a_{ij}(t+1) = a_{ij}(t) - \eta.\frac{\partial E}{\partial a_{ij}} \tag{3.55}$$

where η is the learning rate for parameter a_{ij}, the gradient is obtained using chain rule as

$$\frac{\partial E}{\partial a_{ij}} = e.1.\frac{(p_i x + q_i y + r_i) - f}{\sum_{i=1}^{n} w_i}.\frac{w_i}{\mu_{A_{ij}}}.\frac{\partial \mu_{A_{ij}}}{\partial a_{ij}} \tag{3.56}$$

$$\frac{\partial E}{\partial b_{ij}} = e.1.\frac{(p_i x + q_i y + r_i) - f}{\sum_{i=1}^{n} w_i}.\frac{w_i}{\mu_{B_{ij}}}.\frac{\partial \mu_{B_{ij}}}{\partial b_{ij}} \tag{3.57}$$

By using forward pass and backward pass alternately, the ANFIS parameters have been trained. But the training time requirement of hybrid learning algorithm is much lengthy. This drawback could be overcome by the novel Extreme ANFIS learning algorithm.

3.5.2 Extreme ANFIS Learning Algorithm

The advantages of ANFIS architecture and its approximating capabilities with great accuracy have been proved already by Jang in 1993 [73]. The architecture and Hybrid Learning Algorithm introduced by Jang is explained in previous Section. HLA uses both the forward pass and backward pass, which is a combination of least square error and back propagation based on gradient descent. As it uses gradient based method, it has certain drawbacks such as the calculation of gradient is possible with differentiable membership functions only, over fitting, and an iterative method hence time consuming.

The proposed algorithm, called Extreme ANFIS learning algorithm is simple and derivative-less algorithm which eliminate drawbacks of gradient based Hybrid Learning Algorithm. The proposed algorithm can be summarized as follows:

Algorithm: Consider a particular training data set scattered over the possible input range are available for given regression or modeling problem as:

$[I_1 \;\; I_2 \;\; I_3 \ldots\ldots\ldots\ldots \;\; I_n; \; f]$

where I_i, i=1, 2… n are inputs, and f is corresponding output.

Step 1: Calculate range of every input to get the universe of discourse for input membership functions as,

$$range_i = max\{I_i\} - min\{I_i\} \tag{3.58}$$

where i = 1, 2… n is number of input.

Step 2: Decide shape of membership function and number of membership functions which represent linguistic partitions of universe of discourse. Bell shaped membership function is commonly used because of its advantages over other membership functions such as smoothness in change in membership grade which is a drawback of triangular and trapezoidal membership functions, also it provides flexibility in core of membership function which is not possible in Gaussian type membership function. The mathematical representation of bell shape membership function is given as,

$$\mu_{ij} = \frac{1}{1 + \left|\frac{I_i - c_j}{a_j}\right|^{2b_j}} \tag{3.59}$$

where μ_{ij} represents membership grade of i^{th} input and j^{th} membership function, a_j, b_j, c_j are position and shape deciding parameters. The c_j represents the centre of j^{th} membership function, a_j decides half width of membership function, $b_j/2a_j$ represents slope at membership grade μ_{ij} = 0.5. The significance of parameters is shown in Fig. 3.10.

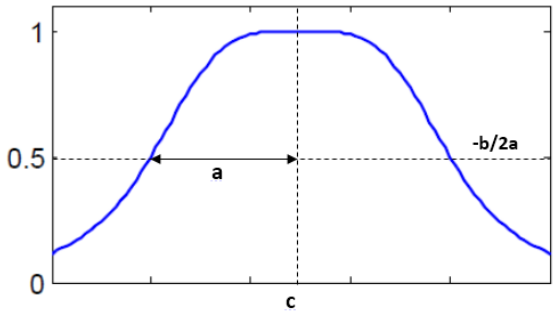

Fig 3.10 Bell shape membership function parameters

Step 3: The premise parameters (a_j, b_j, c_j) values are generated randomly with some constraints on ranges of those parameters. These ranges depend on the number of membership functions used and size of universe of discourse.

Consider there are m uniformly distributed membership functions with parameters (a_j^*, b_j^*, c_j^*) in universe of discourse. The procedure to select random parameters is described below:

Selection of a_j: The parameter a_j decides the width of membership function. The default value of parameter in uniformly distributed membership functions is expressed as,

$$a_j^* = \frac{range_i}{2m-2} \tag{3.60}$$

The range of selection of random value a_j is given as,

$$\frac{a_j^*}{2} \leq a_j \leq \frac{3a_j^*}{2} \tag{3.61}$$

Selection of b_j: The parameter b_j with the help of a_j gives slope as $b_j/2a_j$. The default value of b_j in uniformly distributed membership functions is 2. The slight change in this value significantly changes the slope hence its range is limited within 1.9 to 2.1.

Selection of c_j: The range for random value for center (c_j) of membership function is decided such that one center should not cross the center of consecutive membership function. The range of center value (c_j) selection is shown in equation (3.62)

$$\left(c_j^* - \frac{d_{cc}}{2}\right) < c_j < \left(c_j^* - \frac{d_{cc}}{2}\right) \tag{3.62}$$

where

c_j^* is center of uniformly distributed membership function,

d_{cc} is distance between two consecutive centers of uniformly distributed membership.

Step 4: Once the randomly generated membership functions are available, the node output up to fourth layer can be obtained easily. Now the final output *f* becomes a simple linear combination of consequent parameters as shown in equation (3.63)

$$f = \sum_{k=1}^{m^n} \bar{W}_k \left(\sum_{i=1}^{n} R_{ki} I_i + Q_k \right) \tag{3.63}$$

where *k* represents the number of rules, *m* specifies the number of membership functions, *n* specifies the number of inputs, m^n represents maximum number of rules, *i* represents number of inputs, I_i is a value of i^{th} input, R_{ki} and Q_k are consequent parameters corresponding to k^{th} rule and i^{th} input.

Consider there are *p* numbers of training data pairs, then linear matrix of *p* equations are represented as,

$$F_{p\times 1} = \beta_{p\times m^n\,(n+1)}\, U_{m^n(n+1)\times 1} \tag{3.64}$$

where F is output matrix, β is weighted input parameter matrix, U is unknown consequent parameter matrix of shown dimension. It can be solved using least square error (LSE) method as has been explained in forward pass of Hybrid Learning Algorithm.

Step 5: Run algorithm after step 2 for 50-70 times and find out the root mean square error in each epoch. And finally generate the Sugeno type FIS model using the obtained premise and consequent parameters for least RMSE.

3.5.3 Comparison of conventional ANFIS and Extreme ANFIS

The proposed algorithm is implemented in Matlab and tested for its time efficiency and accuracy upon some benchmark problems. The simulation results and performance evaluation is discussed in next section.

The performance analysis is conducted on the basis of time required to learn parameters from training data, training error and testing error of proposed algorithm. The results obtained have been compared with conventional ANFIS results.

Example 1: consider a single input single output nonlinear sinc function (3.65)

$$\begin{aligned} f(x) &= \sin(x)/x, \quad x \neq 0 \\ &= 1 \quad x = 0 \end{aligned} \tag{3.65}$$

100 training samples are created where the input x is uniformly distributed on the interval (-10,10). A uniform noise distributed in [-0.1, 0.1] is added to training samples. 50 noise free samples are used for testing the algorithms.

Table 3.3 Performance analysis of Example 1

Algorithm	Number of membership functions	Time (s)	Training error (RMSE)	Testing error (RMSE)
Conventional ANFIS	4	0.57	0.012	0.192
Extreme ANFIS		0.0021	0.018	0.140
Conventional ANFIS	7	0.92	0.004	0.124
Extreme ANFIS		0.001	0.0062	0.104
Conventional ANFIS	11	8.6	0.0018	0.051
Extreme ANFIS		0.004	0.0021	0.048

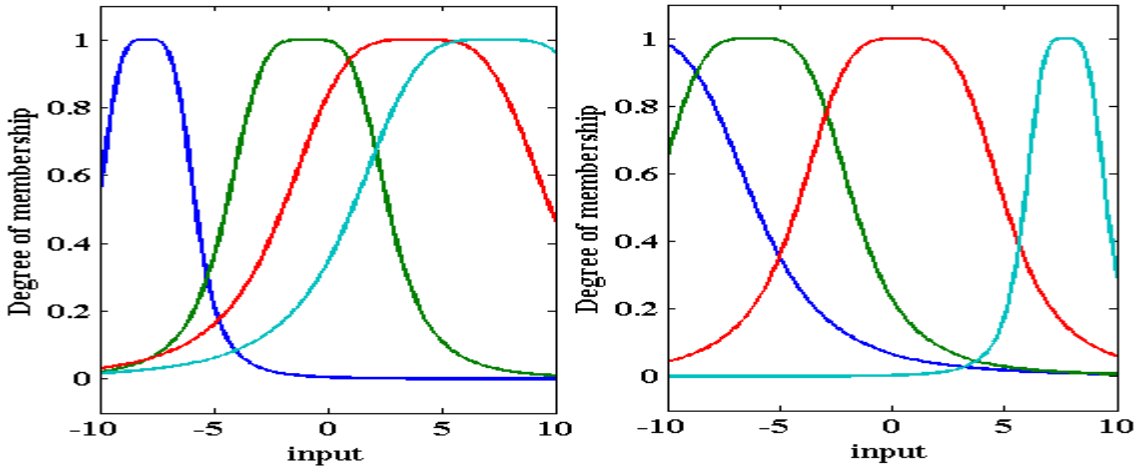

Fig 3.11 Final membership functions after learning with (a) Extreme ANFIS (b) Conventional ANFIS

The final membership functions after learning using conventional ANFIS algorithm and Extreme ANFIS algorithm are given in Fig. 3.11. The learning time, training error and testing error are listed in Table 3.3. It is clearly observed that if the number of membership functions increases, the model accuracy also increases with less training time. But in case of conventional ANFIS algorithm time to learn parameters also increases. Thus the drawback of large training time is totally eliminated by Extreme ANFIS learning algorithm without affecting accuracy and generalization of conventional method.

In the subsequent section, results of two benchmark problems from previous work of Jang are compared [73, 75].

Example 2: consider two input nonlinear function given in equation (3.66)

$$z = \frac{\sin(x)}{x} . \frac{\sin(y)}{y} \tag{3.66}$$

Consider the inputs x and y varies in range of [-10, 10] × [-10, 10]. Equally spaced 121 training data pairs and 16 rules with four membership functions are used for adapting parameters of ANFIS using both conventional Hybrid Learning algorithm and proposed Extreme ANFIS learning algorithm. And equally spaced 100 testing data are used to check generalization. The final membership functions after learning using conventional ANFIS algorithm and Extreme ANFIS algorithm are given in Fig. 3.12 and Fig.3.13.

Table 3.4 Performance analysis of Example 2

Algorithm	Number of membership functions	Time (s)	Training error (RMSE)	Testing error (RMSE)
Conventional ANFIS	4	0.814	0.0352	0.0862
Extreme ANFIS		0.479	0.0418	0.0438
Conventional ANFIS	7	3.933	0.0004	0.0304
Extreme ANFIS		1.077	0.0066	0.0411
Conventional ANFIS	11	23.4315	9.8×10^{-8}	0.0688
Extreme ANFIS		1.718	1.35×10^{-14}	0.0625

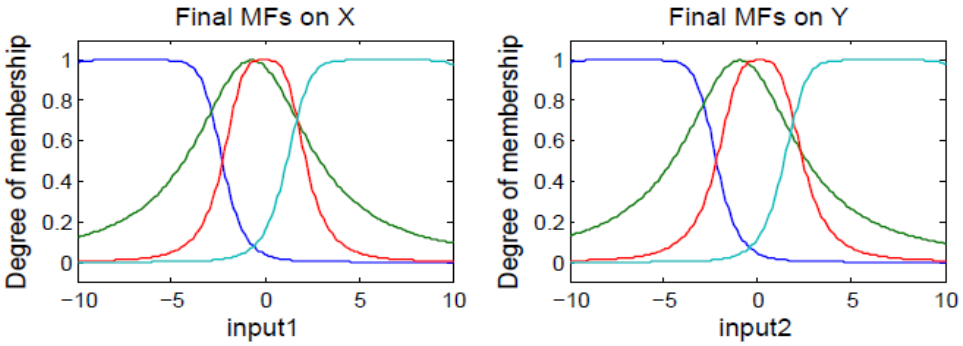

Fig. 3.12 Final membership functions after learning with Conventional ANFIS

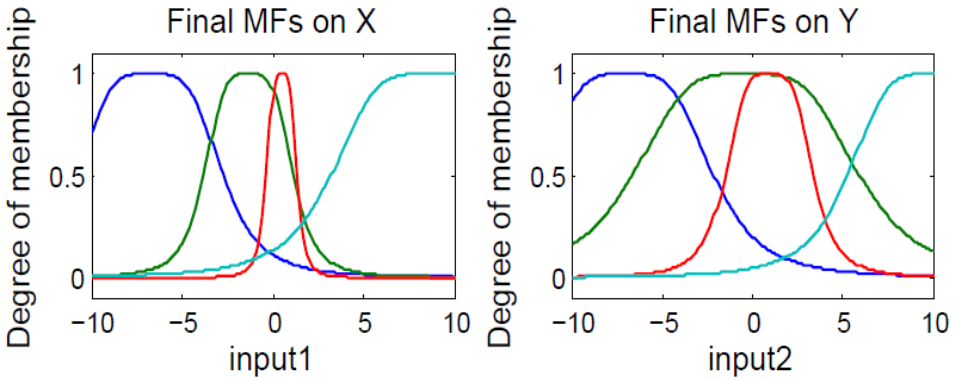

Fig. 3.13 Final membership functions after learning with Extreme ANFIS algorithm

Example 3: Modeling of nonlinear function with three inputs using ANFIS. The function is given as,

$$O = (1 + x^{0.5} + y^{-1} + z^{-1.5})^2 \tag{3.67}$$

where *x*, *y* and *z* are inputs with ranges [1,6]×[1,6]×[1,6]. 216 training data are used to train ANFIS. The time efficiency and training error is checked by using training data.

Generalization of obtained ANFIS model is analyzed for 125 testing data in range [1.5, 5.5] × [1.5, 5.5] × [1.5, 5.5]. The overall performance analysis for different number of membership functions is listed in Table 3.5.

From observed results it is examined that, as the number of membership function increases the generalization ability of conventional gradient based algorithm is affected badly due to over-fitting. But in the proposed algorithm generalization is much better by the accurate model. Learning times and errors listed in tables are obtained by taking average of ten to fifteen trails.

Table 3.5 Performance analysis of Example 3

Algorithm	Number of membership functions	Time (s)	Training error (RMSE)	Testing error (RMSE)
Conventional ANFIS	2	0.588	0.0254	0.3128
Extreme ANFIS		0.389	0.0942	0.4057
Conventional ANFIS	4	18.671	3.5×10^{-4}	2.1263
Extreme ANFIS		2.503	0.0023	0.4825
Conventional ANFIS	5	96.06	7.9×10^{-5}	4.5413
Extreme ANFIS		4.073	1.78×10^{-4}	1.4510

3.5.4 Prediction accuracy of Extreme ANFIS model

This section describes the prediction accuracy of Extreme ANFIS model for a benchmark example, the Duffing's equation representing the relationship between a mass, damper and a stiffening spring as in equation (3.45) is considered for study.

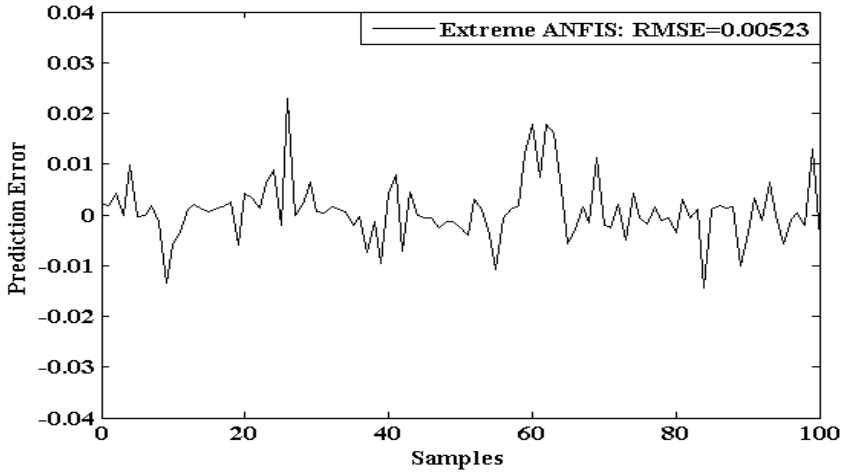

Fig 3.14 Prediction accuracy of Extreme ANFIS model

A sequence of 300 samples and 10 rules with 10 membership functions are used for adapting parameters of ANFIS using proposed Extreme ANFIS learning algorithm. And the same 100 testing data used to analyze prediction errors in SVM and RVM in previous sections are used to check generalization of developed Extreme ANFIS model. The prediction accuracy of Extreme ANFIS based MPC is comparable with SVM and RVM. This highlights the good generalization capability of the proposed novel Extreme ANFIS model.

3.6 CONCLUSION

The model predictive control using Support vector machines, Relevance vector machines and neuro-fuzzy techniques are described in this chapter. The simulation results tabulating the prediction error and prediction time for a benchmark example explains the significance of accurate and sparse model in MPC applications. Also the proposed novel Extreme ANFIS algorithm performs better than conventional ANFIS algorithm with good generalization capability as SVM and RVM with very less training time.

Chapter 4
NONLINEAR MODEL PREDICTIVE CONTROL OF A SINGLE INPUT SINGLE OUTPUT PROCESS

This chapter describes the control of nonlinear CSTR SISO process using Relevance Vector Machines (RVM) regression model, Support vector regression (SVR) model, neuro-fuzzy techniques and NN model based MPC's. The optimization problem in the above control algorithm is made faster by particle swarm optimization with controllable random exploration velocity method of optimization. The simulation results comparing MPC using deterministic sparse kernel learning technique, probabilistic sparse kernel technique, neuro-fuzzy techniques and NN technique are shown.

4.1 INTRODUCTION

Before selecting a control strategy many requirements have to be considered. The implementation cost, maintenance cost, computational cost, operation transparency are the important criterion considered. Model predictive controller is one among the controllers which satisfies the above requirements opted for practical implementation. Model predictive controller does prediction and optimization at each sampling instant as discussed in the previous chapters. Hence, system identification technique and optimization technique are the norms which decide the quality of MPC. This chapter applies novel system identification techniques and optimization algorithm to make MPC the best controller suitable for practical application.

A nonlinear model predictive control strategy which utilizes all the machine learning techniques discussed in previous chapters and PSO-CREV optimization algorithm is applied to a single input single output process. A CSTR is the SISO process occupied for analysis. An accurate reliable nonlinear model is first identified by the above mentioned machine learning techniques and then the optimization of control sequence is speeded up by PSO-CREV. Additional stochastic behaviour in PSO-CREV is omitted for faster convergence of nonlinear optimization. An improved system performance is guaranteed by an accurate model and an efficient and fast optimization algorithm. Performance comparisons of MPC's using probabilistic sparse kernel learning technique called RVM's regression model, deterministic sparse kernel learning technique called LS-SVM regression model, a proposed novel neuro-fuzzy based (Extreme adaptive neuro-fuzzy inference system (ANFIS)) model and neural network based model are done on a CSTR process. Relevance vector regression shows improved tracking performance with very less computation time which is much essential for real time control. To the author's best

knowledge, combinations of RVM model and MPC approach are not reported in literatures.

A novel method of optimization, particle swarm optimization with controllable random exploration velocity developed by Xin Chen et al. [91, 92], which is significant for its computational efficiency and improved performance, is used in this chapter. Thus nonlinear model predictive controller combining Extreme ANFIS/ relevance vector regression/ support vector regression/ neural network model and particle swarm optimization with controllable random exploration velocity merges the advantage of accurate prediction and less computational effort.

This section is followed by simulation results of SISO process comparing MPC using deterministic sparse kernel learning technique, probabilistic sparse kernel technique, neuro-fuzzy techniques and NN technique.

4.2 THEORY BEHIND MPC FOR A SISO PROCESS

The structure of model predictive control of a SISO system is shown in Fig. 4.1.

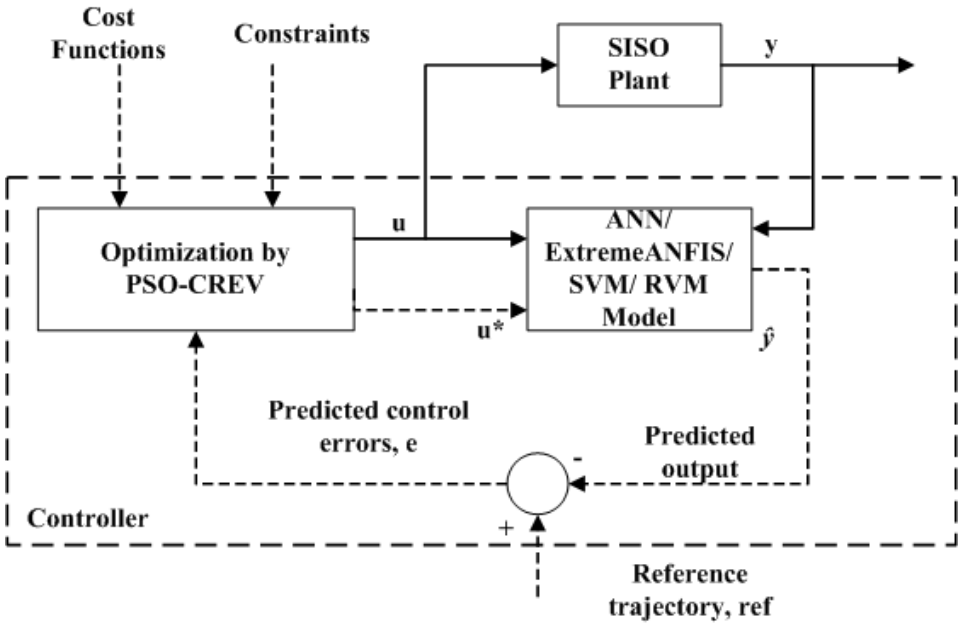

Fig. 4.1 Structure of nonlinear model predictive control of SISO system

MPC uses the present dynamic state of the plant, the past process measurements, its models, and the plants target variables and constraints to evaluate future deviations in the manipulated variables. These deviations are evaluated to keep the process output near to the target while constraints imposed on the input variables and output variables are considered. MPC generally applies the first alteration in the manipulated variable to the

real plant, and uses the remaining alterations for the prediction of process output. The non-linear model responsible for accurate prediction can be easily generated by ANN, Extreme ANFIS, SVM or RVM models. Key terms behind the theory of MPC are:

1. Moving horizon window (MHW): the time dependent window which starts from an arbitrary time to + where the length of the window remains fixed.
2. Prediction horizon: means the distance how far ahead the future output is to be predicted. The value of this parameter equals to the length of the moving horizon window, .
3. Receding horizon control: Within the MHW, the optimal trajectory of the future manipulated variable is fully defined but only the initial step of the calculated manipulated variable signal is applied to the real process.
4. The designed model must cover the entire dynamics of the process. For predictive control, designing a good model of the process is very important because the model is the one responsible for predicting the precise future process outputs.
5. To implement the objective, a condition is required which can take the best decision. The objective function is known as the performance function and it is nothing but an error function between the actual and desired responses. Hence, the best control action can be generated by minimizing the performance function within the optimization window.
6. Minimization of performance function could be carried out by any optimization algorithm and in this chapter a novel method of optimization, particle swarm optimization with controllable random exploration velocity is used for accurate and fast optimization.

Thus the principles of model predictive control consists of

- Dynamic model of the plant.
- Past control moves.
- Cost function over the receding prediction horizon.
- Prediction horizon.
- Control horizon.

4.2.1 Different machine learning techniques for MPC's

The basic structure of different machine learning techniques based nonlinear model predictive controller of a SISO process is shown in Fig. 4.1. This structure includes three important blocks, actual plant to be controlled with output *y(k)*. Then the ANN/ Extreme

ANFIS / SVM/ RVM model of the actual plant to be controlled with estimated output $\hat{y}(k)= [\hat{y}(k+1) ,... \hat{y}(k+Np)]$ here, N_p refers the MPC's prediction horizon which dictates how far we wish the future to be predicted for. Next is the optimization block which provides the optimized control signal $u(k)=[u_1(k),...u_{Nu}(k)]$ where N_u refers to the control horizon of MPC which dictates number of control moves used to attain the upcoming control trail, subjected to the specified constraints that is required for the plant to achieve the desired trajectory $ref(k)=[ref_1(k)_1 ....ref_{Np}(k)]$. Here k stands for the current sampling instant.

Thus at each sampling instant a sequence of manipulated variable $u(k)$ is calculated to minimize the deviation of the predicted output of approximated model from the preferred reference trajectory over the specified prediction horizon N_p as shown in (4.7). The control horizon N_u decides the number of steps of manipulated variable in the sequence. This procedure is repeated at every sampling moment.

4.2.2 Performance index formulation

For a single input single output nonlinear process the predicted output of NN/ Extreme ANFIS/ SVM/ RVM model is a function of past process outputs, $Y(k)=[y(k).....y(k-n_y+1)]$ and past process inputs, $U(k-1)=[u(k-1)...u(k-n_u+1)]$. The numeral value of delayed controlled variables and delayed manipulated variables depends on the corresponding process orders n_u and n_y respectively.

Thus a single step ahead prediction of a SISO nonlinear process could be illustrated by the subsequent discrete time model,

$$\hat{y}(k+1)=f[y(k),..y(k-n_y+1),u(k),..u(k-n_u+1)] \tag{4.1}$$

where k is the discrete time index

The above equation is rewritten as

$$\hat{y}(k+1)=f[Y(k),u(k),U(k-1)] \tag{4.2}$$

Here, $Y(k)$ and $U(k-1)$ represents the vectors holding past controlled variables and past manipulated variables respectively. Thus after system identification using the regression data set the single step ahead prediction for ANN model is,

$$\hat{y}(k+1)=\sum_{j=1}^{hid}\left\{w_j f_j\left(net_j(k+1)\right)\right\}+b \tag{4.3}$$

The one step ahead prediction for Extreme ANFIS model is,

$$\hat{y}(k+1) = \sum_{k=1}^{m^n} \overline{W}_k \left(\sum_{i=1}^{n} R_{ki\ I_i} + Q_k \right) \tag{4.4}$$

The one step ahead prediction for SVM model is,

$$\hat{y}_j(k+1) = \sum_{i=1}^{M} \alpha_i K(x_i, x) + b \tag{4.5}$$

The one step ahead prediction for RVM model could is,

$$\hat{y}(k+1) = \sum_{m=0}^{M} w_m \varphi_m(x) = w^T \varphi \tag{4.6}$$

where M is the subset of training samples.

Accordingly, the performance index to be minimized to achieve the optimal control sequence *u(k)* can be obtained as shown below,

$$J[u(k)] = \sum_{j=N_1}^{N_2} \left[ref(k+j) - \hat{y}(k+j) \right]^2 + \sum_{j=1}^{N_u} \lambda \left[\Delta u(n+j) \right]^2 \tag{4.7}$$

In the performance index formulated in Equation (4.7) $\hat{y}$depends on the kernel function in case of SVM and RVM model, which in turn is a function of manipulated variable u, which is optimized and applied to the actual plant to diminish the deviation between the desired value and controlled variable.

N_1	-	minimum value of prediction horizon
N_2	-	maximum value of prediction horizon
N_u	-	value of control horizon
k	-	current sampling instant
ref(.)	-	reference trajectory
λ	-	control input weighting factor
$\hat{y}(.)$	-	predicted output of ANN/ Extreme ANFIS / SVM/ RVM
Δu	-	control input change defined as *u(k+j)-u(k+j-1)*

The predictive controller presented in this chapter is a single step ahead predictive controller, which does one step ahead prediction at each sampling instant. Since the process under simulation is a low order process and as the SVR model and RVR model developed are accurate, a single step ahead prediction is sufficient to provide satisfactory tracking performance.

The MPC parameters for the best control of CSTR process are given below.

Prediction horizon=1

Control horizon=1

Penalty factor on differenced control signal =0.5

Control input constraint $0 \leq u_1(t) \leq 4$

4.3 CONVENTIONAL PSO AND NEED FOR PSO-CREV ALGORITHM

PSO is an optimization algorithm supported by population, which reproduces the communal behavior of group of particles. The victory of majority of evolutionary optimization algorithms are realized by balancing two objectives, exploration and exploitation. Exploration refers to diversification which satisfies the goal of searching the whole solution domain in order to provide a consistent estimation of the global optimum. On the other hand, the term exploitation refers to intensification which intensifies the search attempt around the best solutions [147]. From Equation (2.5) and Equation (2.6) in chapter 2 it is understood that the strength of exploration performance is merely calculated by the degrading speed of $(P_{id}^{(t)} - X_{id}^{(t)})$ and $(P_{gd}^{(t)} - X_{id}^{(t)})$ as r_1 and r_2 are supplemented as relational coefficients to $(P_{id}^{(t)} - X_{id}^{(t)})$and $(P_{gd}^{(t)} - X_{id}^{(t)})$ respectively. These random variables r_1 and r_2 cannot be tuned without restraint. Hence if a particle converges to a local minimal solution, the algorithm may not have the capability to neglect it and hence the strength of exploration behavior of the conventional PSO algorithm needs improvement. This task of improving the exploration strength is achieved in a modified novel algorithm PSO-CREV.

The introduction of the terms ε, ξ and the positive coefficient α makes PSO-CREV algorithm different from conventional PSO. In the modified PSO-CREV algorithm $\xi(n)$ is introduced into velocity updating equation and hence convergence is not guaranteed since it is a disturbance to the system. Hence 'ε(n)'which reaches zero as n→infinity is included to repress $\xi(n)$, but again the reduced system seems to be non-convergent. Hence a new term α is introduced to make the system convergent. Thus introduction of these terms guarantees stronger exploration capability and faster convergence speed.

The intensity of exploration capability of conventional PSO was improved significantly by Xin Chen et al. [91,92], after incorporating some modifications in the position and velocity equations as shown in Equation (4.8) and Equation (4.9) respectively.

$$v_{id}^{(t+1)} = \varepsilon^{(t)}[v_{id}^{(t)} + c_1 r_1 (P_{id}^{(t)} - X_{id}^{(t)}) + c_2 r_2 (P_{gd}^{(t)} - X_{id}^{(t)}) + \xi_{id}^{(t)}] \quad (4.8)$$

$$x_{id}^{(t+1)} = \alpha x_{id}^{(t)} + v_{id}^{(t+1)} + \frac{1-\alpha}{\phi_{id}^{(t)}}[c_1 r_1 P_{id}^{(t)} + c_2 r_2 P_{gd}^{(t)}] \tag{4.9}$$

$\xi_{id}^{(t)}$ - Bounded random variable with continuous uniform distribution, $\xi(n) = w(n)\bar{\xi}(n)$

$$w(n) = \begin{cases} 1, & n < \frac{1}{4}N_b \\ \lambda_1 w(n-1), & n \geq \frac{1}{4}N_b, n < \frac{3}{4}N_b \\ \lambda_2 w(n-1), & n \geq \frac{3}{4}N_b \end{cases}$$

N_b - Total number of iterations

λ_1, λ_2 - Positive constants less than one which makes the random velocity $\xi(n)$ a decreasing one.

$\bar{\xi}(n)$ - Stochastic velocity with invariable value range and zero expectant.

$\varepsilon^{(t)}$ - tends to zero as t increases, and $\sum_{t=1}^{\infty} \varepsilon(n) = \infty$

α - ranges between 0 and 1.

The introduction of the terms $\varepsilon(n)$, $\xi$and the positive coefficient α makes the algorithm different from the conventional PSO. Introduction of these terms guarantees stronger exploration capability and faster convergence speed. $\varepsilon(n)$ is different from inertia constant as it is operating on all the three components of velocity update equation. The additional stochastic term ξ increases the exploration capability of the algorithm but decreases the convergence speed. A decreasing $\varepsilon(n)$ which tends to zero as $n \to \infty$ in the velocity update equation and a positive coefficient α, whose value is less than 1, make the algorithm converge faster [92]. A non-zero value of $\xi(n)$ is much helpful to force swarms into unidentified solution space in addition to the consequence carried out by cognitive and communal components of the PSO algorithm. Without extra stochastic behavior or $\xi(n)$ =0, the behavior of PSO-CREV and conventional PSO are same, but the convergence rate of PSO-CREV is fast. Hence in this chapter this additional stochastic behaviour is omitted to achieve faster convergence. The adverse effect on exploration capability can be reduced by selecting dynamic strategy for $\varepsilon(n)$ as $\varepsilon(n) = \frac{a}{(n+1)^b}$ where *a* and *b* are scalars. To balance both exploration capability and speed of convergence, the value of b must be properly selected. Choosing small value of b formulate the algorithm with powerful exploration capability [92]. Hence the weakened exploration capability of PSO-CREV

algorithm by the deletion of ξcan be compensated by suitably selecting a value for b. Other PSO-CREV parameters are $\varepsilon(n) = \frac{3.5}{(n+1)^{0.4}}$, $c_1 = c_2 = 2$ and $\alpha = 0.95$.

4.4 CATALYTIC CSTR PROCESS

The above discussed methods are applied for a SISO process. The SISO process considered for demonstration is CSTR. This section describes the schematic of a catalytic stirred tank reactor process which is simulated to show the performances of RVM based NMPC than LS-SVM based NMPC, Extreme ANFIS based NMPC and NN based NMPC. The continuous stirred-tank reactor, otherwise called as vat- or backmix reactor, is the reactor commonly used in chemical engineering. The Physical arrangement of CSTR plant is given in Fig. 2. In a catalytic reactor, the speed of catalytic reaction is relative to the amount of catalyst the reagents contact.

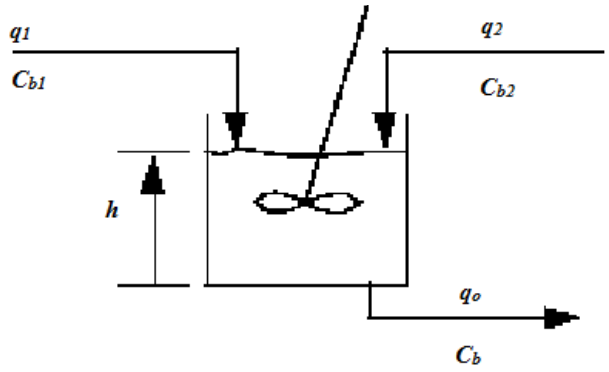

Fig. 4.2 Schematic of the CSTR process

The dynamic model of the Catalytic CSTR process is,

$$\frac{dh(t)}{dt} = q_1(t) + q_2(t) - 0.2\sqrt{h(t)} \tag{4.10}$$

$$\frac{dC_b(t)}{dt} = (C_{b1} - C_b(t))\frac{q_1(t)}{h(t)} + (C_{b2} - C_b(t))\frac{q_2(t)}{h(t)} - \frac{k_1 C_b(t)}{(1+k_2 C_b(t))^2} \tag{4.11}$$

where,

$h(t)$ - liquid level in the reactor,

$C_b(t)$ - concentration of product at the output of the process

$C_{b1}(t)$ - concentration of the concentrated feed of the process

$C_{b2}(t)$ - concentration of the diluted feed of the process

$q_1(t)$ - concentrated feed flow rate C_{b1},

$q_2(t)$ - diluted feed flow rate C_{b2},

q_0 - product flow rate at the output of the process.

The nominal conditions for the feed concentrations are set to $C_{b2} = 24.9\ mol\ L^{-1}$ and $C_{b1} = 0.1\ mol\ L^{-1}$. The rate of consumption of both the feeds are associated with the constants k_1=1 and k_2=1. The objective of this catalytic reactor is to obtain the desired concentration C_b at the product by adjusting the flow rates q_1 and q_2 .The illustration is made simple by setting $q_2(t) = 0.1\ L\ min^{-1}$. The Matlab Simulink model of the above illustrated CSTR process is shown in Fig.4.3.

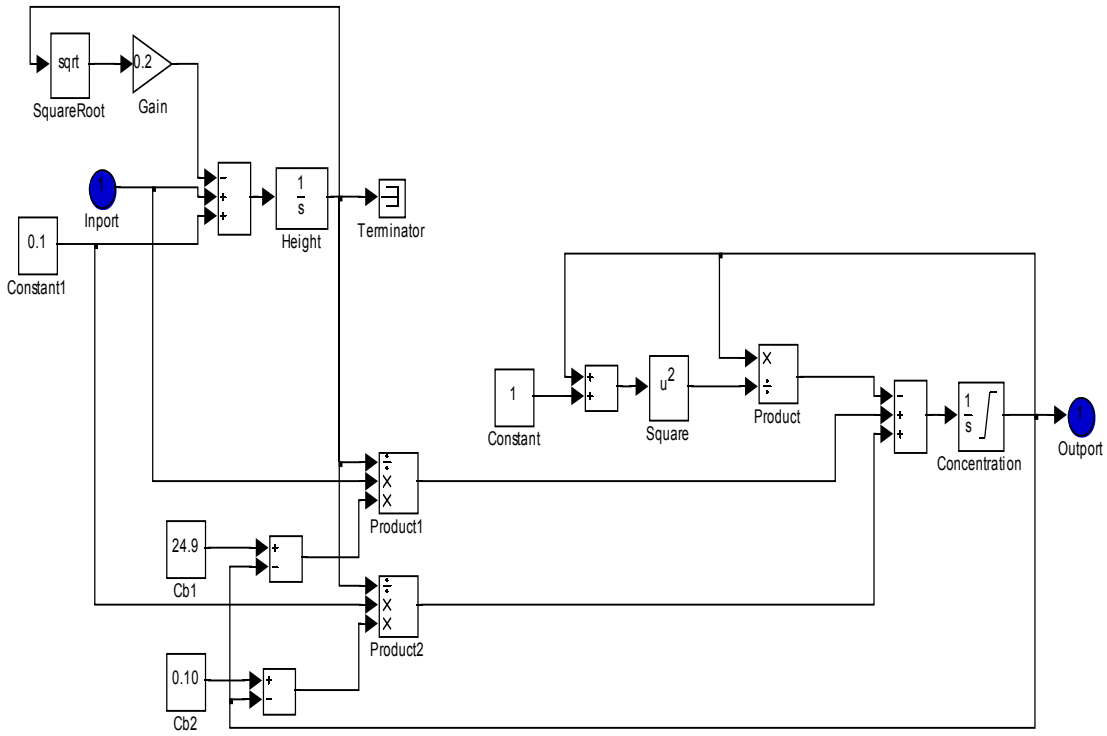

Fig. 4.3 MATLAB SIMULINK model of the CSTR process

4.5 NEURAL NETWORK BASED MPC OF CSTR PROCESS

This section describes the multilayer feed forward NN model based nonlinear MPC using PSO-CREV optimization algorithm. In order to develop an accurate model accurate data set is necessary and is generated by simulating the simulink model of CSTR process. This data set is meant for training, validating and testing the model. The trained, validated and tested model is then incorporated in MPC to provide accurate predictions of process dynamics.

4.5.1 Training and Testing the Model

A two layer feed forward neural network comprising of five hidden neurons with sigmoid activation function and an output neuron with linear activation function is used as the model in the NN model based MPC. NN requires large number of training data to

attain reasonable generalization and hence a sequence of 1000 samples is used for training the NN model and is done through Levenberg-Marquardt learning algorithm.

Fig. 4.4 and Fig. 4.5 correspond to the modeling results neural network method. The datasets are generated by providing random constrained signal as input to the plant. The constraint to the input signal is $0 \leq u(t) \leq 4$. The identification performance of NN model is assessed by the Root mean square error (RMSE) performance function.

$$RMSE = \left\{ \sum_{k=1}^{N} [\hat{y}(k) - y(k)]^2 \Big/ N \right\}^{1/2} \tag{4.12}$$

where, $\hat{y}(k)$ - predicted output for the sampling instant k,

$y(k)$ - output of the plant for the sampling instant k

N - total number of samples.

The RMSE value of the Neural Network models training error is 0.0312 and is tabulated in Table 4.1. The trained NN model is further tested with 100 samples of random inputs which are beyond the training data and there corresponding absolute prediction errors are calculated. The absolute prediction errors of NN model are shown in Fig. 4.5. The RMSE value of testing error is 0.0989 as tabulated in Table. 4.1. This proves the poor generalization capability of NN model.

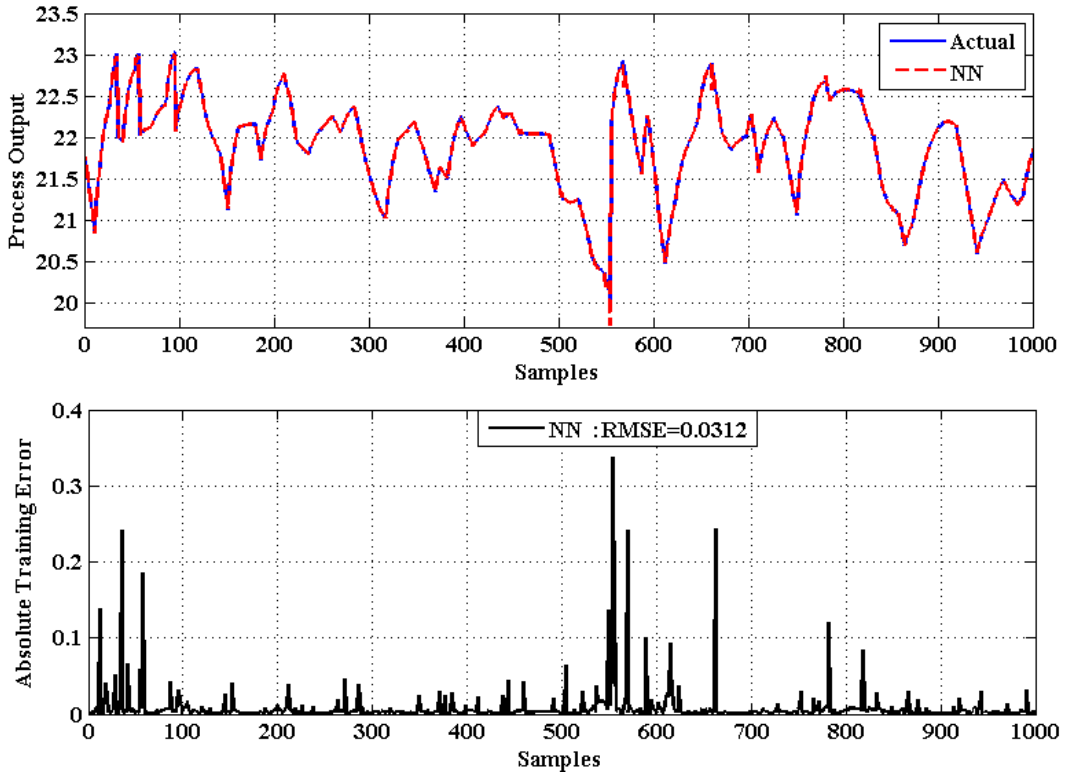

Fig. 4.4 Training performance of NN model

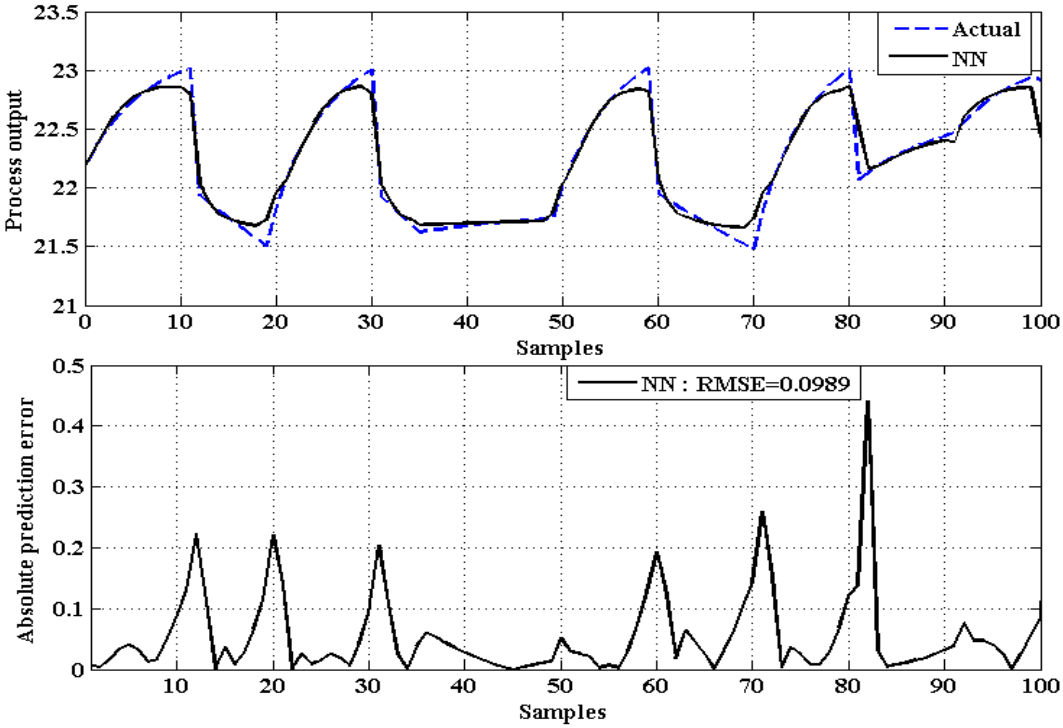

Fig. 4.5 Testing performance of NN model

4.5.2 Performances of NN –PSO-CREV-MPC

This section describes the set point tracking performance and unmeasured disturbance rejection capability of NN-PSO-CREV based MPC. The offline trained and validated NN model is then used as the nonlinear model for nonlinear MPC. The value of N_1, N_2 and c N_u of NN based MPC are set to 1, 1 and 1 correspondingly. The control input weighting factor λ is set to 0.5.

Fig. 4.6 illustrates the random set point tracking performances of NN model based MPC with PSO-CREV optimization algorithm. Fig. 4.7 shows the corresponding changes in the process variable, flow rate. The tracking performance of NN-PSO-CREV based MPC suffers from overshoots and undershoots which is due to their poor generalization capability.

As the PSO-CREV algorithm converges to the finest solution at each sampling instant, the control variable flow rate corresponding to the controller is with very less fluctuations. The control variable u(k) i.e., flow rate of the concentrated feed C_{b1} **,** is shown in Fig. 4.7 whose smoothness shows the index of control performance.

The unmeasured disturbance rejection capability of NN-PSO-CREV based MPC is examined by subjecting the CSTR process with dissimilar magnitudes of disturbances at

different sampling instants. The control variable, flow rate of the concentrated feed C_{b1} with disturbances at different sampling instants are shown in Fig. 4.8. The unmeasured disturbance rejection performance of NN-PSO-CREV based MPC is shown in Fig 4.9. The controlled variable, product concentration settles down slowly with oscillations after getting affected by unmeasured disturbances.

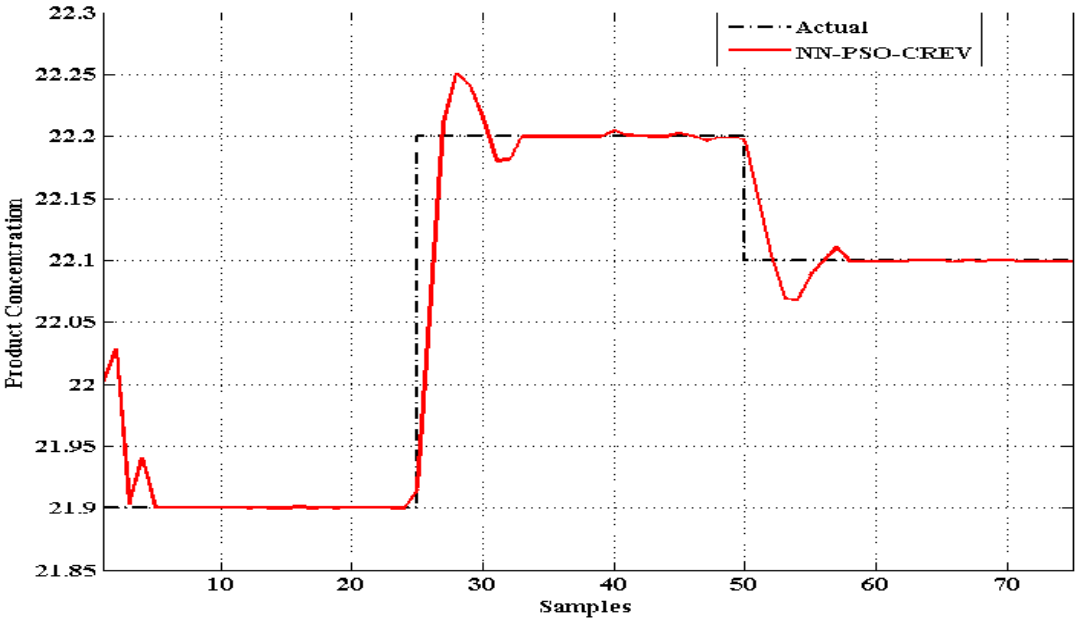

Fig. 4.6 Tracking performance of product concentration for CSTR process

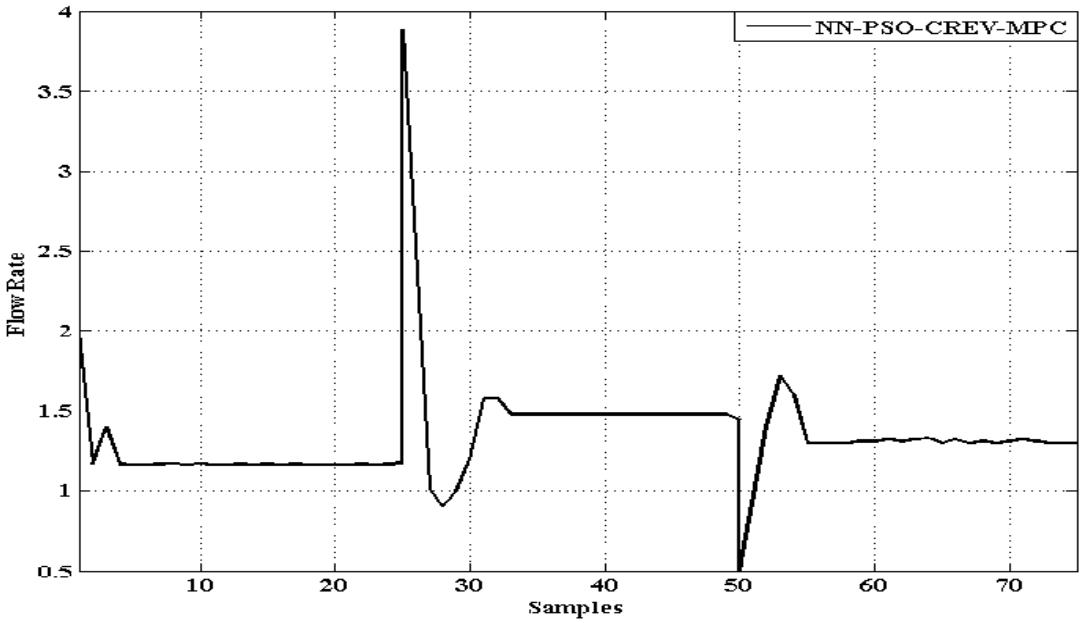

Fig. 4.7 Changes in the process variable for tracking the product concentration of CSTR process.

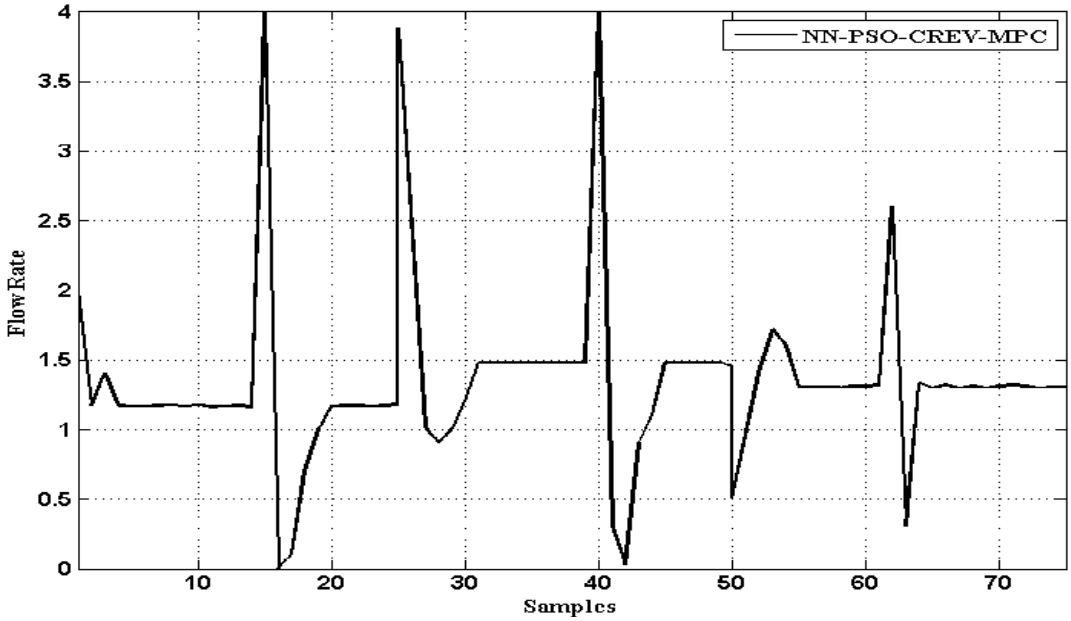

Fig. 4.8 Changes in the process variable to show unmeasured disturbances

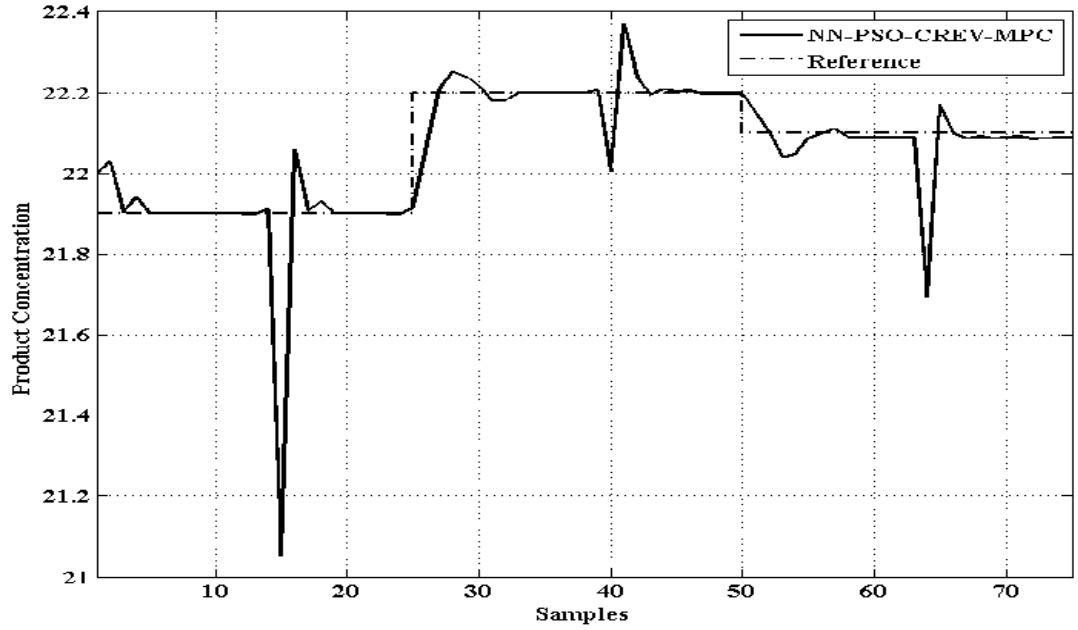

Fig. 4.9 Performance of unmeasured disturbance rejection

4.6 SVM BASED MPC OF CSTR PROCESS

This section describes the behaviour of SVM based MPC using PSO-CREV optimization algorithm. In order to develop an accurate model accurate data set is recorded by simulating the Matlab simulink model and is meant for training, validating and testing the model. The trained, validated and tested model is then incorporated in MPC to provide accurate predictions of process dynamics.

4.6.1 Training and Testing the Model

A sequence of 300 samples is used to train the SVR model offline using the leave one out method discussed in chapter 3. Figure 4.10 and Figure 4.11 corresponds to the modeling results of Support vector regression model. The identification routine of SVR model is assessed by the RMSE performance function as in equation (4.13).

$$RMSE = \left\{ \sum_{k=1}^{N} [\hat{y}(k) - y(k)]^2 \Big/ N \right\}^{1/2} \tag{4.13}$$

where $\hat{y}(k)$ represents the support vector regression model's output for the sampling instant k, $y(k)$ represents the actual plant's output for the sampling instant k and N represents total samples.

The RMSE value of the training error corresponding to support vector regression model is given in Table 4.1, where the training accuracy of SVR model and Extreme ANFIS model are almost same. The trained SVR model is further tested with 100 samples of random inputs which are beyond the training data and there corresponding absolute prediction errors are calculated.

The prediction errors of SVR model for 100 different samples are shown in Figure 4.11, which explores the good generalization capability of SVR model. Thus one can conclude that SVM based empirical modeling behaves suitably for industrial process control with good generalization.

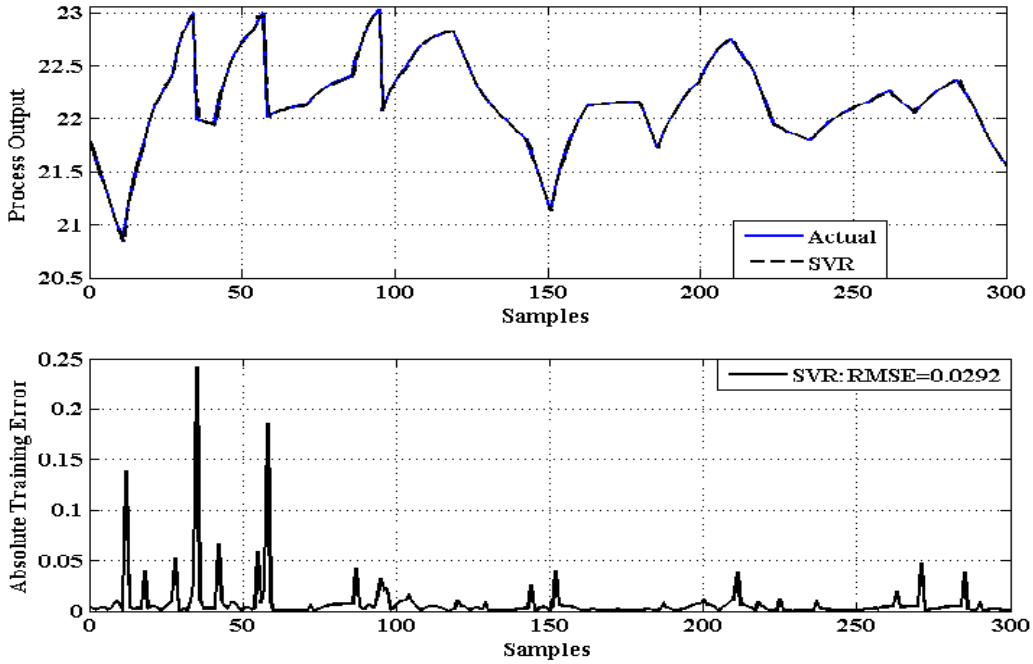

Fig. 4.10 Training performance of SVR model

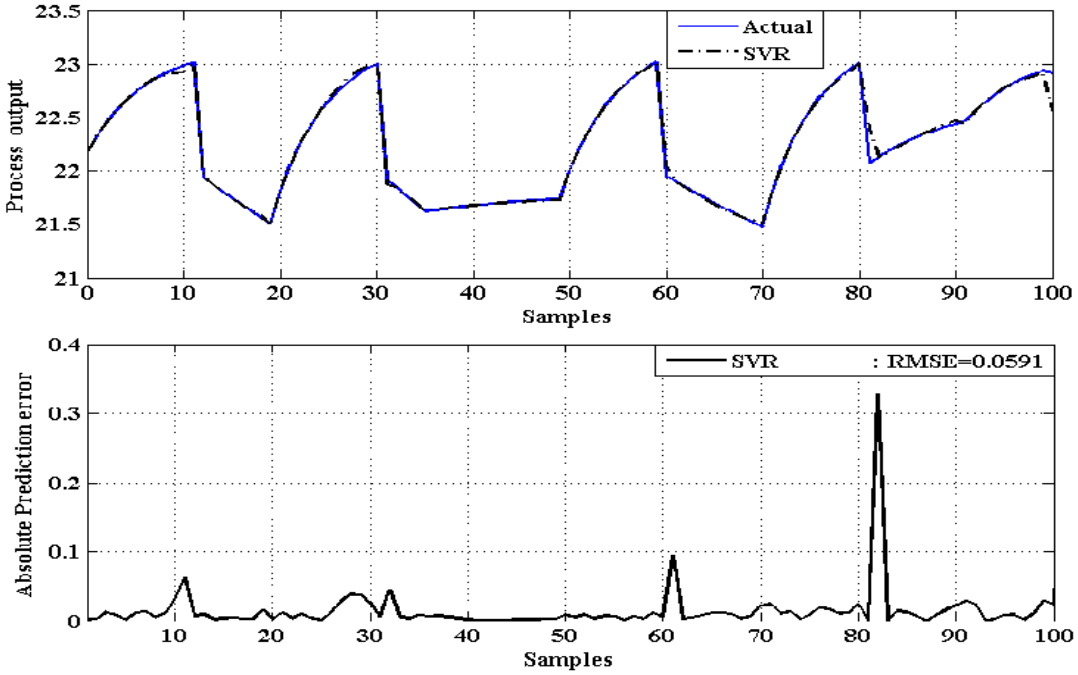

Fig. 4.11 Testing performance of SVR model

4.6.2 Performances of SVM-PSO-CREV-MPC

The offline trained and tested SVR model is then used as the nonlinear model for nonlinear MPC. Fig. 4.12 illustrates the random set point tracking performances of support vector regression model based MPC with PSO-CREV optimization algorithm. Fig. 4.13 shows the corresponding changes in the process variable, flow rate.

The SVR based MPC tracks the set point very well without any overshoot or undershoot as a result of its accurate model with good generalization capability.

Next, the unmeasured disturbance rejection capability of SVM-PSO-CREV based MPC is examined by subjecting the CSTR process with dissimilar magnitudes of disturbances at different sampling instants. The control variable, flow rate of the concentrated feed C_{b1} with disturbances at different sampling instants are revealed in Fig. 4.14.

The unmeasured disturbance rejection capability of SVM based MPC is shown in Fig. 4.15. The controlled variable, product concentration settles down smoothly with very less oscillations after getting affected by unmeasured disturbances.

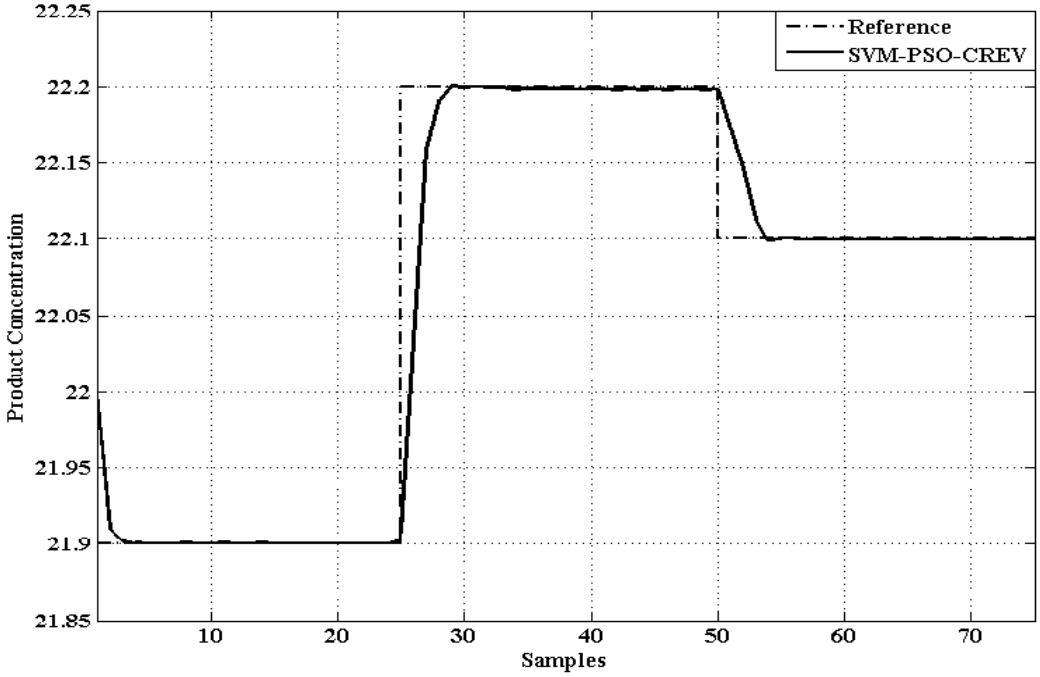

Fig. 4.12 Tracking performance of product concentration for CSTR process

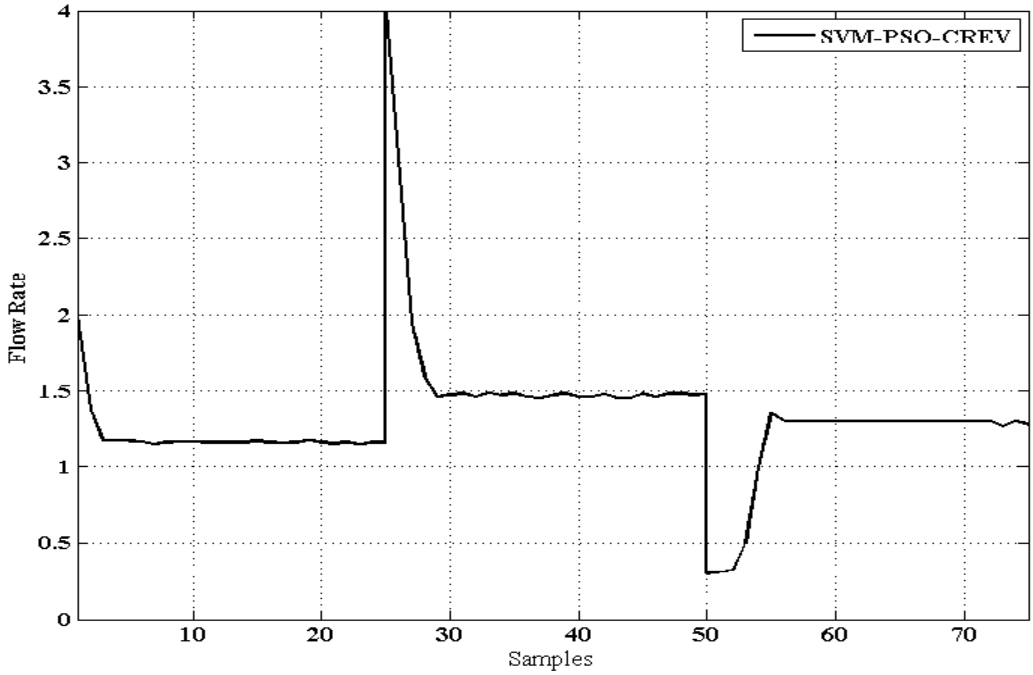

Fig. 4.13 Changes in the process variable for tracking the product concentration of CSTR Process

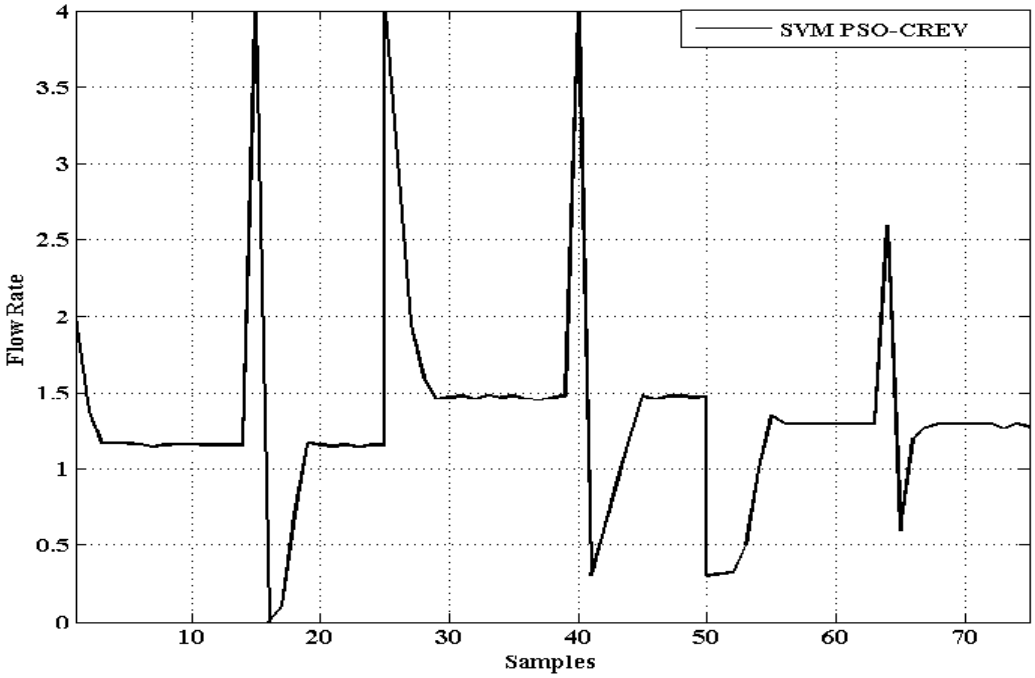

Fig. 4.14 Changes in the process variable to show unmeasured disturbances

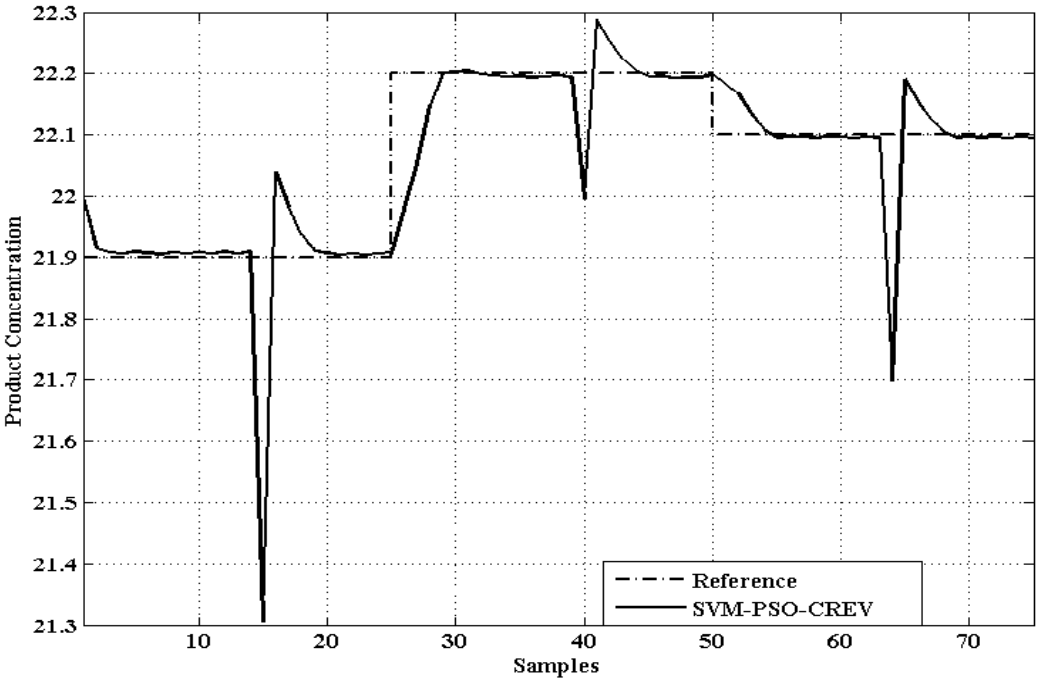

Fig. 4.15 Performance of unmeasured disturbance rejection

4.7 RVM BASED MPC OF CSTR PROCESS

This section describes the suitability of RVM based MPC using PSO-CREV optimization algorithm in tracking random set points and overcoming unmeasured disturbances. In order to develop an accurate model accurate data set is necessary. This data set is meant for training and testing the model. The trained and tested model is then incorporated in MPC to provide accurate predictions of process dynamics.

4.7.1 Training and Testing the Model

A sequence of 300 samples is used to train the sparse Bayesian RVR model offline.

RBF kernel function is used by the model and kernel width parameter is estimated accurately, which get better generalization ability and less computational complexity of the training process. The value of N_1, N_2 and N_u of RVM based MPC are set to 1, 1 and 1 correspondingly to reduce the computational burden. The control input weighting factor λ is set to 0.5.

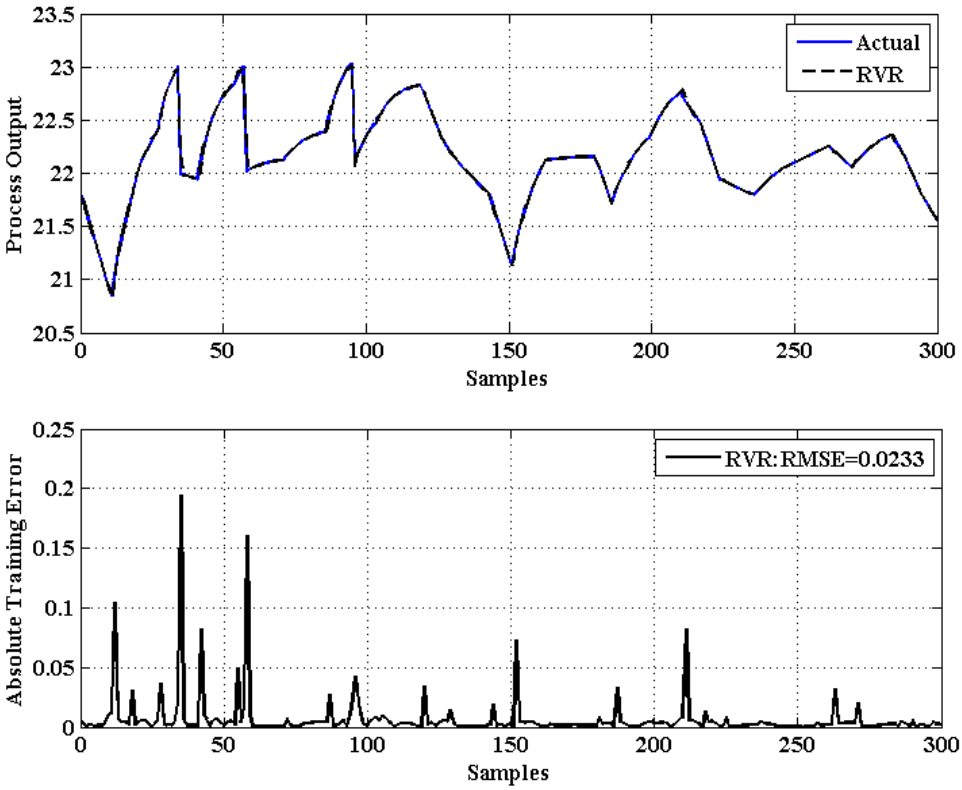

Fig. 4.16 Training performance of RVR model

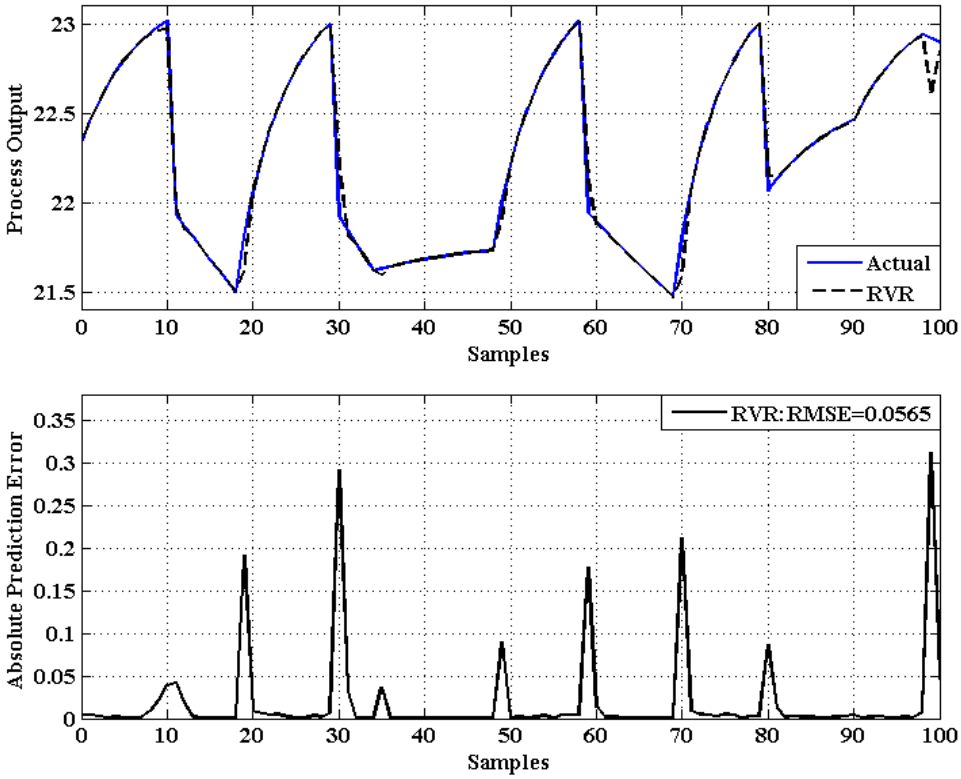

Fig. 4.17 Testing performance of RVR model

Fig. 4.16 and Fig. 4.17 correspond to the modeling results of RVR model. The datasets are generated by providing random constrained signal as input to the plant. The constraint to the input signal is $0 \leq u(t) \leq 4$. The training error of RVR model shows the training accuracy of RVR model.

The trained RVR model is further tested with 100 samples of random inputs which are beyond the training data and there corresponding absolute prediction errors are calculated. The graph of prediction errors of RVR model is shown in Fig. 4.17, which explores the better generalization capability of RVR model. Thus one can conclude that the RVR based empirical modeling can prevail better with accurate model having good generalization capability. The model accuracy is accessed by RMSE value which is the square root of the average of the squared error values as in equation (4.11).

4.7.2 Performances of RVM-PSO-CREV-MPC

The offline trained and tested RVR model is then used as the nonlinear model for nonlinear MPC. Fig.4.18 illustrates the random set point tracking performances of RVR model based MPC with PSO-CREV.

The tracking performance of RVR based MPC is free from overshoots and undershoots due its accurate prediction and precise optimization using PSO-CREV.

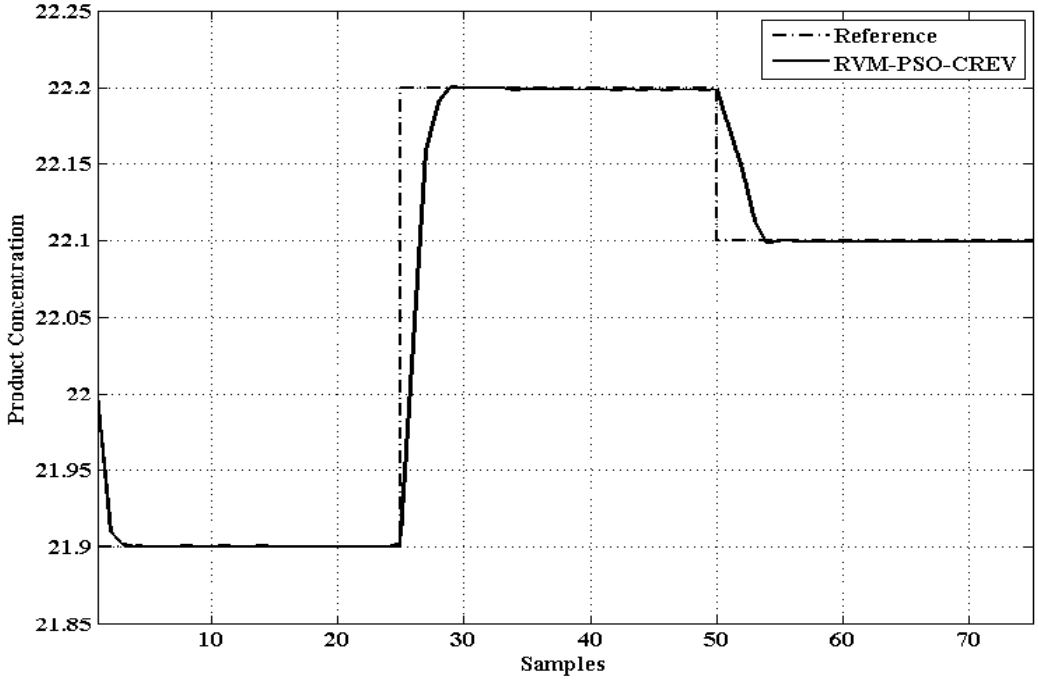

Fig. 4.18 Tracking performance of product concentration for CSTR process

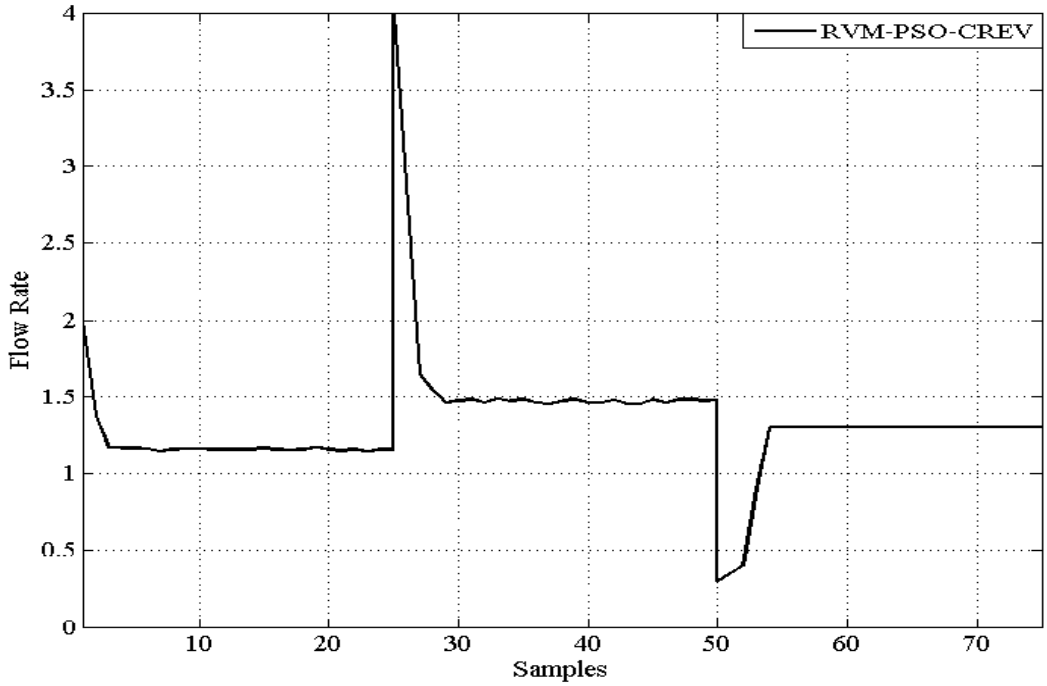

Fig. 4.19 Changes in the process variable for tracking the product concentration of CSTR process

As the PSO-CREV algorithm converges to the finest solution at each sampling instant, the control variable flow rate corresponding to RVM-PSO-CREV is with very less fluctuations. The control variable u(k) i.e., flow rate of the concentrated feed C_{b1} , is shown in Fig. 4.19 whose smoothness shows the index of control performance.

The unmeasured disturbance rejection capability of RVM-PSO-CREV based MPC is analyzed by subjecting the CSTR process with dissimilar magnitudes of disturbances at different sampling instants. The control variable, flow rate of the concentrated feed C_{b1} with disturbances at different sampling instants is shown in Fig. 4.20.

Certainly the unmeasured disturbance rejection performance of RVM-PSO-CREV based MPC is very good with very less oscillations. The controlled variable, product concentration settles down faster after getting affected by unmeasured disturbances as shown in Fig. 4.21. Thus RVM-PSO-CREV based MPC behaves suitably for process control industrial applications.

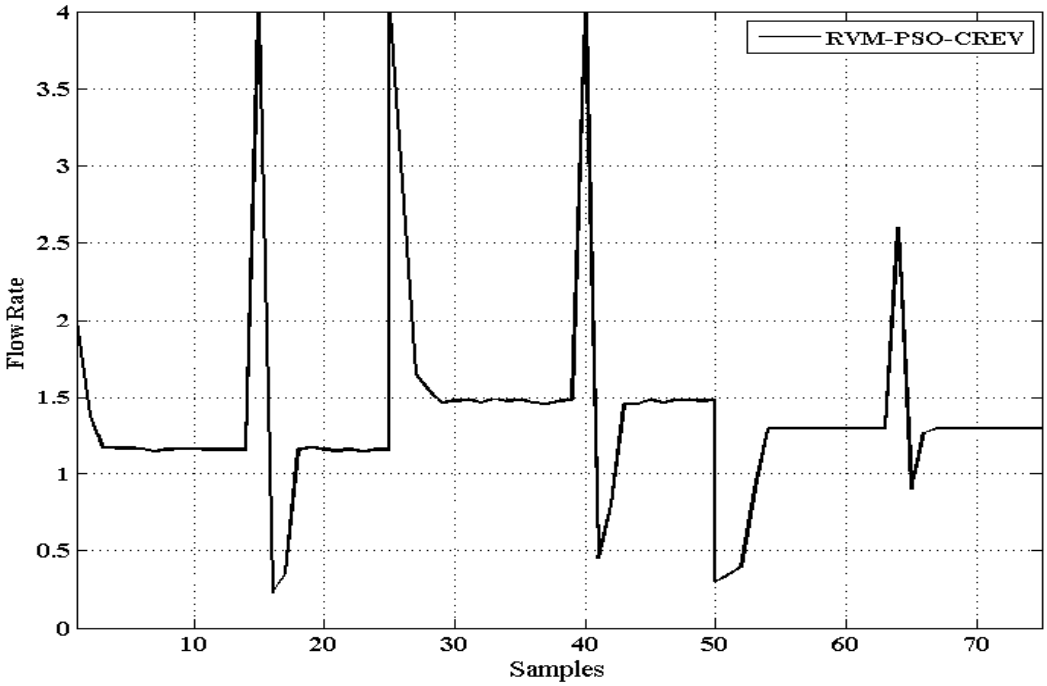

Fig. 4.20 Changes in the process variable to show unmeasured disturbance

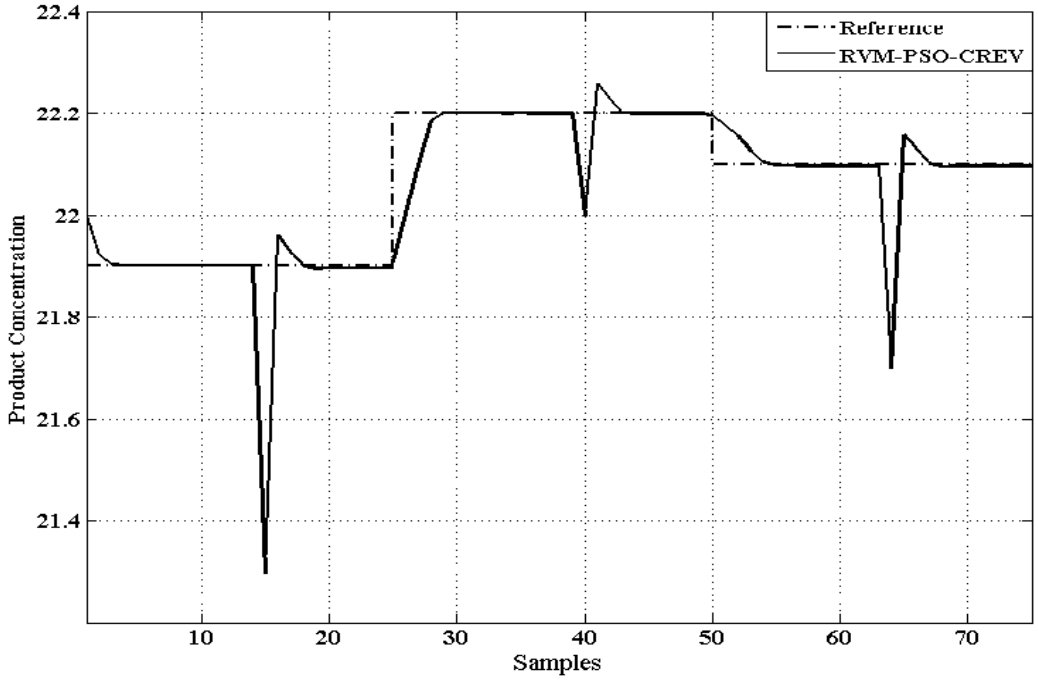

Fig. 4.21 Performance of unmeasured disturbance rejection

4.8 EXTREME ANFIS BASED MPC OF CSTR PROCESS

This section describes the proposed novel neuro-fuzzy model based MPC called Extreme ANFIS based MPC using PSO-CREV optimization algorithm. The accurate data set spreading randomly over the entire operating region of the CSTR process is generated by simulating the simulink model of CSTR process. This data set is meant for training, validating and testing the model. The trained and tested model is then incorporated in MPC to provide accurate predictions of process dynamics.

4.8.1 Training and Testing the Model

Randomly generated 300 training data pairs are used to train the Extreme ANFIS network offline. The four inputs to the network are *y(k), y(k-1), u(k) and u(k-1)*. Here, y*(k)* and *y(k-1)* are the current process output and delayed process output respectively. Also *u(k)* and *u(k-1)*are current process input and delayed process input respectively. Three bell shaped membership functions with 81 rules are used for adapting parameters of ANFIS. This novel Extreme ANFIS model has good generalization ability with less training time. The final membership function after learning the nonlinear CSTR process with Extreme ANFIS algorithm is shown in Fig.4.22.

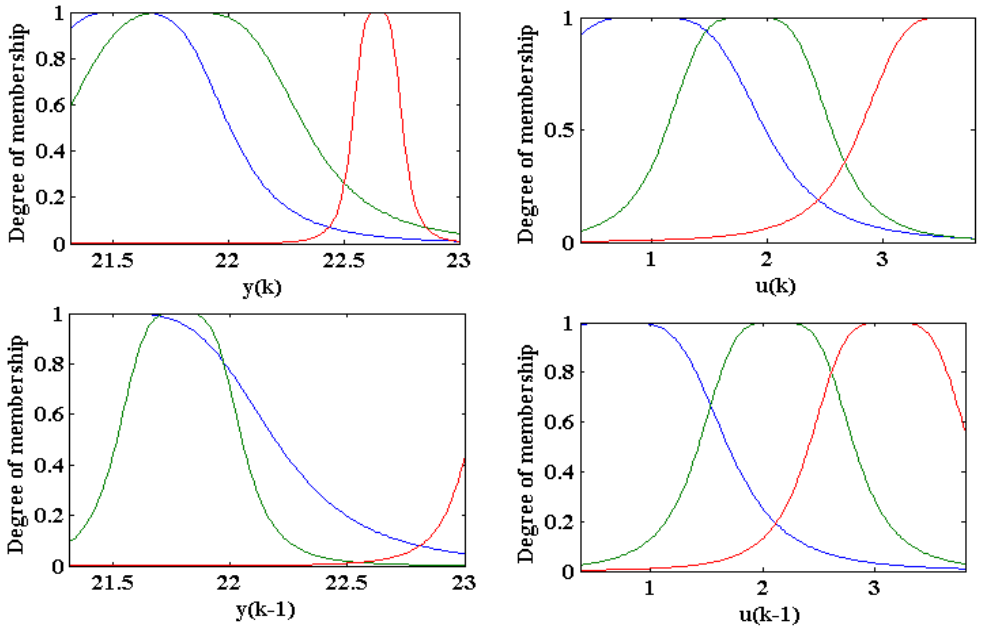

Fig. 4.22 Final membership functions after learning with Extreme ANFIS algorithm

The datasets are generated by providing random constrained signal as input to the plant. The constraint to the input signal is $0 \leq u(t) \leq 4$. The identification performance of Extreme ANFIS model is assessed by the RMSE performance function.

$$RMSE = \left\{ \sum_{k=1}^{N} [\hat{y}(k) - y(k)]^2 \Big/ N \right\}^{1/2} \tag{4.14}$$

where $\hat{y}(k)$ represents the Extreme ANFIS model's output for the sampling instant k, where $y(k)$ represents the actual plant's output for the sampling instant k and N represents total number of samples.

Fig. 4.23 and Fig. 4.24 correspond to the modeling results of Extreme ANFIS method.

The RMSE value of training error for Extreme ANFIS model is 0.0303 and is tabulated in Table 4.1, the training accuracy of Extreme ANFIS model and NN model used in previous section are almost same. The trained Extreme ANFIS model is further tested with 100 samples of random inputs which are beyond the training data and there corresponding absolute prediction errors are calculated. The prediction errors of Extreme ANFIS model is given in Fig. 4.24. The RMSE of testing error for Extreme ANFIS model is 0.0605 and is tabulated in Table. 4.1. The generalization capability of Extreme ANFIS model is good. This is because of the qualitative adaptive capability of Extreme ANFIS model.

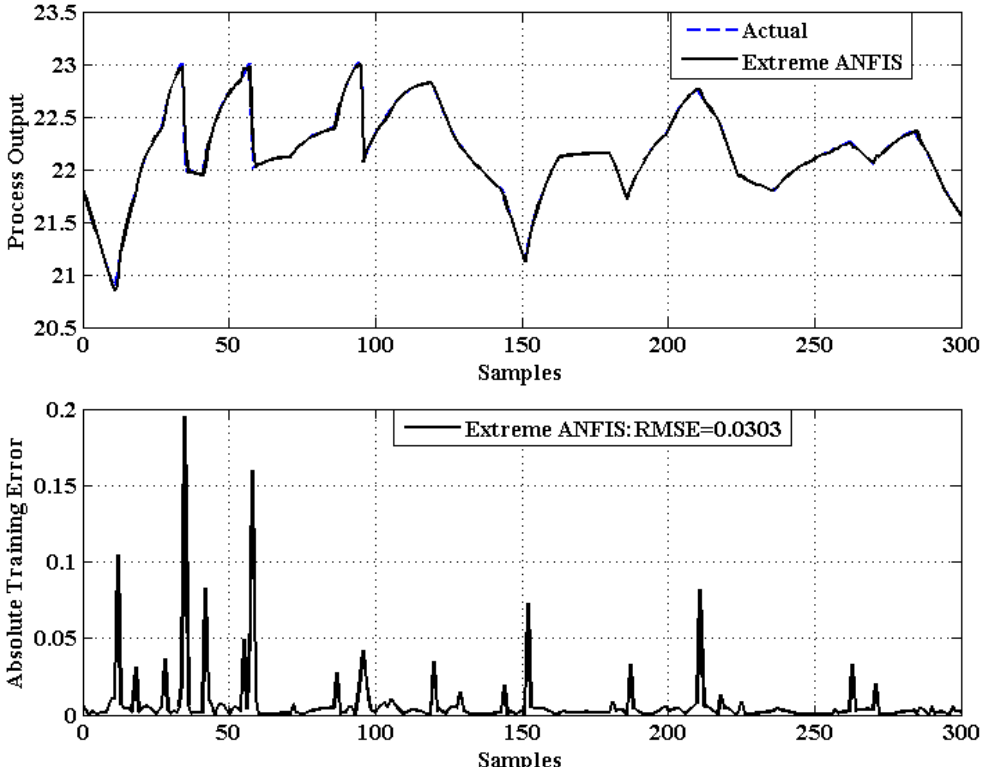

Fig. 4.23 Training performance of Extreme ANFIS model

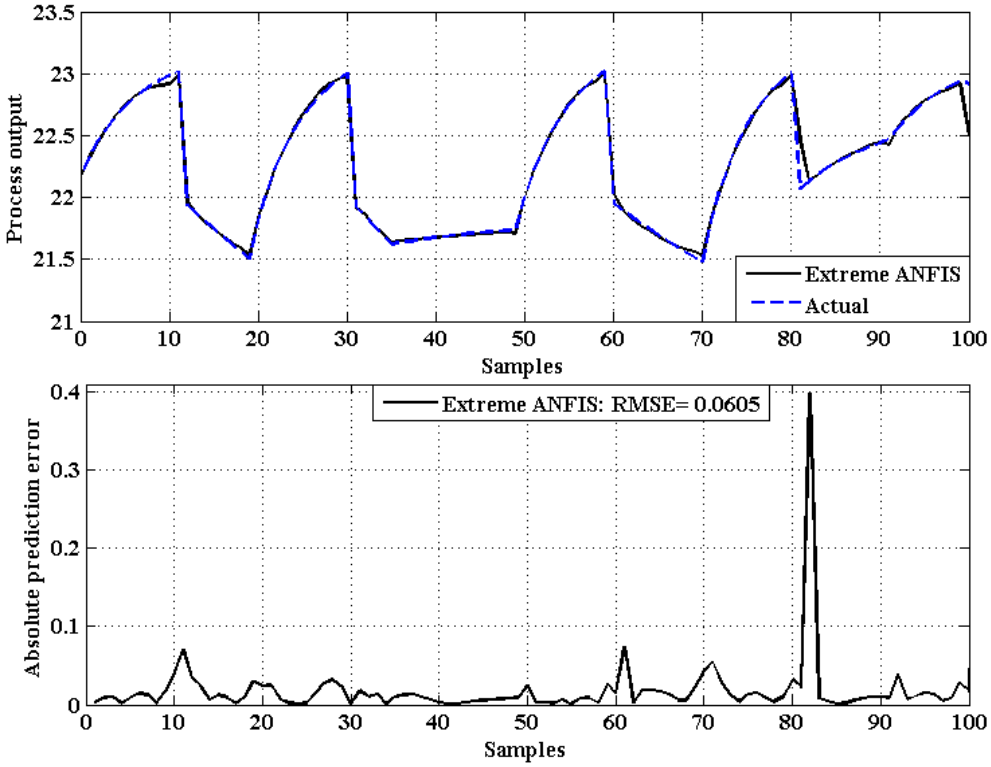

Fig. 4.24 Testing performance of Extreme ANFIS model

4.8.2 Performance of Extreme ANFIS-PSO-CREV-MPC

The offline trained and tested Extreme ANFIS model is then used as the nonlinear model for nonlinear MPC. The value of N_1, N_2 and N_u of Extreme ANFIS based MPC and NN based MPC are set to 1, 1 and 1 correspondingly. The control input weighting factor λ is set to 0.5. Fig. 4.25 illustrates the random set point tracking performances of Extreme ANFIS model based MPC with PSO-CREV. Fig. 4.26 shows the corresponding changes in the process variable, flow rate.

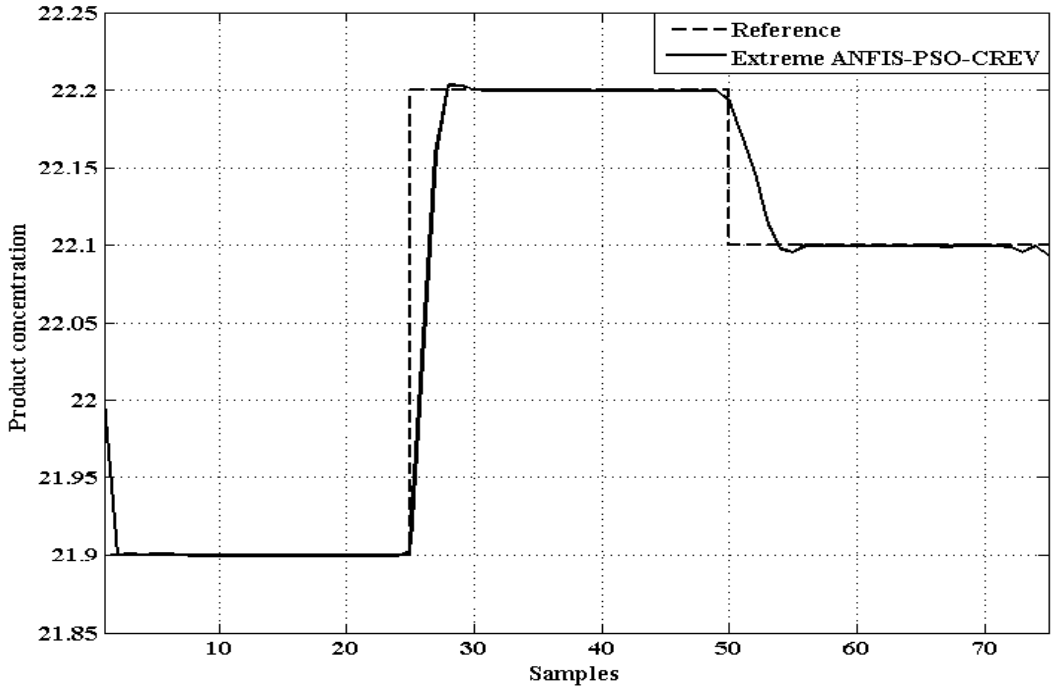

Fig. 4.25 Tracking performance of product concentration for CSTR process

The tracking performance of Extreme ANFIS based MPC is free from large overshoots or undershoots due to their accurate predictions because of its good generalization capability and precise optimization using PSO-CREV.

As the PSO-CREV algorithm converges to the finest solution at each sampling instant, the control variable flow rate corresponding to Extreme ANFIS-PSO-CREV is with very less fluctuations. The control variable u(k) i.e., flow rate of the concentrated feed C_{b1}, is shown in Fig. 4.26 whose smoothness shows the index of control performance.

The unmeasured disturbance rejection capability of Extreme ANFIS-PSO-CREV based MPC is analyzed by subjecting the CSTR process with dissimilar magnitudes of disturbances at different sampling instants. The control variable, flow rate of the concentrated feed C_{b1} with disturbances at different sampling instants are shown in Fig. 4.27. The unmeasured disturbance rejection performance of Extreme ANFIS-PSO-CREV

based MPC is shown in Fig. 4.28. The controlled variable, product concentration settles down smoothly with very less oscillations after getting affected by unmeasured disturbances.

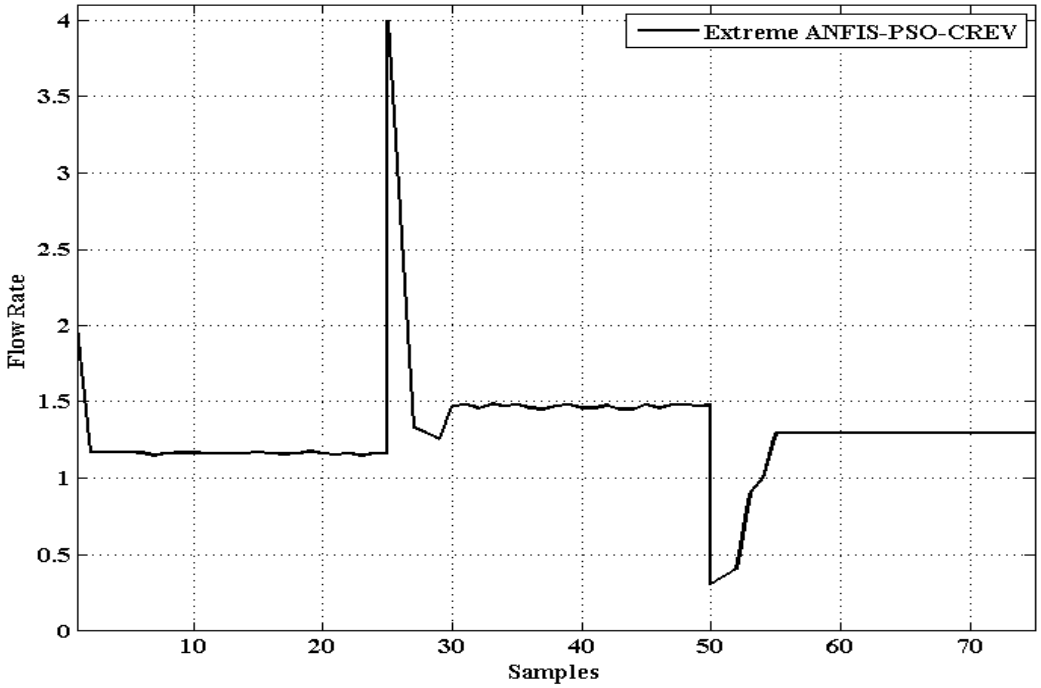

Fig. 4.26 Changes in the process variable for tracking the product concentration of CSTR process

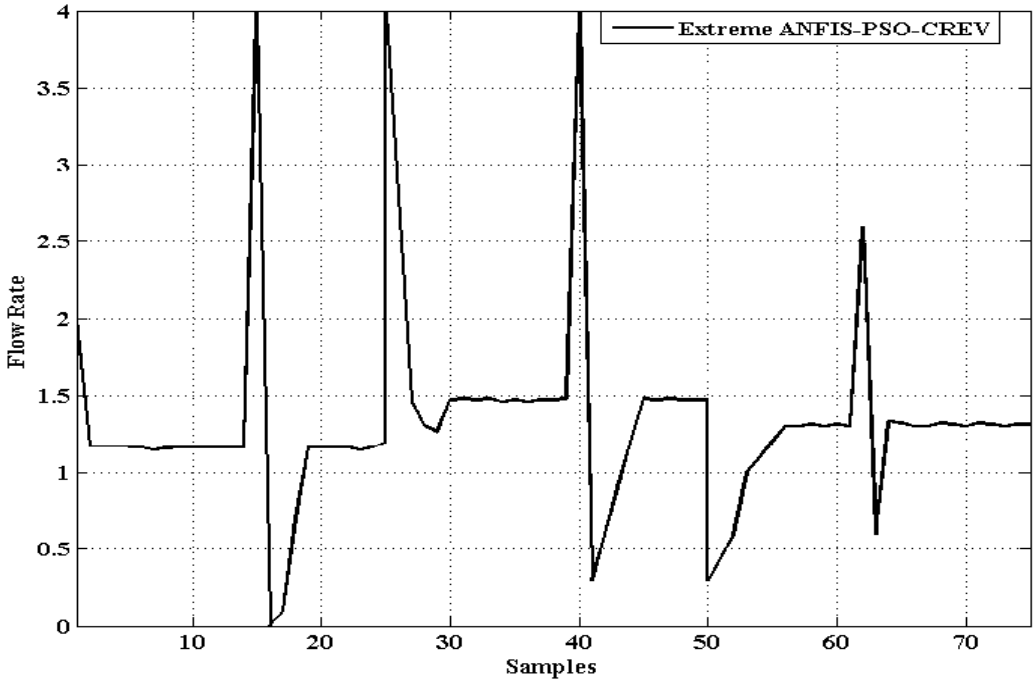

Fig. 4.27 Changes in the process variable to show unmeasured disturbances

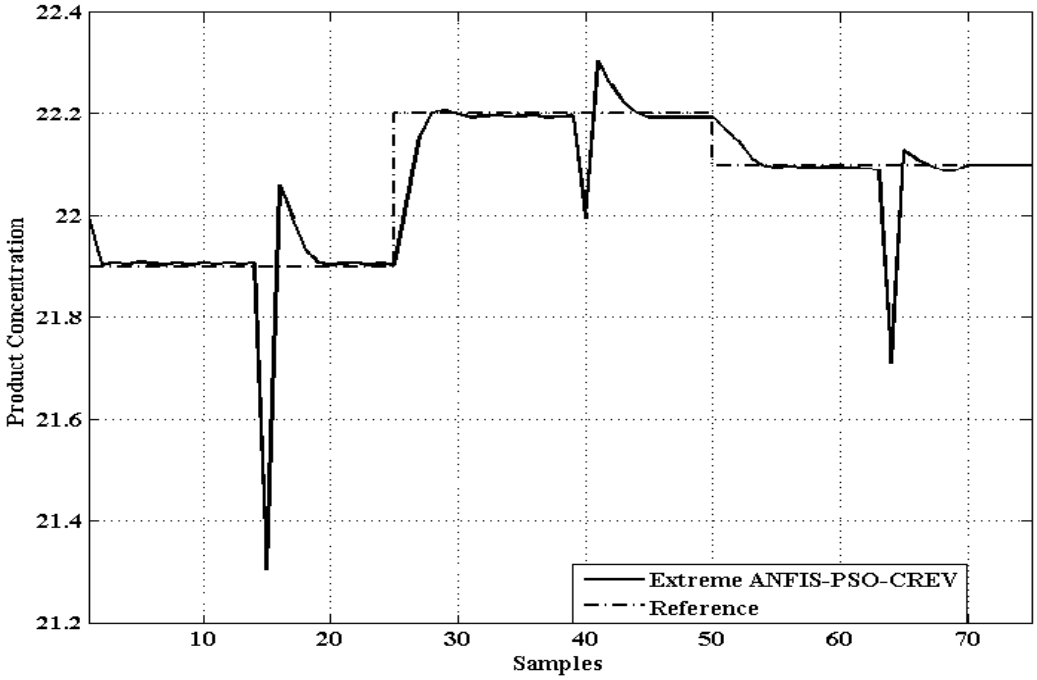

Fig. 4.28 Performance of unmeasured disturbance rejection

4.9 TABULATION OF PERFORMANCE INDICES OF DIFFERENT MPC's

This section enunciates the model accuracy of different empirical models developed and the performance indices and computational cost of all the controllers designed in the previous sections.

Table 4.1 shows the training accuracy, testing accuracy of various models with their corresponding number of training data. Even though the NN model is trained using 1000 training data, its training error and testing error are more. But the Extreme ANFIS model, SVR model and RVR model uses less number of training data and behaves with good generalization capability. This is clear from their corresponding prediction errors for unseen test data.

Table 4.1 Accuracy of different empirical models

Model	Number of Training data	$RMSE_{training}$	$RMSE_{testing}$
NN	1000	0.0312	0.0989
Extreme ANFIS	300	0.0303	0.0605
SVR	300	0.0292	0.0591
RVR	300	0.0233	0.0565

Table.4.2 shows the Integral absolute error (IAE) value, number of support vectors/ relevance vectors and computational time related to each controller for the simulation results carried out for 75 samples. IAE is the performance criteria which quantifies the accuracy of all controllers. Fig 4.29 and Fig. 4.30 illustrates the set point tracking performance of all the controllers discussed in previous sections.

The sparseness property of SVR model reduces the computational time of SVR-MPC to 10.06 seconds for 75 samples (i.e. nearly 0.1341 Seconds for sample). But the sparse nature of RVR model sharply reduces the computational time of RVR-MPC to 4.71 seconds for 75 samples (ie. nearly 0.06 Seconds for sample), which is much shorter than the sampling time of the CSTR process. The number of support vectors and relevance vectors used while modeling SVR and RVR model respectively tabulated in Table 4.2, signifies the sparseness of the models.

Table 4.2 Performance Indices of various control strategies

Conditions	Control tactics	Number of Training Samples	IAE	Number of Support Vectors/ Relevance Vectors	computational time (Seconds)
No Disturbance	NN-PSO-CREV	1000	0.982	-	21.23
	Extreme ANFIS-PSO-CREV	300	0.7701	-	5.21
	LS-SVM-PSO-CREV	300	0.7619	162	10.06
	RVM-PSO - CREV	300	0.6640	27	4.71
Disturbance	NN-SVM-PSO-CREV	1000	3.4100	-	22.43
	Extreme ANFIS-PSOCREV	300	2.9008	-	6.04
	LS-SVM-PSO-CREV	300	2.8838	162	11.40
	RVM-PSO - CREV	300	2.2933	27	4.78

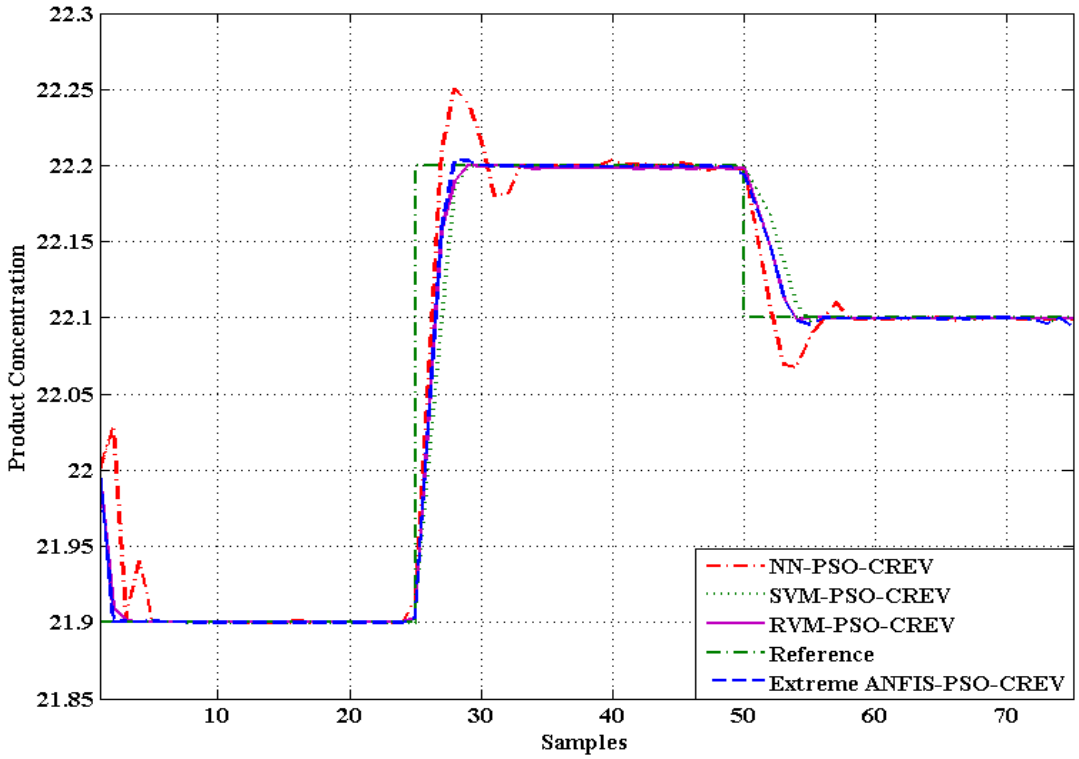

Fig. 4.29 Tracking performance comparison of product concentration for CSTR process

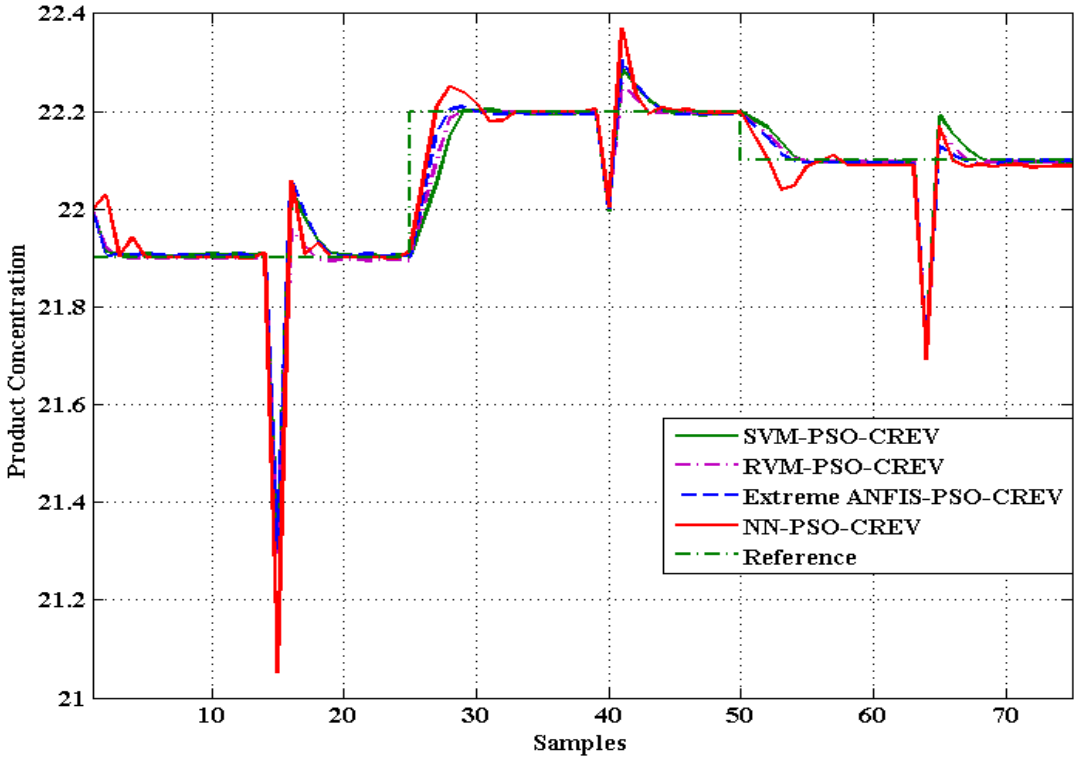

Fig. 4.30 Performance comparison of unmeasured disturbance rejection

The Extreme ANFIS model based MPC consumes only 5.21 seconds (ie. nearly 0.06 Seconds for sample) due to its simple structure and simple algorithm. Thus the time consumption in Extreme ANFIS based MPC is same as RVM based MPC. But, the dynamic NN model based MPC consumes 21.23 Seconds which is much larger than the above mentioned controllers.

Hence, it is clear that NN based MPC is the one which consumes more time with more integral average error. As a result, RVR-PSO-CREV, SVR-PSO-CREV and Extreme ANFIS-PSO-CREV model predictive controller are better than NN based controller with less computational load and less integral average error.

Also Extreme ANFIS-PSO-CREV and RVR-PSO-CREV based MPC's consumes very less time with good tracking performances, which is much essential for real time applications. Therefore we conclude RVR-PSO-CREV and Extreme ANFIS-PSO-CREV based MPC's to be the best controller based on various attributes like usage of less number of training data, better prediction accuracy, better generalization capability, excellent set point tracking performance, better unmeasured disturbance rejection capability. Hence it is well suitable for industrial process control applications.

4.10 CONCLUSION

A viable solution to the problem of fast implementation of NMPC is proposed in this chapter. Different machine learning techniques are used to create an accurate for prediction model and a derivative free optimization method, PSO-CREV is incorporated to achieve faster convergence. Based on the simulation results of CSTR process the tracking performance of RVM- PSO-CREV, Extreme ANFIS-PSO-CREV, LS-SVM-PSO-CREV based MPC's are much better than NN based MPC with very less computational cost and better unmeasured disturbance rejection capability which confirms its feasibility. Simulation results convey that such better performance is due to their better prediction accuracy and better generalization capability. Also the computation time of Extreme ANFIS based MPC and RVM based MPC are very less and equal which makes them well suitable for real time control applications.

Chapter 5

NONLINEAR MODEL PREDICTIVE CONTROL FOR MAXIMUM POWER POINT TRACKING OF PHOTO VOLTAIC ARRAY

This chapter describes the model predictive control of a system with faster dynamics. A photovoltaic array with power converter is considered as the fast dynamic system. Thus the control of photovoltaic array Maximum Power Point Tracker system through Nonlinear Model Predictive Control strategy using novel Neuro-Fuzzy technique, Relevance Vector Machines regression model and Support Vector Machines regression model is described. The optimization problem in the above control algorithm is neglected by Finite Control Set Model Predictive Control technique. The simulation results comparing model predictive control of PV array using deterministic sparse kernel learning technique, neuro-fuzzy technique and probabilistic sparse kernel technique are shown.

5.1 INTRODUCTION

Control of systems with faster dynamics is beyond the scope of MPC, as it does prediction and optimization at each sampling instant. But it is made feasible in this chapter by incorporating FCS-MPC technique. As a result of this, the necessity for optimization of control sequence at each sampling instant is avoided. A photo voltaic array maximum power point tracking system is the fast dynamic process considered for analysis.

In recent years, fossil fuels are in its deteriorating path which increases the demand for alternative sources of energy. In spite of its installation cost, photo voltaic system persist to draw worldwide wellbeing as it is a renewable source with zero noise or pollution with very less maintenance cost. Maximum energy could be extracted via the load from the photovoltaic systems, by the operation of a direct current (DC) converter which boosts the solar panel output voltage. The power efficiency of PV system could be maintained maximum by a controller which forces the system to operate at a sole point called Maximum Power Point (MPP), found at the knee of the I-V characteristics, at which the array delivers maximum power. A maximum power point tracker which identifies and supplies reference maximum power point input to the controller is also essential.

One of the commonly used methods to identify the MPP is Perturb and Observe method [148-150] which has the drawbacks of tracking the wrong direction under rapidly fluctuating irradiance levels and oscillation around the MPP under steady state operation. The incremental conductance is the other method which overrules the above mentioned drawbacks but it has the disadvantage of increased complexity and cost [151, 152]. Photovoltaic Power Output Estimation is done by Genetically Evolved Fuzzy Predictor in

[153]. Neural networks are extensively used for Maximum Power Point Tracking of photo voltaic array [154, 155]. However, the generalization capabilities of the neural networks are very poor. Another concern regarding neural network solutions is the presence of multiple local minima which may not result in a distinctive solution. Hence advanced machine learning techniques are to be used for Maximum Power Point Tracking of photo voltaic arrays.

The sparse kernel learning is a new nonlinear system identification method initially proposed in the machine learning area [12, 13]. A sparse kernel technique, Support Vector Regression introduced in [15], overwhelms the over fitting and poor generalization capability of neural network with less number of training data and less training time. But, practical applications of Least Squares Support Vector Machines are limited because of its requirement of larger number of support vectors to approximate the optimal solutions. In LS-SVM the regularization parameter γ and the kernel width parameter σ are the two free parameters to be tuned to improve the generalization ability of predicted model.

The promising results for the application of SVM model in PV power forecasting is discussed in [157]. But as the LS-SVM model is burdened with additional externally determined parameters, time consumption is more. Subsequently Tipping [21] introduced Relevance Vector Machine which attracted much interest in the research community owing to its advantages over support vector machine. They are established on a Bayesian formulation which results in usage of less number of relevance vectors leading to much more sparse representation than Support Vector Machine. RVM has no restriction on the basis functions [15, 22]. The sparse RVR model could generalize better with very less computation time with only one tuning parameter kernel width σ. The generalization performance of RVM and SVM are compared in the results given in [21].

The guaranteed model accuracy and generalization capability of Relevance Vector Regression model are explicitly acknowledged by many researchers [64-69]. Thus by RVR, the problem of over fitting can be avoided; generalization ability can be improved with better extrapolation capability with less number of training data and less training time.

Extreme ANFIS model is the proposed novel model which provides accurate model with good generalization capability and fast prediction. The better generalization capability of Extreme ANFIS model and RVR model could guarantee to tackle the changing characteristics of PV array with time. Hence, Extreme ANFIS model/ RVR model could identify and supply the reference MPP input to the controller at all possible system conditions with more accuracy, speed and less cost.

The MPP tracking of PV array is structured by the controller operating the boost converter switch, whose responsibility is to shift and maintain the panel's output power to the above identified optimal maximum power. With the motivation of achieving maximum efficiency many control schemes for maximum power point tracking of PV array has been emerged using fuzzy model based approach [158-161], genetic algorithm [162], Ripple-Based Extremum Seeking Control [163], Neural network based approach [164] etc.

Model Predictive Control is an advanced accurate control approach applicable to a wide range of practical applications due to multivariable handling nature and easy implementation [165, 166]. MPC has the skill to handle constrictions imposed on process inputs and outputs, process nonlinearities, dead times, overcoming disturbances and model uncertainties.

Designing an accurate nonlinear model and solving nonlinear optimization problem online are the tough tasks in nonlinear model predictive controller. A commonly accepted shortcoming of model predictive control is its capability to control systems with slow dynamics with sample time in the order of seconds or minutes. The sole reason for this short coming is the repetition of prediction and optimization at each sampling instant. A well recognized method for implementing fast MPC is by adopting the Finite Control Set Model Predictive Control technique.

FCS-MPC is a method, in which the optimization problem can be made simple by utilizing the discrete nature of power converters. In FCS-MPC method, a discrete model is used to predict the behavior of the system for every admissible actuation sequence up to the prediction horizon. The switching action that minimizes a predefined cost function is finally selected to be applied in the next sampling instant [167]. The main advantage of FCS-MPC lies in the direct application of the control action to the converter. Many successful applications of FCS-MPC method for power converters and drives are discussed in articles [93-100].

In the present work, the control of photo voltaic array maximum power point tracker is accomplished by Extreme ANFIS-MPC and RVR-FCS-MPC and SVR-FCS-MPC for the first time, which operates the DC converter switch combined with Extreme ANFIS/ RVR/ SVR based MPPT which identifies and supplies reference MPP input to the controller at different operating conditions. Thus the advantages of accurate prediction by Extreme ANFIS/ RVR/ SVR model and simplified optimization by FCS-MPC principle are utilized. A real time interaction of the Extreme ANFIS- MPPT/ RVR-MPPT/ SVR-MPPT which provides optimal reference trajectory to the Extreme ANFIS-MPC/ RVR-

FCS-MPC/ SVR-FCS-MPC based on instant solar energy, promotes better transient response under sudden alterations in solar irradiance when compared to state space model based MPC proposed by in [101].

5.2 STRUCTURE OF PHOTOVOLTAIC CELLS, MODULES AND ARRAYS

A photovoltaic cell performs photoelectric effect which is nothing but conversion of sunlight into electricity. Sunlight falling on the PV cell may be reflected, absorbed, or passed through; yet, electricity is generated only by the absorbed light. The electrons in the PV cell atoms acquire the energy of absorbed light. With the power of this energy, the energy acquired electrons gets interrupted from its standard locations in the atoms of PV material and supports the flow of electric current in the electrical circuit. "Built-in electric field," which is a particular electrical property of the PV cell offers the force or voltage necessary to drive the current via an external load [168]. Two layers of semiconductor material say p-type and N-type are kept in touch with each other, to encourage the built-in voltage inside a PV cell. N-type semiconductor material is the one which has electrons as the majority carriers and has negative electrical charge. The p-type semiconductor material is the one which has holes as the majority carriers and has positive electrical charge. In N-type semiconductor material electrons are the majority carriers and in P- type semiconductor material holes are the majority. Stuffing of these P-type and N-type material together generates a P-N junction at their boundary with an electric field. The P-N junction of a PV cell is shown in Fig. 5.1.

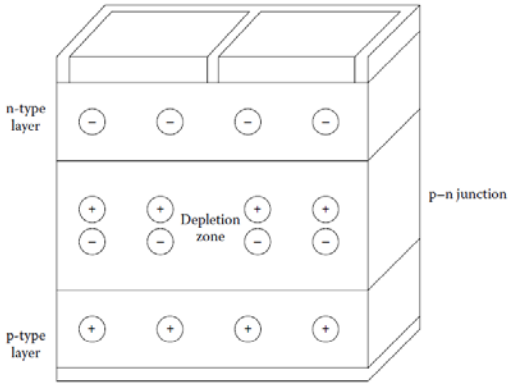

Fig. 5.1 P-N Junction of a Solar Cell

After the formation of a PN junction, excess electrons in the N-type material shift to the P-type material, and excess holes in the p-type side shift to the N- type side. The process builds up positive charge and negative charge along the sides of the N-type and P-type semiconductor material respectively. This movement of charges results in generation

of an electric field at the PN junction. This is due to the flow of negatively charged electrons and positively charged holes. The electrical field developed pushes the electrons towards the negative surface to carry current and holes towards the positive surface, where they stay for incoming electrons [168]. Power converters are essential to convert DC to alternate current (AC) for commercial applications. This electricity can also be stored in batteries, for upcoming use. All these are known as the "balance of system" (BOS) mechanism [169]. Combination of PV modules and BOS mechanism constructs the PV system. This PV system satisfies the energy demand of day to day life. Fig. 5.2 describes the schematic of a solar cell, solar module and solar array.

A PV or solar cell is the fundamental construction block of a PV or solar power system. A single PV cell is typically quite small and produces power of about 1 or 2W [169]. The output power is enhanced by connecting together many PV cells to form bigger unit called PV modules. Larger units called PV array is formed by interconnecting more number of PV modules together to generate more power. Serial connection of cells or modules results in the increase of output voltage and the parallel connection of cells or modules results in increase of output current.

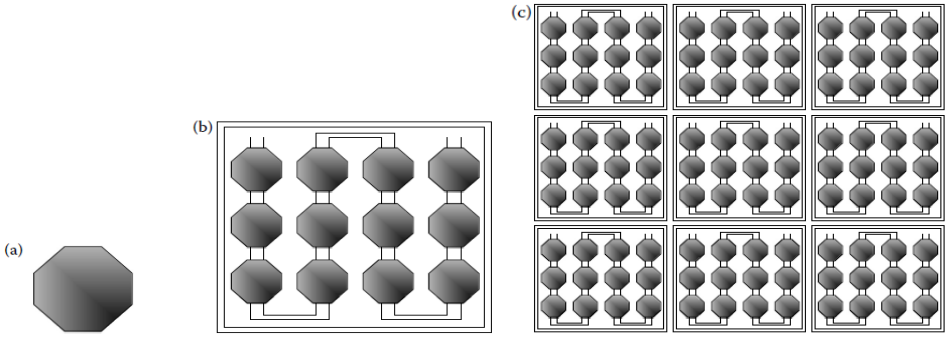

Fig. 5.2: (a) Solar Cell, (b) Solar Module, (c) Solar Array

Based on the method of collection of sunlight, PV systems come under two common groups: flat-plate type and concentrator type [169]. Flat panel type PV systems capture the sunlight either directly or use the diffused one from the surroundings. Concentrator systems collect the sunlight as a whole and concentrate by focusing the sunlight to target PV panels with the help of lenses and reflectors. By this means of absorbing the solar light, the effectiveness of the PV cell is improved.

5.3 STRUCTURE OF OVERALL SYSTEM

The structure of overall photo voltaic array MPPT system through NMPC strategy utilizing RVM regression model is shown in Fig 5.3. The PV array captures the lunar

radiation. The output of PV array is associated to DC-DC boost converter whose output is organized by RVM based model predictive controller through switch *s*, to trap maximum power from the PV module.

A MPPT based on another RVR model utilizes the PV module temperature and solar irradiance to predict the MPP current i^* and MPP voltage v^* to provide the desired reference trajectory to the RVM MPC. Thus the PV array voltage v_{pv}, PV array current $i_{pv,}$ reference current i^*, reference voltage v^*, converter output voltage v_c are provided as inputs to RVM based MPC to gather appropriate information at each sampling instant to either open the switch or close the switch *s* with binary output.

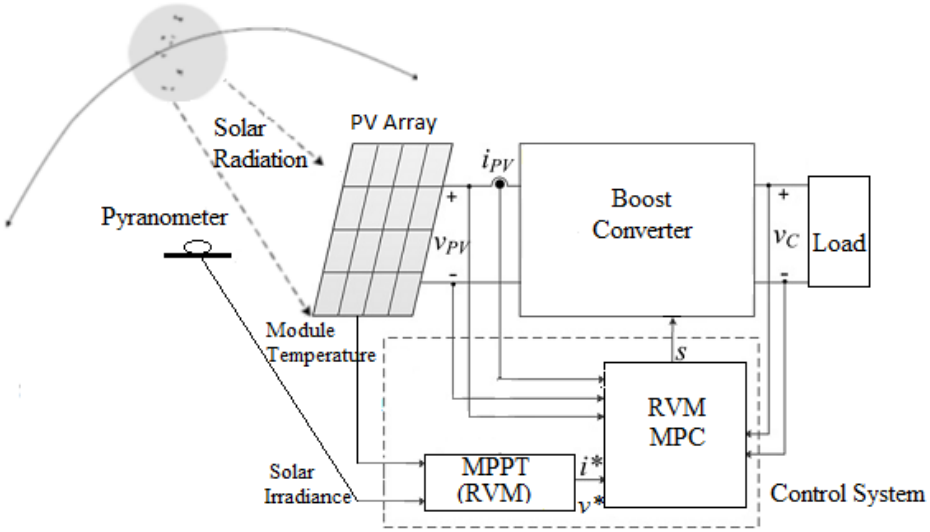

Fig. 5.3 Structure of overall PV array MPPT system through RVM-MPC strategy

The basic structure of overall photo voltaic array MPPT system through NMPC strategy utilizing Extreme ANFIS/ SVM regression model is obtained by replacing RVM model by Extreme ANFIS/ LS-SVM model in Fig. 5.3.

5.4 PV SYSTEM CONFIGURATION AND CHARACTERISTICS

A typical double diode PV model shown in Fig.5.4 is considered to examine the efficiency of the proposed NMPC. In double diode PV model two diodes are connected in parallel to get more accurate I-V characteristics than single diode PV model. The modeling of double diode PV array is derived by [170, 171]. The net output current of the PV cell is the difference between the photovoltaic current I_{ph} generated by solar irradiation and diode reverse saturation currents I_{D1} , I_{D2} as in (5.1)

$$I = I_{ph} - I_{01}\left\{\exp\left[\frac{V + I.R_s}{a_1.V_t}\right] - 1\right\} - I_{02}\left\{\exp\left[\frac{V + I.R_s}{a_2.V_t}\right] - 1\right\} - \frac{V + I.R_s}{R_p} \qquad (5.1)$$

where,

I	-	Net output current of solar cell
I_{ph}	-	photovoltaic current
I_{01},I_{02}	-	Diode reverse saturation currents
V	-	Terminal voltage
R_s	-	Series resistor
R_p	-	Parallel resistor
a_1,a_2	-	Diode constants
V_t	-	Thermal voltage

Photovoltaic current–voltage (*I–V*) curves shown in Fig. 5.5 are obtained by measuring the current produced by varying the load resistance, after maintaining the cell temperature constant. The *I–V* curve characteristically goes through two points:

- Short-circuit current (I_{SC}): I_{SC} is the current generated by short circuiting the output terminals of the solar cell or in other words current generated when the terminal voltage of the solar cell and load resistance are maintained zero.
- Open-circuit voltage (V_{OC}): V_{OC} is the voltage generated by open circuiting the output terminals of the solar cell or in other words voltage corresponding to zero current or infinite load resistance.

The I_{SC} and V_{OC} of the PV array considered is tabulated in Table 5.1

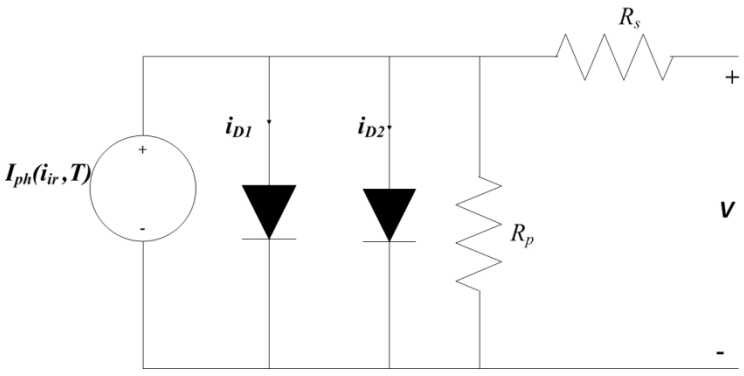

Fig. 5.4 Double diode PV model

Table 5.1 PV array details

Parameter	Values
Short-circuit current (Isc)	20.5 A
Open-circuit voltage (Voc)	66.5 V

The cell may be operated over a wide range of voltages and currents. By varying the load resistance from zero (a short circuit) to infinity (an open circuit), the MPP of the cell can be determined. On the *I–V* curve, the maximum power point occurs when the product of current and voltage is maximum. No power is produced at the short-circuit current with no voltage, or at the open-circuit voltage with no current. Therefore, MPP is somewhere between these two points. Maximum power is generated at about the "knee" of the curve. This point represents the maximum efficiency of the solar device in converting sunlight into electricity [172]. A PV system consists of many cells connected in series and parallel to provide the desired output terminal voltage and current. This PV system exhibits a nonlinear *I–V* characteristic [173]. The voltage-current and power-current characteristics of a PV array with specification given in Table 1 are shown in Fig. 5.5 and Fig.5.6.

The serial and parallel combination of solar cell constitutes photovoltaic array. A photovoltaic array containing multiple photovoltaic modules is meant for converting sunlight into usable electricity. A photovoltaic system supplying power for residential, commercial, or industrial energy supply generally contains an array of solar modules, power converters, electrical wiring, interconnections and mounting for other components.

While modeling the PV array, photovoltaic current $I_{ph,}$ diode reverse saturation currents I_{01} *and* I_{02} are multiplied by the number of solar cells linked in parallel and the thermal voltage V_t is multiplied by the number of solar cells linked in series. The V-I and I-P characteristics of the PV array are shown in Fig.5.5 and Fig. 5.6 respectively.

Ideally the maximum power point current corresponding to irradiance 1000 W/m^2 and 1200 W/m^2 are 18.6 A and 22.3 A respectively. Hence for a sudden change in irradiance from 1000 W/m^2 to 1200W/m^2, MPC for photovoltaic array MPPT system is anticipated to track MPP1 to MPP2 in Fig.5.6 which corresponds to the maximum power point reference currents 18.6 A and 22.3 A respectively.

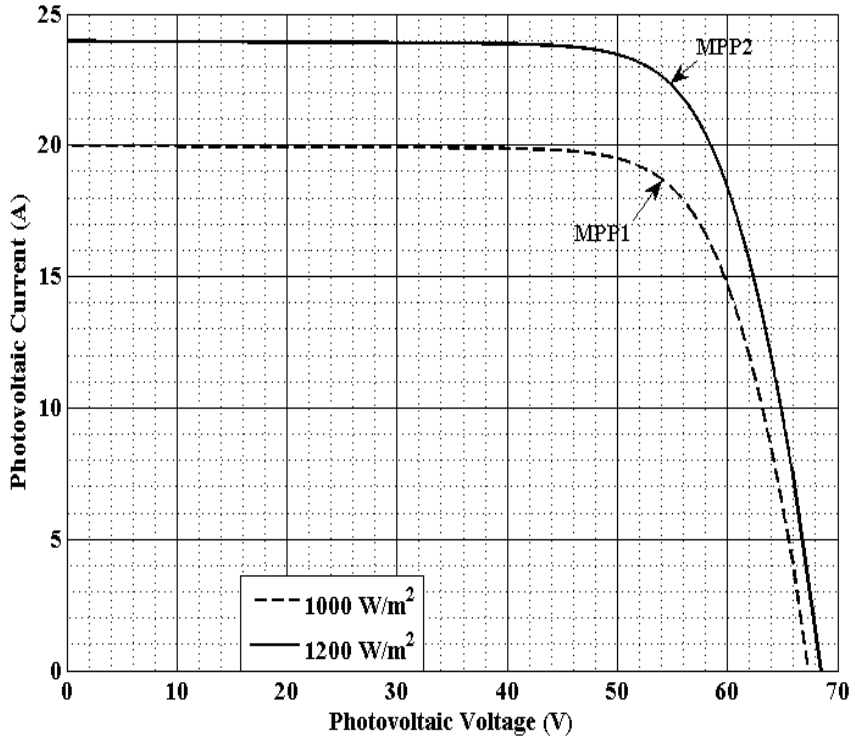

Fig. 5.5 V-I Characteristics of PV system for two irradiance levels

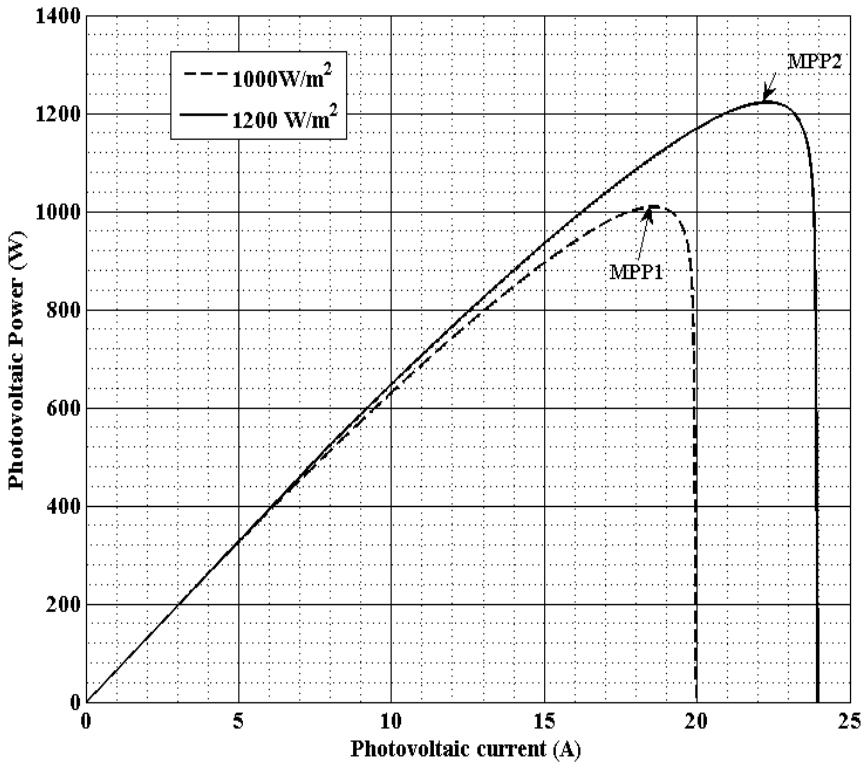

Fig. 5.6 I-P Characteristics of PV system for two irradiance levels

5.5 ANALYSIS OF PROPOSED CONTROL ALGORITHM

The proposed control methodology is based on Extreme ANFIS/ RVM/ SVM based finite control set model predictive controller, which operates the DC converter switch combined with Extreme ANFIS/ RVM/ SVM based MPPT which identifies and supplies reference maximum power point input to the controller. Thus, one Extreme ANFIS/ RVM/ SVM model duplicates PV array and the other model duplicates boost converter accompanying actual PV array output. The theory behind Extreme ANFIS, RVR and SVR are discussed in detail in chapter 3.

5.5.1 MPPT based on Extreme ANFIS, RVM and SVM models

Maximum power point tracking is a technique to trap the utmost possible power from solar panels. It is the task of the MPPT system to sample the output of the solar cells and apply suitable load to withdraw maximum power for any environmental condition.

The Extreme ANFIS/ RVM/ SVM model is employed to supply the controller with appropriate MPP reference inputs. For this present work, the data is obtained by simulating the proposed photovoltaic module in an open-loop system. The simulation is carried out at random irradiance constrained between 400 W/m^2 and 1200 W/m^2 and its corresponding maximum power point voltage and current values are recorded.

A sequence of 100 samples is used to train the sparse Bayesian RVR model offline. Hyperparameter estimation is carried out by Expectation Maximization (EM) updates on the objective function [22]. For this RVR model RBF kernel is used with the width parameter estimated automatically by the learning procedure [22] which improves generalization ability and reduces computational complexity of the training process. Also in the RVR model confidence intervals, likelihood values and posterior probabilities could be explicitly encoded easily. The SVM model involved in NMPC strategy is also trained offline using the same sequence of 100 samples by leave one out method. In SVR model the leave one out method is used to train offline using a sequence of 100 samples.

The Extreme ANFIS model involved in NMPC strategy is also trained offline using the same sequence of 100 samples. Three bell shaped membership functions are used for adapting parameters of ANFIS.

Fig. 5.7 and Fig. 5.8 correspond to the modeling results of RVR model and SVR model for reference current. The prediction error for training data of RVR model and SVR model are shown in Fig. 5.7. Also the prediction accuracy for random unseen testing data of RVR model and SVR model are shown in Fig. 5.8. They explain the little better prediction capability of RVR model compared to SVR model due to its better

generalization capability and model accuracy. The modeling results of Extreme ANFIS model is same as SVR model hence only the later is shown in Fig. 5.7 and Fig. 5.8.

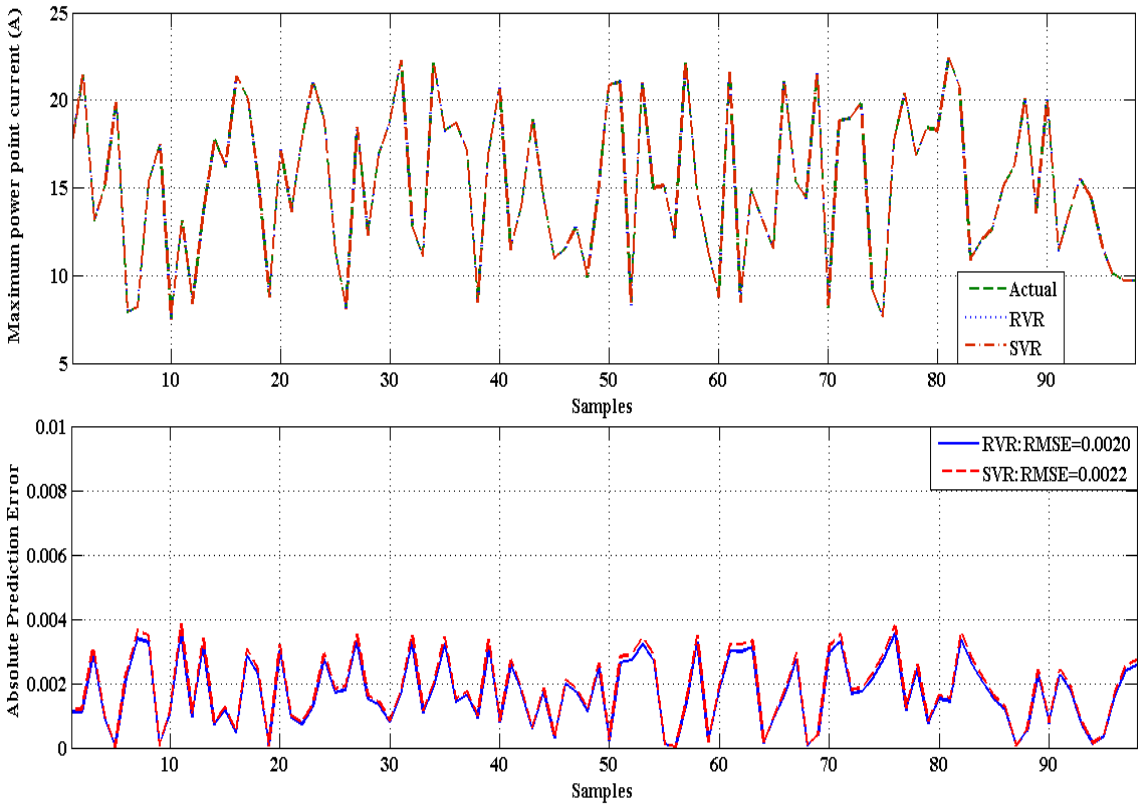

Fig. 5.7 Training performance of RVR/SVR model for maximum power point current

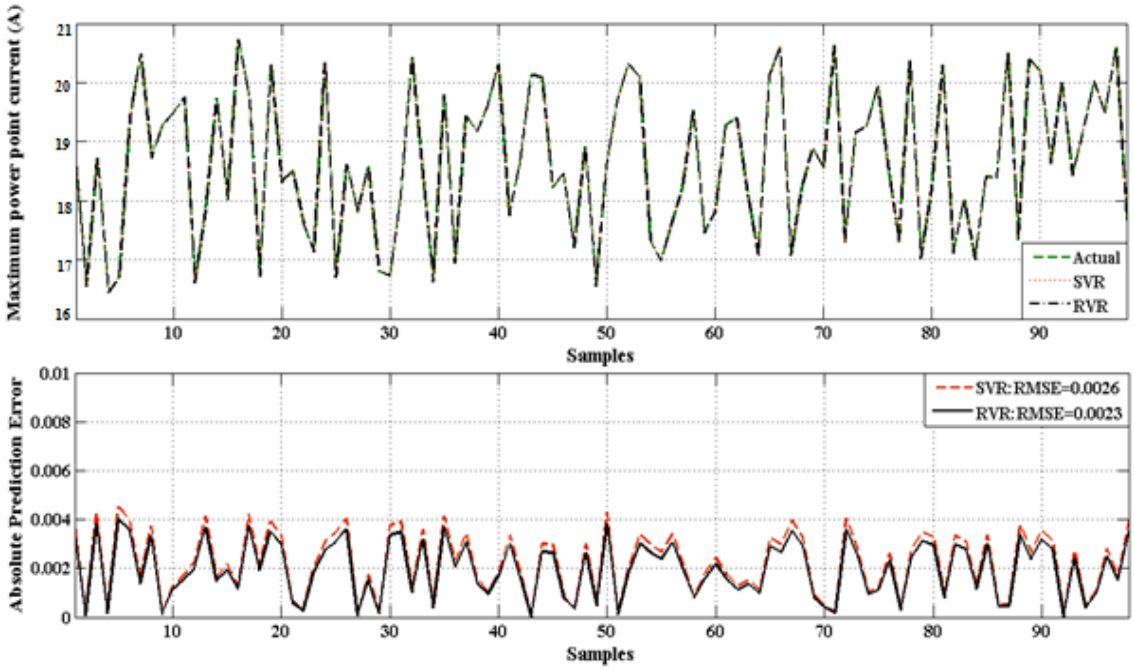

Fig. 5.8 Testing performance of RVR/SVR model for maximum power point current

The trained Extreme ANFIS model, RVR model and SVR model are further tested with 100 samples of random unseen inputs which are beyond the training data and there corresponding absolute prediction errors are shown in Fig. 5.8, which explores slightly better generalization capability of RVR model than Extreme ANFIS model and SVR model. Thus one can conclude that all the three empirical models Extreme ANFIS, RVR and SVR have good generalization capability.

5.5.2 Different machine learning techniques Based MPC Principle

The machine learning techniques employed here are used to predict *n* sampling instant *(k+n)* ahead behaviour of boost converter in order to implement control action at current sampling instant *k*. Thus MPC technique has the capability of understanding the forthcoming behaviour of the process and to take protective action at the current instant *k* itself in order to avoid undesirable offsets and oscillations at $(k+n)^{th}$ instant guaranteeing robustness to system performance.

Block diagram of RVM based MPC using FCS-MPC principle is shown in Fig. 5.9 which presents the FCS-MPC algorithm. It holds five main steps as summarized below [164]

1) Measure the instantaneous PV array voltage v_{pv}, PV array current $i_{pv,}$ reference current i^*, reference voltage v^*, converter output voltage v_c .
2) Predict the converter output voltage v_c and PV array current i_{pv} for the subsequent sampling instant for all the feasible switching states.
3) Compute the performance function for each prediction.
4) Choose the switching state corresponding to minimum performance function.
   Apply the latest switching state.

Extreme ANFIS/ SVM based MPC using FCS-MPC principle is attained by replacing RVM model in Fig. 5.9 by Extreme ANFIS/ SVM model.

Boost converter otherwise called as step-up converter is a DC-to-DC power converter which produces an output which is superior to its input. It is categorized under switched-mode power supply (SMPS) with one diode and one transistor as switches and an energy storing element either a capacitor or inductor, or the combination of both. Filters can be connected at the output side to reduce the ripples in the output voltage.

Power could be supplied to the boost converter from batteries, solar panels, rectifiers, DC generators etc. A DC to DC converter is one which alters one DC voltage to another DC voltage. A boost converter is otherwise called as step-up converter as it “steps up” the input voltage.

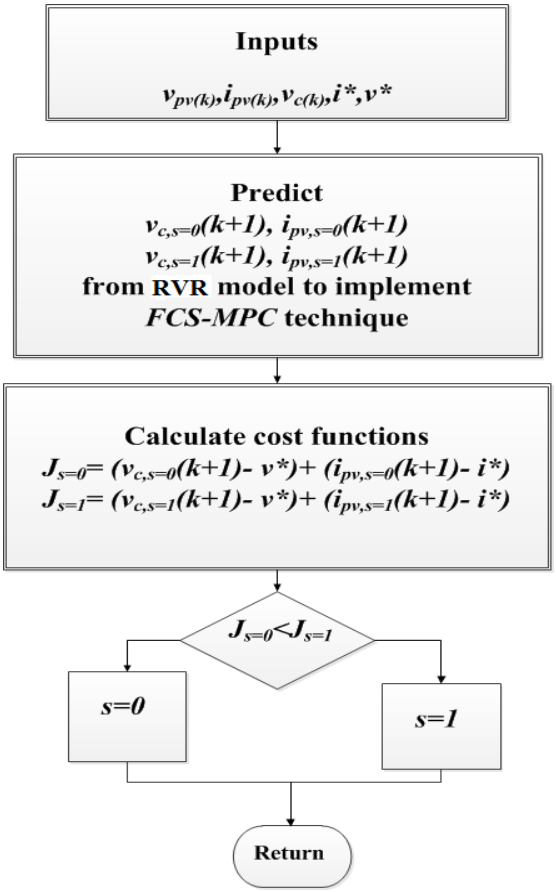

Fig. 5.9 Block diagram of RVM based MPC using FCS-MPC principle

The dynamic equations of DC-DC boost converter adopted here to boost the value of solar panel output voltage for utmost power withdrawal has the following two ideal switching states,

When switch is open, *s=1*, current will be reduced due to higher impedance. Therefore, modification or drop in current will be restricted by the inductor. Hence higher voltage is allowed to charge the capacitor.

$$\frac{di_{pv}}{dt} = \frac{1}{L}(i_{pv} + v_{pv}) \tag{5.2}$$

$$\frac{dv_c}{dt} = \frac{1}{C}(i_{pv} + \frac{1}{R}v_{pv}) \tag{5.3}$$

When switch is closed, *s=0*, the inductor stores the energy.

$$\frac{di_{pv}}{dt} = \frac{1}{L}v_{pv} \tag{5.4}$$

$$\frac{dv_c}{dt} = \frac{1}{RC}v_C \tag{5.5}$$

The boost converter presented in Fig. 5.10 is configured using *C= 50μF and L=20mH.*

For a nonlinear process the predicted output of different models is a function of past process outputs, *Y(k)=[y(k)…..y(k-n_y+1)]* and past process inputs, *U(k-1) = [u(k-1)…u(k-n_u+1)]*. The number of past controlled outputs and past manipulated inputs depends on the corresponding process orders n_u and n_y respectively.

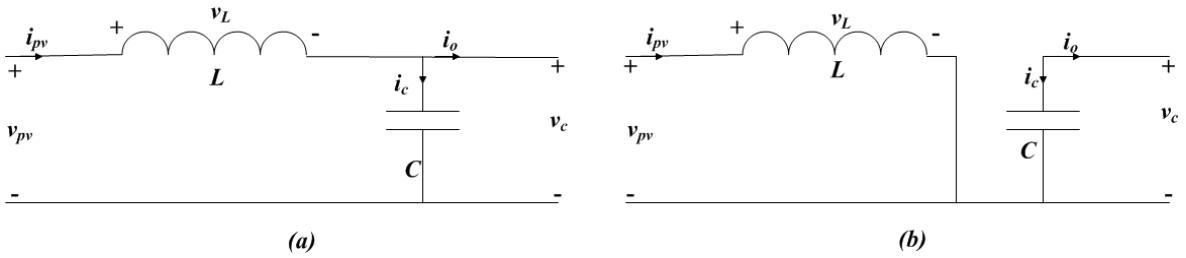

Fig. 5.10 Equivalent circuit of boost converter for two switching states (a) Open switch s=1 (b) Closed switch s=0

Thus a single step ahead performance prediction of the system output can be illustrated by the subsequent discrete time model,

$$\hat{y}(k+1) = f[y(k),....y(k-n_y+1),u(k),...,u(k-n_u+1)] \tag{5.6}$$

where *k* is the discrete time index

The above equation can also be written as

$$\hat{y}(k+1) = f[Y(k),u(k),U(k-1)] \tag{5.7}$$

Here, *Y(k)* and *U(k-1)* are the vectors holding past controlled outputs and past manipulated inputs respectively.

Thus after system identification of boost converter accompanying actual PV array outputs the one step ahead predicted converter output voltage v_c and the one step ahead predicted PV array current i_{pv} for their two possible switching states are given in (5.8) and (5.9) respectively.

$$\begin{aligned} v_{c,s=0}(k+1) &= w^T\varphi(0) \\ v_{c,s=1}(k+1) &= w^T\varphi(1) \end{aligned} \tag{5.8}$$

$$\begin{aligned} i_{pv,s=0}(k+1) &= w^T\varphi(0) \\ i_{pv,s=0}(k+1) &= w^T\varphi(1) \end{aligned} \tag{5.9}$$

In (5.8) and (5.9) $\varphi(0)$ and $\varphi(1)$ are the RBF kernel functions corresponding to manipulated variable (switch position) *s=0* and *s=1* respectively.

The performance index calculated for photo voltaic array MPPT system is,

$$J_{s=0} = (v_{c,s=0}(k+1) - v^*) + (i_{pv,s=0}(k+1) - i^*)$$
$$J_{s=1} = (v_{c,s=1}(k+1) - v^*) + (i_{pv,s=1}(k+1) - i^*) \quad (5.10)$$

FCS-MPC is the methodology incorporated to simplify the optimization problem and to track the right direction of P-V curve to extract and maintain maximum power. FCS-MPC procedure for one controlled variable i_{pv} is shown in Fig. 5.11. The reference trajectory offered by different models based MPPT is specified by dashed line. As the prediction horizon length is one the model predictive controller has to select the control action either S_0 or S_1 corresponding to minimum cost function J_0 or J_1 respectively.

The behaviour of both the controlled variables i_{pv} and v_c are entirely dependent on the instant dynamic output of operating PV module. In [101] a simplified equivalent equation describing the PV system is used and hence a regulation factor is involved to minimize prediction error. In this work, as the model used itself duplicates boost converter accompanying actual PV array output, the necessity for any equivalent equation describing PV system behaviour is neglected. Thus the prediction of boost converter performance is made simple, fast and accurate without any extra regulation factor.

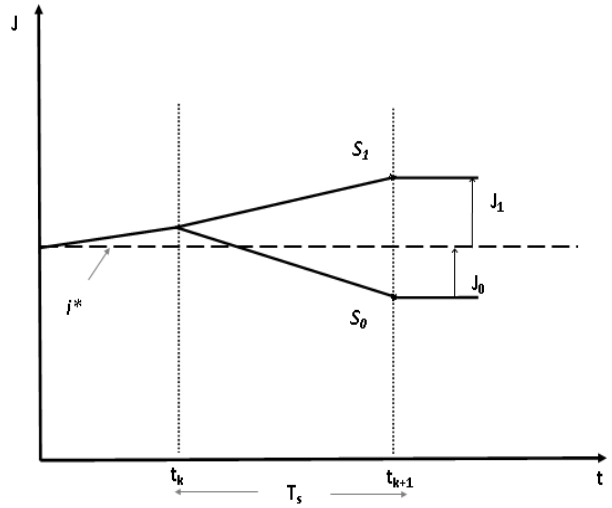

Fig. 5.11 FCS-MPC methodology

In [101] the horizon length is set to 2 in order to minimize estimation error which increases computation complexity. In this work, the horizon length is chosen to be 1 which minimizes computation burden and makes practical implementation of manipulated variable (switch position) very simple.

5.6 RESULTS AND DISCUSSIONS

This section describes the performances of PV array MPPT through RVM based NMPC strategy, Extreme ANFIS based NMPC strategy and SVM based NMPC strategy.

The performances of the proposed methods are better than space model based NMPC approach proposed by Kakasimos et.al [101].

In the proposed control system methodologies, RVM regression model, Extreme ANFIS model and SVM regression model plays vital role in providing reference inputs and accurate nonlinear model for executing MPC strategy.

Hence for a rapid change in irradiance from 1000 to 1200 W/m^2 the accurate RVR model, Extreme ANFIS model and SVR model with good generalization capability developed in this work as MPPT, has the capability to afford fast, fluctuation free, steady state reference input to the model predictive controller as revealed in Fig. 5.12 (The response of MPPT-RVR, MPPT-Extreme ANFIS and MPPT-SVR are similar and hence only the former is plotted). The reference input provided by modified incremental conductance algorithm proposed in [101] is slow with more oscillations.

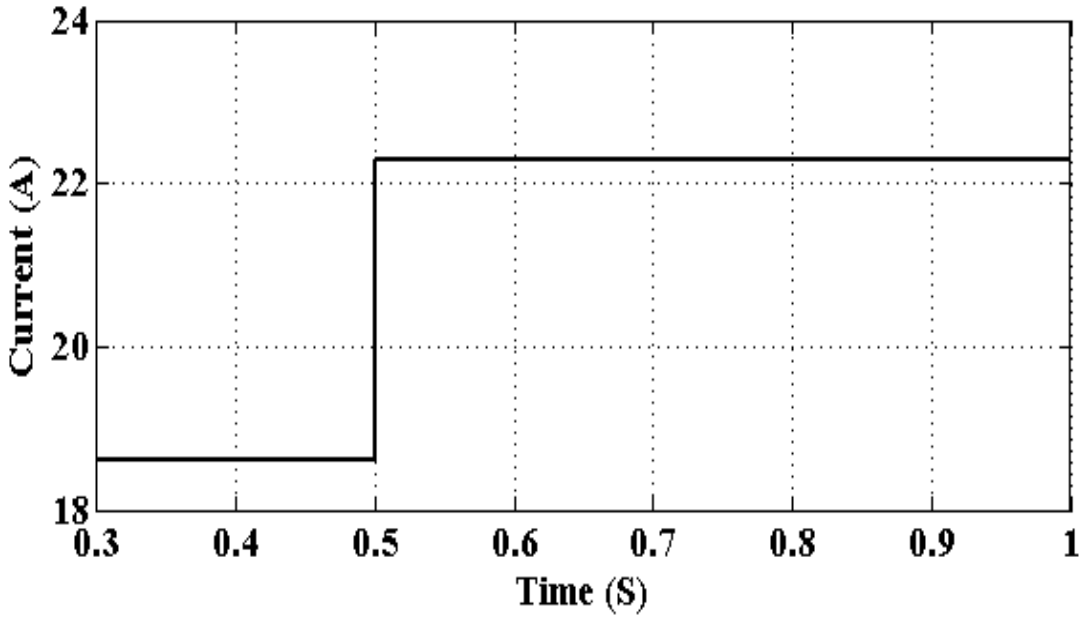

Fig. 5.12 Desired output current for irradiance variation from 1000 W/m^2 to 1200 W/m^2

Also, the accuracy and better generalization property of RVR model, Extreme ANFIS model and SVR model used in RVM based MPC, Extreme ANFIS based MPC and SVM based MPC respectively enables better tracking performance than state space model based MPC under any transient conditions. Thus in the proposed methodologies the PV array current i_{pv} tracks the accurate reference current i^* steadily with zero oscillations, which is much better than the state space model based MPC strategy proposed in [101] as shown in Fig.5.13 (The response of RVR-MPC Extreme ANFIS- MPC and SVR-MPC are similar and hence only the former is plotted). The PV array output power is shown in Fig.5.14 in

which the extracted output power is more efficient in RVM/ Extreme ANFIS/ SVM based MPC. The tracking efficiency corresponding to irradiance variation 1000 W/m^2 and 1200 W/m^2 are 99.85% and 99.91% respectively.

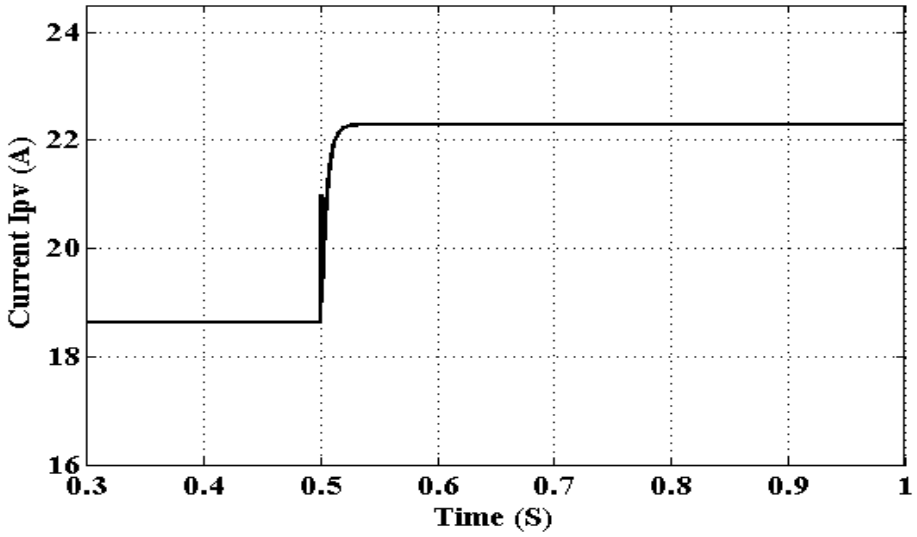

Fig.5.13 PV array current tracking performance for irradiance variation from 1000 W/m^2 to 1200 W/m^2

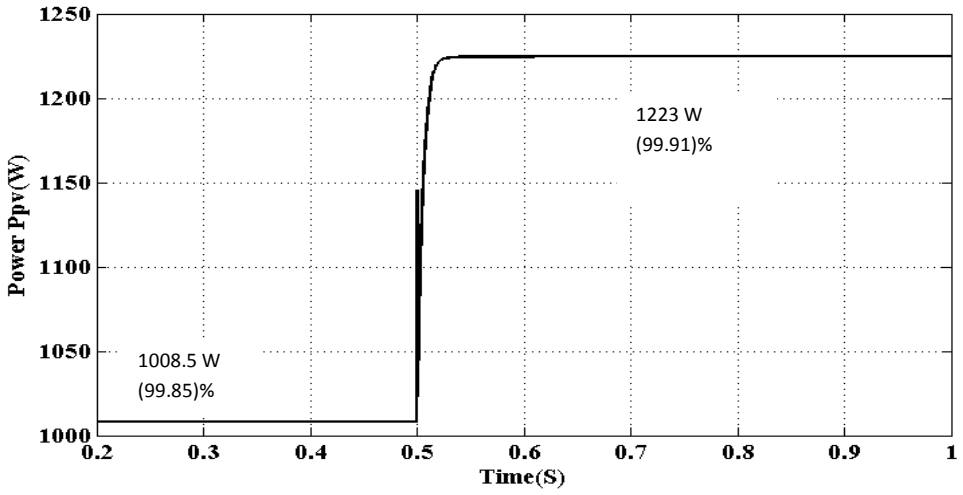

Fig.5.14 Overall power extracted from PV system by RVM based MPC under irradiance variation from 1000 W/m^2 to 1200 W/m^2

5.6.1 Tabulation of performance indices for different controlling techniques

This section enunciates the performance indices and sampling time of the proposed methodologies and state space model based MPC proposed by Kakasimos et.al. [101]. Table 5.2 and Table 5.3 show the number of support vectors/ relevance vectors and sampling time related to each controller for the simulation carried out.

The number of relevance vectors of RVR model is very less than the number of support vectors of SVM model which sharply reduces the computational time of RVR-

MPC. The overall simulation of RVM based MPC system and Extreme ANFIS based MPC system are carried out with a sampling time of 26μs and 27μs respectively, guaranteeing the capability of MPC control system to provide control action at every 26μs and 27μs correspondingly, which is sufficient for fast real time application. Thus the RVM based MPC and Extreme ANFIS based MPC has the capability to tackle the transient conditions intelligently with accuracy and fast convergence characteristics than LS-SVM based MPC and state space model based MPC [101] whose sampling time is 50μs.

Thus when compared to state space model based MPC, Extreme ANFIS based MPC and RVM based MPC performs better based on various attributes like better prediction accuracy, better generalization capability, better set point tracking performance and very less computation time. This makes it well suitable for industrial process control applications.

Table 5.2 Performance Indices of various control strategies

Control tactics	Number of Training Samples	Model	Number of Support Vectors/ Relevance Vectors	MPC Sampling time (Seconds)
LS-SVM-MPC	100	SVR	54	0.00005
Extreme ANFIS - MPC	100	Extreme ANFIS	-	0.000027
RVM-MPC	100	RVR	9	0.000026
Kakasimos et.al	-	State space	-	0.00005

Table 5.3 Performance Indices of various MPPT strategies

MPPT model	Number of Training Samples	Number of Support Vectors/ Relevance Vectors
LS-SVM	100	53
RVM	100	11

5.7 CONCLUSION

Novel simple nonlinear control strategies, RVM/ Extreme ANFIS/ SVM based MPC using FCS technique are proposed for PV system to accomplish maximum energy utilization. The accurate reference trajectory delivered by one RVR/ Extreme ANFIS/ SVR model based on instant real time solar energy empowers the NMPC to handle abrupt changes in solar irradiance more efficiently. The performances of the proposed simple algorithms are much better than the performance of state space model based MPC. Simulation results convey that such better performance is due to the better prediction accuracy of RVM/ Extreme ANFIS/ SVM model owing to its good generalization capability. The proposed controller does not require heavy computations; hence, implementation is feasible.

Chapter 6
NON LINEAR MODEL PREDICTIVE CONTROL OF A MULTI INPUT MULTI OUTPUT PROCESS

This chapter describes the control of highly nonlinear distillation column MIMO process using Relevance Vector Machines (RVM) regression model, Support vector regression model Extreme ANFIS model and NN model based MPC's. The optimization problem in the above control algorithm is made faster by particle swarm optimization with controllable random exploration velocity method of optimization. The simulation results comparing MPC using deterministic sparse kernel learning technique, probabilistic sparse kernel technique, neuro-fuzzy technique and NN technique are shown.

6.1 INTRODUCTION

Recent technological developments and ever-increasing demands of automation have necessitated the sophisticated control of complex systems [174]. The simulation and design of controller for higher order processes is a complicated task since the cost and complexity of the controller increases with system order [175]. High performance robust control and tracking control using nonlinear surface of the system is discussed in [176]. Model predictive control which is a subset of optimal control problem is a suitable, well efficient controller for highly nonlinear and interacting MIMO systems. Generally, in model based control approaches physical systems are analyzed and controlled by an accurate relevant model. Although an extensive research has been done on multivariable system modeling, identification and control most of them make use of linear demonstrations of physical systems, which shrinks the controllers operating region. This shortcoming could be overcome by the usage of nonlinear support vector regression models, relevance vector regression models, neuro-fuzzy models, neural network models.

Even though control of multivariable systems is the extension of the control of SISO systems, there are certain complicated tasks associated with it. The sole reason for the complications in MIMO control system is the presence of interaction among process variables. Hence the success of multi input multi output control strategy is evaluated based of its ability to deal with the consequences of interaction.

One among the widely used control for MIMO system is the PID Multi loop control which is a kind of control applicable for systems having minor interactions among process variables. It is the method of transformation of MIMO system to separated SISO systems. Here paring between controlled and manipulated variables is an important feature [177]. But if the interaction among process variables is strong, a decoupling component,

decoupler has to be designed. As a whole, all the separate control loops has to be tuned independently. This simple robust PID controller attracted most of the industries [178, 179]. Proper selection of manipulated variable pairing and accurate design of decoupler is necessary for efficient control using this PID controller. Hence a thorough knowledge of the process is must before designing the PID controller. As SVM/RVM based MPC's makes use of SVM /RVM model, thorough knowledge of the process is not necessary instead accurate input output data of the process is sufficient. This feature makes neural network/SVM/RVM model based MPC's attractive.

One among the significant reasons which makes MPC successful in industries is its capability to handle highly nonlinear MIMO process with ease. The key role of Model predictive controller is the minimization of the formulated performance index, which is the deviation of predicted output from the desired one. This minimization is achieved by a novel method of optimization, particle swarm optimization with controllable random exploration velocity described in chapter 4.

This chapter describes the simulation results of a highly nonlinear multi input multi output distillation column process with severe interacting process variables illustrates, the tracking performance of RVM based MPC with a radial basis function kernel, deterministic sparse kernel SVM based MPC with a radial basis function kernel, neuro-fuzzy technique and neural network based MPC are shown with relevant results.

6.2 BASIC STRUCTURE OF MPC FOR A MIMO SYSTEM

The general MPC configuration is shown in Fig. 6.1. It includes the blocks representing the MIMO system to be controlled, dynamic nonlinear model of the process, controller for control calculations. The variable residuals are equivalent to the disturbance affecting the process if the model is faultless. But practically none of the model is perfect hence, residuals is alike the effect of disturbance and model mismatch. The nonlinear dynamic model predicts the dynamic outputs of the plant based on systems past inputs and outputs. The predictions could be made for more than one time step ahead. Then the control calculations are done by the controller which is based on future predictions and current measurements. This control value is responsible for minimizing the deviation between the plant's output and set point. The feedback mechanism provided for the actual MIMO process compensates the error in the prediction which occurs because of the mismatch between the process and the model.

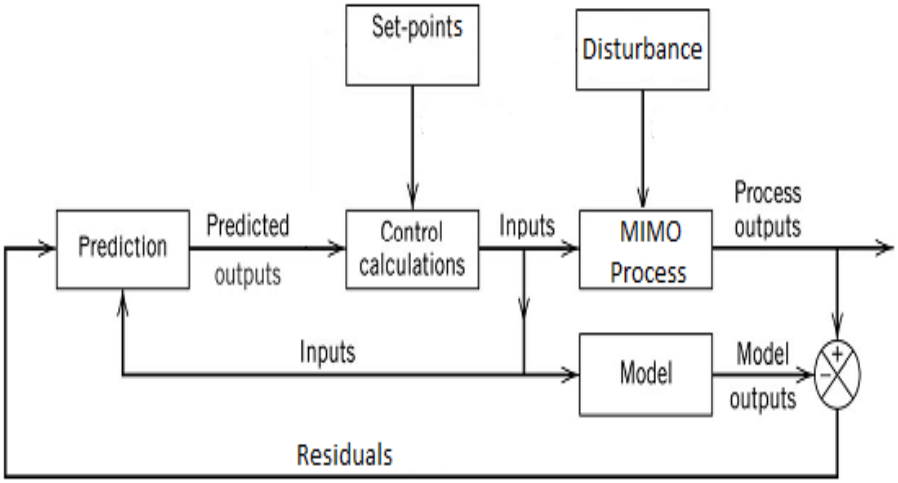

Fig. 6.1 MPC basic structure

6.2.1 Cost function formulation

For a MIMO $n \times m$ nonlinear process the predicted outputs of NN/ LS-SVM/ RVM model from is a function of past process outputs, $Y(k)=[y_1(k)\ldots..y_1(k-n_y+1), y_2(k)\ldots..y_2(k-n_y+1),\ldots..,y_m(k)\ldots..y_m(k-n_y+1)]$ and past process inputs, $U(k-1)=[u_1(k-1)\ldots u_1(k-n_u+1),u_2(k-1)\ldots u_2(k-n_u+1),....,u_n(k-1)\ldots u_n(k-n_u+1)]$. Which could be compactly rewritten as $Y(k)=[Y_1(k),Y_2(k)\ldots,Y_m(k)]$ *and* $U(k-1)=[U_1(k-1),U_2(k-1)\ldots..,U_n(k-1)]$. Here, $Y(k)$ and $U(k-1)$ are the vectors holding the past controlled outputs and past manipulated inputs respectively. The number of past controlled outputs and past manipulated inputs depends on the corresponding process orders n_u and n_y respectively.

Thus the prediction of m outputs for a MIMO $n \times m$ nonlinear process can be illustrated by the discrete time model given below,

$$\hat{y}_1(k+1) = f[Y_1(k), u(k), U(k-1)]$$
$$\hat{y}_2(k+1) = f[Y_2(k), u(k), U(k-1)] \qquad (6.1)$$
$$\ldots\ldots..$$
$$\hat{y}_m(k+1) = f[Y_m(k), u(k), U(k-1)]$$

where k is the discrete time index

The simple idea behind regression problem using sparse kernel learning structure is to project the input vectors by a nonlinear mapping into the high dimensional kernel Hilbert space and then to carry out linear regression in that feature space. Thus after system identification with the regression data set, the single step ahead prediction of each output using NN model could be formulated as,

$$y_j(k+1) = \sum_{i=1}^{hid} \{w_i f_i(net_i(k+1))\} + b \tag{6.2}$$

single step ahead prediction of each output using LS-SVM model could be formulated as,

$$\hat{y}_j(k+1) = \sum_{i=1}^{M} \alpha_i K(x_i, x) + b \tag{6.3}$$

single step ahead prediction of each output using RVM model could be formulated as,

$$\hat{y}_j(k+1) = \sum_{m=1}^{M} w_m \varphi_m(x) = w^T \varphi \tag{6.4}$$

where j=1…m and M is the subset of training samples.

Accordingly, the performance index to be minimized to achieve the optimal control sequence can be obtained as shown below,

$$J = \sum_{j=1}^{m} \sum_{i=N_1}^{N_2} q_j \left[ref_j(k+i) - \hat{y}_j(k+i)\right]^2 + \sum_{j=1}^{n} \sum_{i=1}^{N_u} \lambda_j \left[\Delta u_j(n+i)\right]^2 \tag{6.5}$$

In the performance index formulated in Equation (6.5) $\hat{y}$ depends on the kernel function which in turn is a function of manipulated variable u, which is optimized and applied to the actual plant to diminish the deviation between the desired value and controlled variable.

N_1	-	minimum prediction horizon
N_2	-	maximum prediction horizon
N_u	-	control horizon
m	-	number of outputs
n	-	number of inputs
ref(.)	-	reference trajectory
$\hat{y}_j(.)$	-	j^{th} predicted output of the model
$\Delta u_j(.)$	-	change of j^{th} control input defined as $u_j(k+i)$-uj(k+i-1)
k	-	current sampling instant
q_j, λ_j	-	time independent weighting coefficients.

6.3 BINARY DISTILLATION COLUMN PROCESS

This section describes the better accuracy and less computational demand of LS-SVM based NMPC than NN based NMPC by simulating a binary distillation column.

The arrangement of distillation column process for the separation of a binary mixture of methanol and n-propanol is shown in Fig. 6.2. Two conventional controllers

represented by *LC* are used to retain the levels in the reflux and bottom product tanks. The MPC algorithm is responsible for controlling the composition of top product x_D and bottom product x_B by manipulating the reflux stream flow rate, *L* and vapour stream flow rate, *V*. Two significant controller performance characteristics of desired set point tracking and disturbance rejection are presented by suitable simulations.

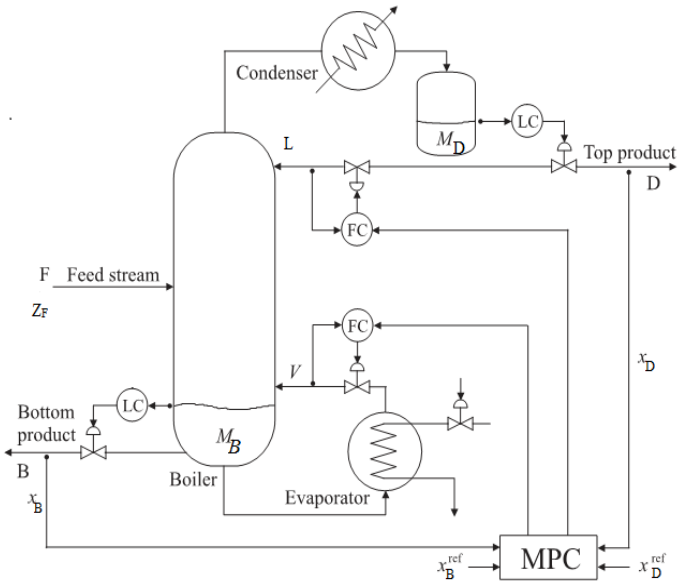

Fig. 6.2 Schematic of the binary distillation column process

The binary distillation column considered is under *LV* –configuration [180]. It exhibits severe nonlinearity and strong cross coupling both under steady state and dynamic operating conditions. Simulation results convey the suitability of NMPC to tackle this nonlinearity and cross coupling.

The fundamental model containing the following nonlinear differential equations is used as the real process during simulation. The molar flows, relative volatility, liquid holdup on all trays are assumed to be constant. Mixing on all stages is perfect and vapour holdup is assumed to be nil.

The important notations of the distillation column are listed below,

F	-	Feed rate [kmol/min]
q_F	-	Fraction of liquid in feed
D and B	-	distillate and bottom product flow rate [kmol/min]
x_D and x_B	-	distillate and bottom product composition
L	-	reflux flow [kmol/min]
V	-	boilup flow [kmol/min]
M_B	-	Liquid holdup on reboiler [kmol]

M_D	-	condenser holdup [kmol]
M_i	-	Liquid holdup on theoretical tray *i* [kmol]
N_T	-	total number of trays
N_F	-	location of Feed tray from bottom
Q_F	-	fraction liquid in feed
L_B	-	Liquid flow rate into reboiler
V_T	-	vapour flow rate on top tray
X_B	-	$\ln x_B$, logarithmic bottom composition
Y_D	-	$\ln (1-y_D)$, logarithmic top composition
x_i	-	liquid mole fraction of light component on stage i
y_i	-	vapour mole fraction of light component on stage i
y_T	-	vapour mole fraction of light component on top tray
Z_F	-	mole fraction of light component in feed
x_D^{ref}	-	desired value of distillate product composition
x_B^{ref}	-	desired value of bottom product composition

Material balance equations for change in holdup of light component on each tray;

$$i = 2, N_T \ (i \neq N_F, \ i \neq N_F + 1) \quad M_i \dot{x}_i = L_{i+1} x_{i+1} + V_{i-1} y_{i-1} - L_i x_i - V_i y_i \tag{6.6}$$

above feed location $i = N_F + 1$

$$M_i \dot{x}_i = L_{i+1} x_{i+1} + V_{i-1} y_{i-1} - L_i x_i - V_i y_i + F_v y_F \tag{6.7}$$

below feed location, $i = N_F$

$$M_i \dot{x}_i = L_{i+1} x_{i+1} + V_{i-1} y_{i-1} - L_i x_i - V_i y_i + F_L x_F \tag{6.8}$$

reboiler, $i = 1$

$$M_B \dot{x}_i = L_{i+1} x_{i+1} - V_i y_i - B x_i, \qquad x_B = x_1 \tag{6.9}$$

total condenser, $i = N + 1$

$$M_D \dot{x}_i = V_{i-1} x_{i-1} - L_i x_i - D x_i, \qquad y_D = x_{N_T+1} \tag{6.10}$$

VLE on each tray, $(i = 1, N_T)$, constant relative volatility

$$y_i = \alpha x_i / (1 + (\alpha - 1) x_i) \tag{6.11}$$

Flow rates above and below feed trays with constant molar flow rates are,

$$i > N_F \text{ above feed, } L_i = L, \ V_i = V + F_V \tag{6.12}$$

$$i \leq N_F \text{ below feed, } L_i = L + F_L, V_i = V \tag{6.13}$$

$$F_L = q_F F \quad , F_V = F - F_L \tag{6.14}$$

condenser holdup is kept constant,

$$D = V_N - L = V + F_V - L \tag{6.15}$$

reboiler holdup is kept constant,

$$B = L_2 - V_1 = L + F_L - V \tag{6.16}$$

Vapour phase and liquid phase composition of the feed x_F, y_F respectively are obtained by solving the equations below.

$$FZ_F = F_L x_F + F_V y_F \tag{6.17}$$

$$y_F = \alpha x_F / (1 + (\alpha - 1)\, x_F) \tag{6.18}$$

6.4 NEURAL NETWORK BASED MPC OF DISTILLATION COLUMN PROCESS

This section describes NN based MPC using PSO-CREV optimization algorithm. In order to develop an accurate model accurate data set is necessary and is generated by simulating the Matlab Simulink model of distillation column. Those recorded data set is meant for learning, validating and testing the model. The learned, validated and tested model is then incorporated in MPC to provide accurate predictions of process dynamics.

6.4.1 Training and testing the model

The dynamic model of the binary distillation column is simulated open loop to collect the training and testing data. The simulation is carried out at random constrained reflux flow and boilup flow and its corresponding distillate and bottom product compositions are recorded. The constraint to the input signals, reflux flow and boilup flow are $2.5 \leq u1(t) \leq 2.9$ and $3 \leq u2t) \leq 3.5$ respectively. The binary distillation column model considered under LV- configuration contains a total of 41 stages including the reboiler and total condenser. Thus the dynamic model contains 41 nonlinear differential equations. In order to capture the above order of dynamics using NN model two past outputs and past inputs are sufficient hence a second order model is chosen.

A two layer feed forward neural network comprising of five hidden neurons with sigmoid activation function and a single output neuron with linear activation function is used as the model. In Neural network based nonlinear MPC, for offline training the multilayer feed forward neural network a sequence of 1000 samples with two delay regression vector format are used and is done through Levenberg-Marquardt learning

algorithm. The identification performance of NN method of approximation is assessed by the root mean square error (RMSE) performance function in (6.19).

$$RMSE = \left\{ \sum_{k=1}^{N} [\hat{y}(k) - y(k)]^2 \Big/ N \right\}^{1/2} \tag{6.19}$$

where $\hat{y}(k)$ - output of the model for k[th] sampling instant,

$y(k)$ - output of the plant for k[th] sampling instant,

N - total number of samples.

Fig. 6.3 and Fig. 6.4 correspond to the modeling results of NN methods. The trained model is further tested with 100 samples of data which are beyond the training data. The graph of prediction errors of NN model for test data beyond the training data are shown in Fig. 6.4, which explores the poor generalization capability of NN model with lengthy training time. Accuracy of the model in terms of RMSE (6.18) is tabulated in Table 6.1.

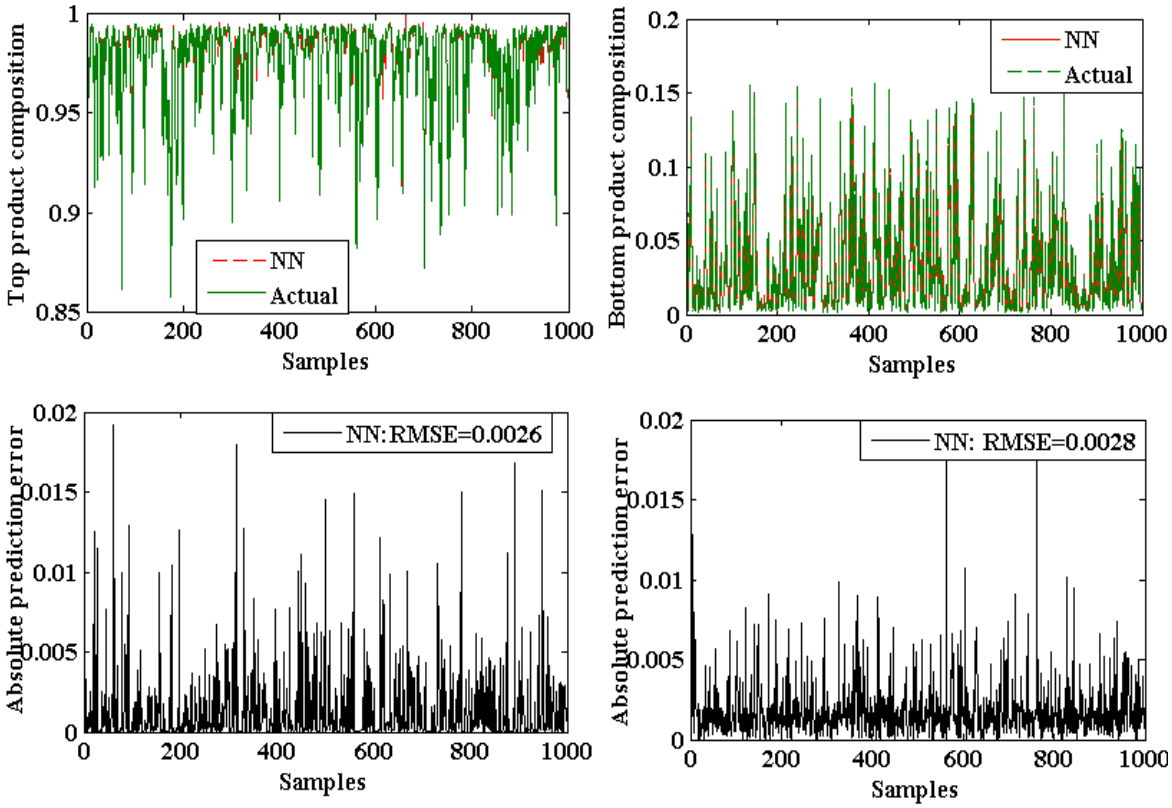

Fig. 6.3 Training performance of NN models

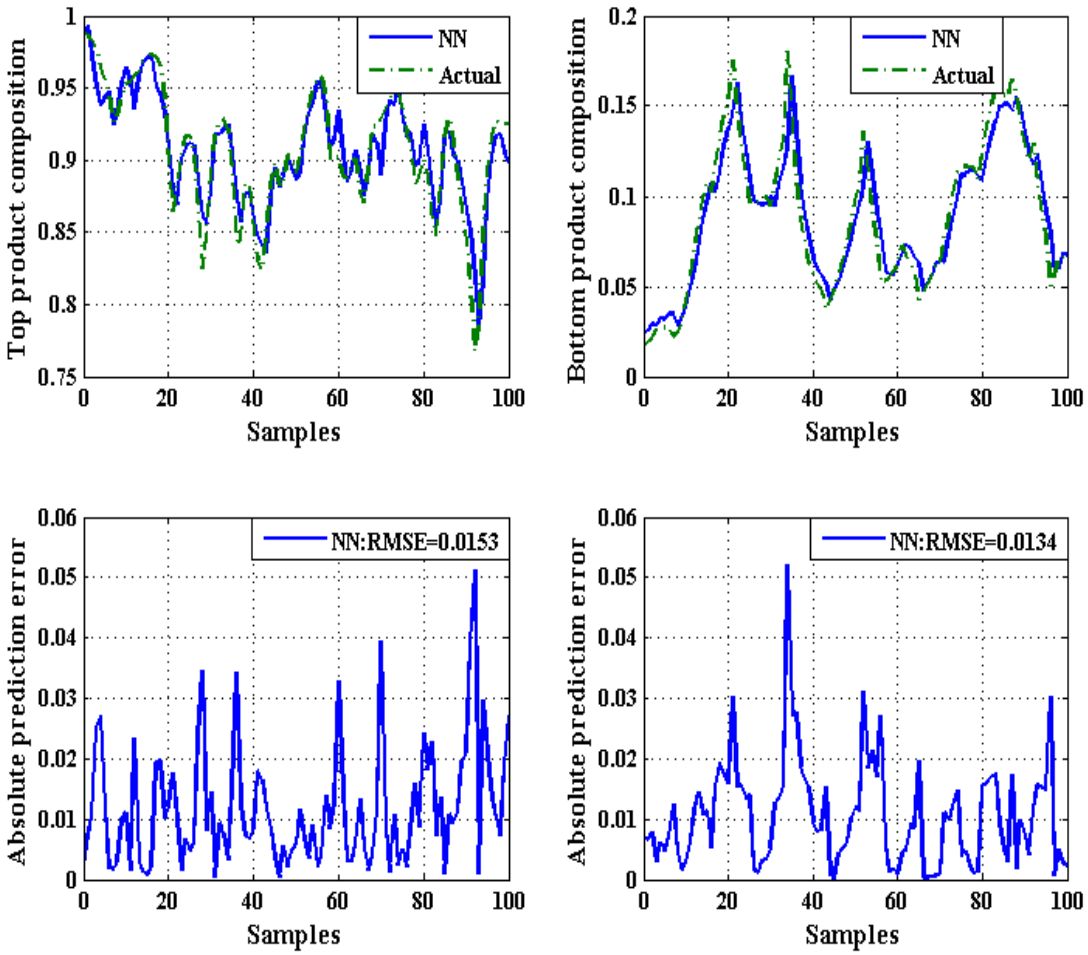

Fig. 6.4 Testing performance of NN models

6.4.2 Performance of NN-PSO-CREV-MPC

The offline trained, validated and tested NN model is then used as the nonlinear model for nonlinear MPC. Fig. 6.5 illustrates the random set point tracking performances of NN model based MPC. The tracking performance of NN based MPC, is with more oscillations due to the presence of strong interacting process variables.

Also as the PSO-CREV algorithm converges to the finest solution at each sampling instant the manipulated variables reflux flow rate, *L* and boilup flow rate, *V* corresponding to NN-PSO-CREV are with smooth fluctuations as shown in Fig.6.6 presenting the index of control performance.

The unmeasured disturbance rejection capability of NN-PSO-CREV based MPC is verified by subjecting the distillation column process with dissimilar magnitudes of disturbance at different sampling instants as shown in Fig. 6.7. The control variables, Reflux flow rate, *L* and boilup flow rate, *V* with disturbances at different sampling instant (indicated by dashed circles) are revealed in Fig. 6.7.

The MPC parameters for the best control of distillation column process are given below.

Minimum prediction horizon=1

Maximum prediction horizon=5

Control horizon=1

Penalty factor on differenced control signal =0.005

Control input 1 constraint $2.5 \leq u1(t) \leq 2.9$

Control input 2 constraint $3 \leq u2t) \leq 3.5$

The unmeasured disturbance rejection performance of NN-PSO-CREV based MPC as shown in Fig. 6.8. The controlled variables, top product composition and bottom product composition settles down slowly with oscillations after getting affected by unmeasured disturbances.

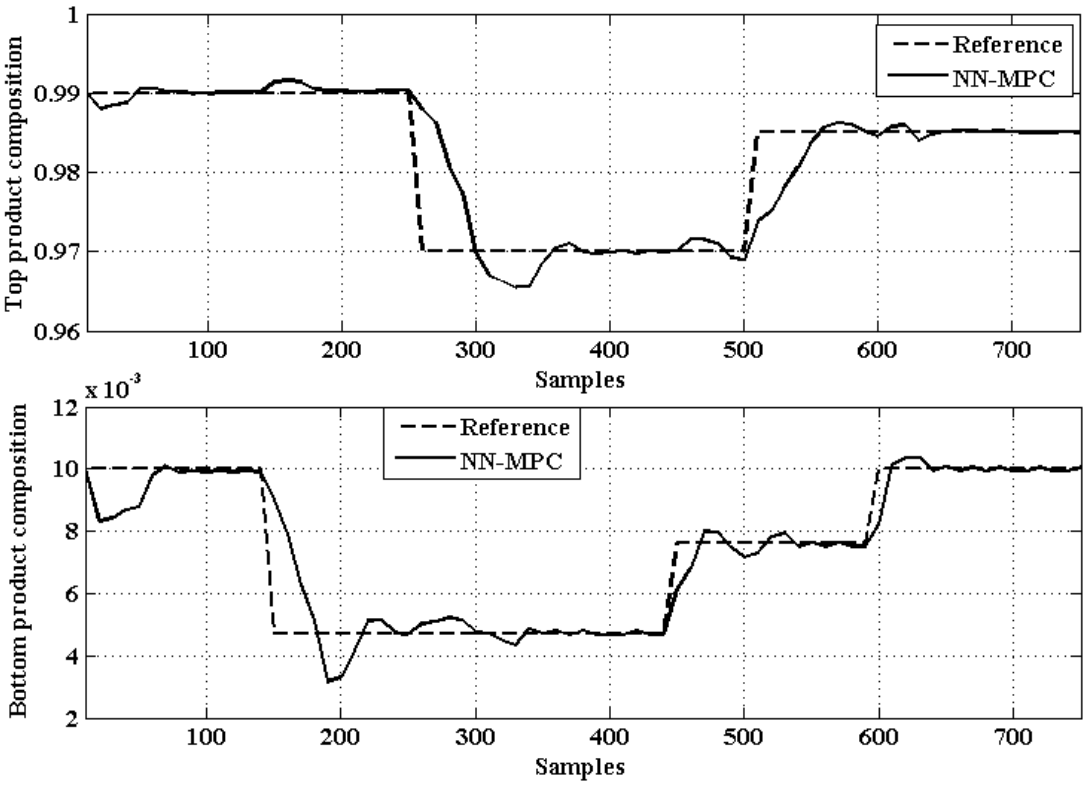

Fig. 6.5 Set point tracking performance of distillation column process by NN-MPC

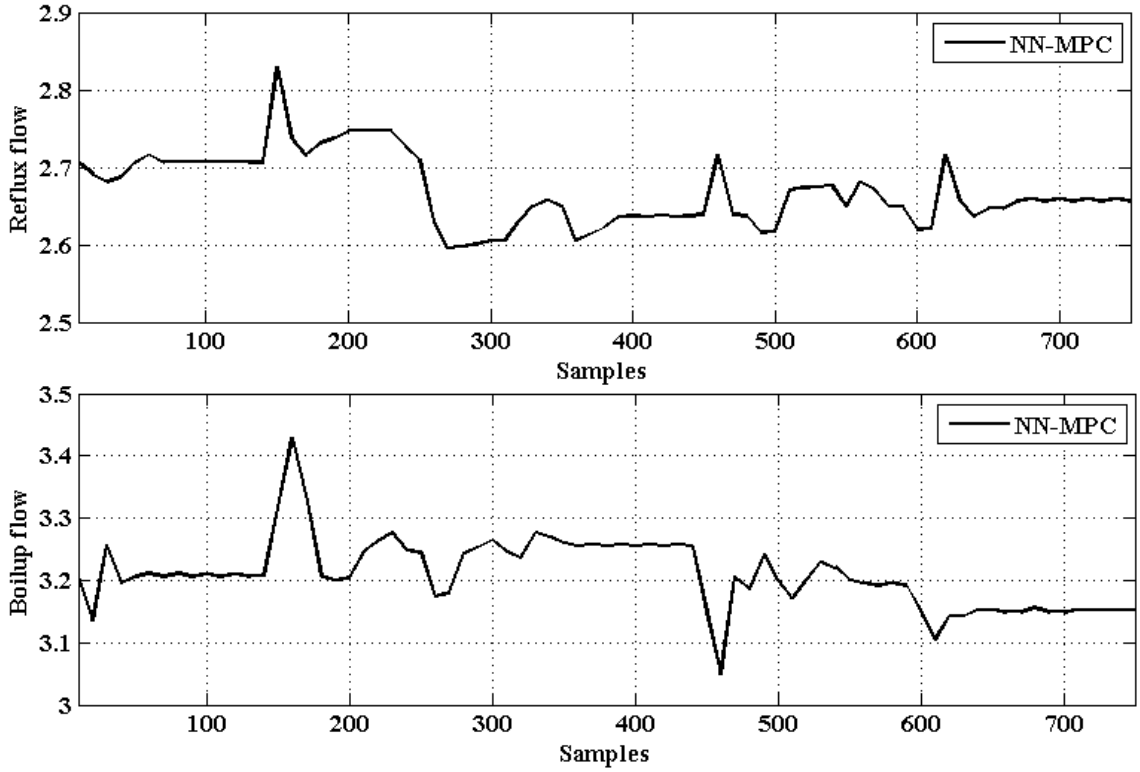

Fig. 6.6 Modifications in the process variables for tracking the top product and bottom product compositions of distillation column process

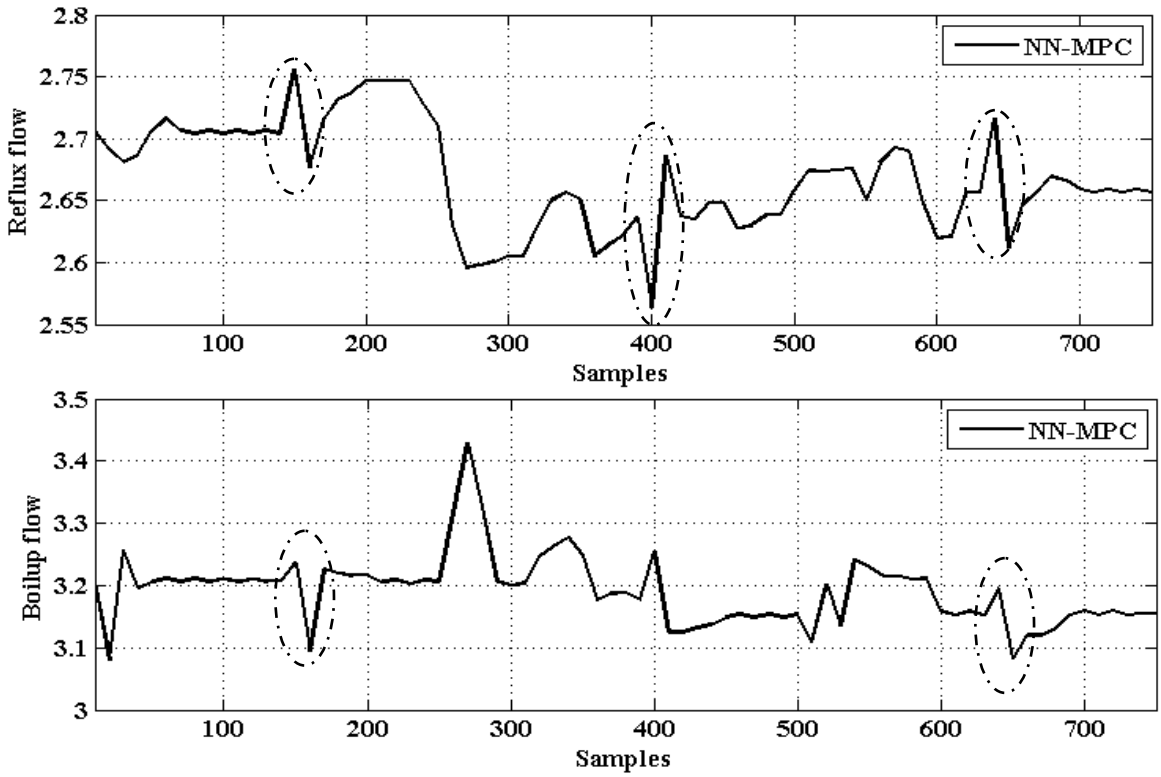

Fig. 6.7 Modifications in the process variable to illustrate unmeasured disturbance

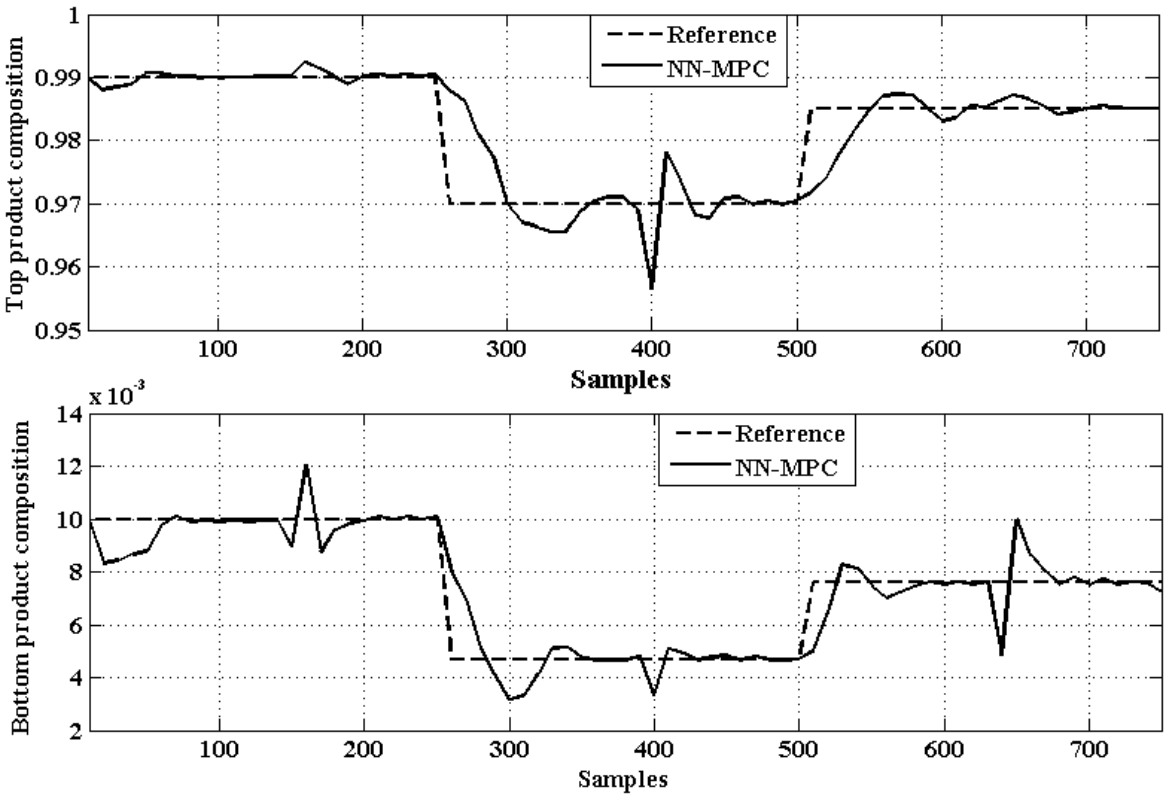

Fig. 6.8 Performance of unmeasured disturbance rejection

6.5 SVM BASED MPC OF BINARY DISTILLATION COLUMN PROCESS

This section describes SVM based MPC using PSO-CREV optimization algorithm for the multi input multi output binary distillation column process.

6.5.1 Training and testing the model

A sequence of 100 samples is used to train the SVR model offline using the leave one out method discussed in chapter 3. Figure 6.9 and Figure 6.10 corresponds to the modeling results of SVR models. The identification performance of support vector regression model is assessed by RMSE performance function in (6.20).

$$RMSE = \left\{ \sum_{k=1}^{N} [\hat{y}(k) - y(k)]^2 \Big/ N \right\}^{1/2} \tag{6.20}$$

where $\hat{y}(k)$ represents the model's output for the sampling instant k, $y(k)$ represents the plant's output for the sampling instant k and N represents total number of samples.

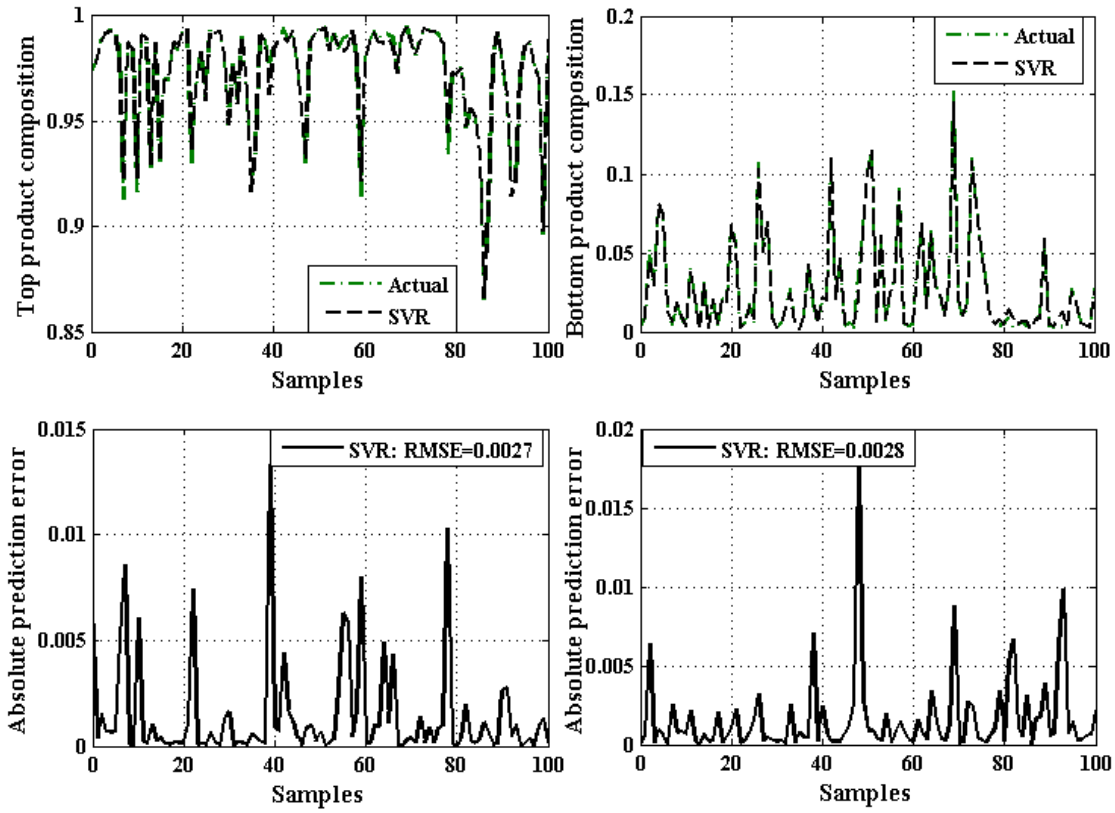

Fig. 6.9 Training performance of SVR models

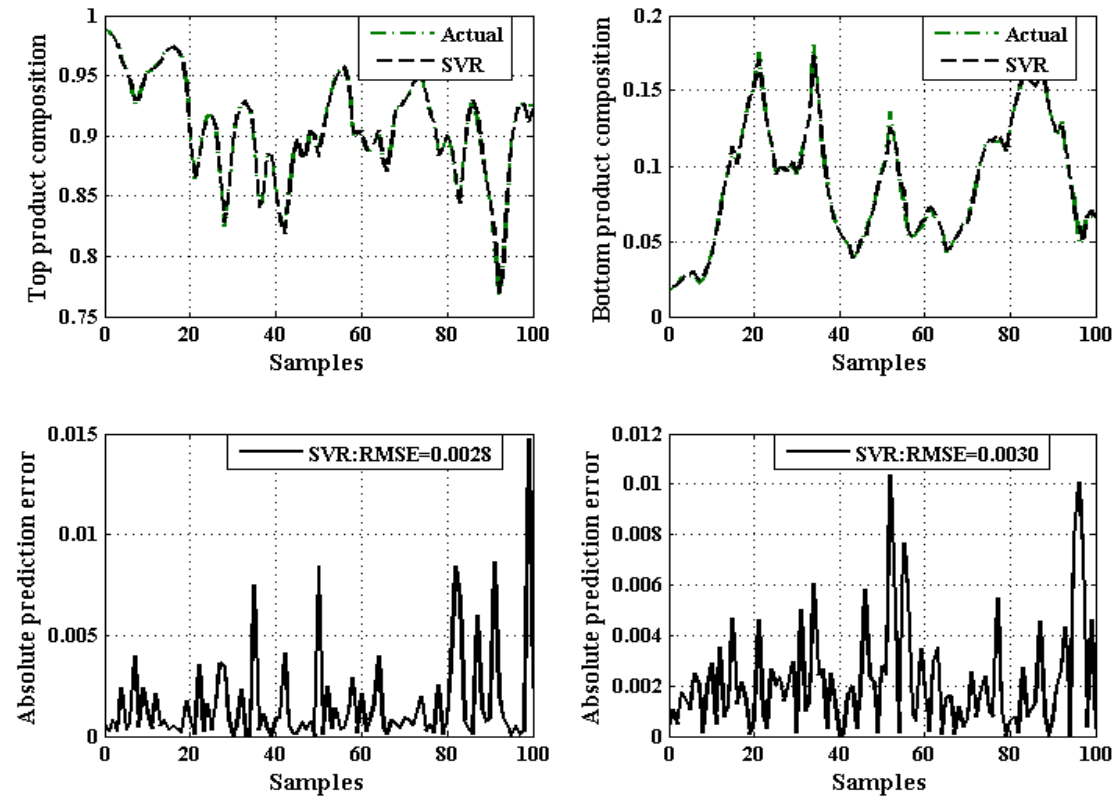

Fig. 6.10 Testing performance of SVR models

The RMSE values corresponding to the training errors of Support Vector Regression models of top product and bottom product compositions are 0.0027and 0.0028 respectively. The trained SVR models are further tested with 100 samples of random inputs which are beyond the training data and there corresponding absolute prediction errors are calculated.

The prediction errors of SVR models for 100 different samples are shown in Figure 6.18, which explores the good generalization capability of SVR models. Thus one can conclude that SVM based empirical modeling behaves suitably for industrial process control with good generalization.

6.5.2 Performance of SVM-PSO-CREV-MPC

The offline trained and tested SVM model is then used as the nonlinear model for nonlinear MPC. Fig. 6.11 illustrates the random set point tracking performances of SVR based MPC. The SVM based MPC performs well even in the presence of severe interacting process variables.

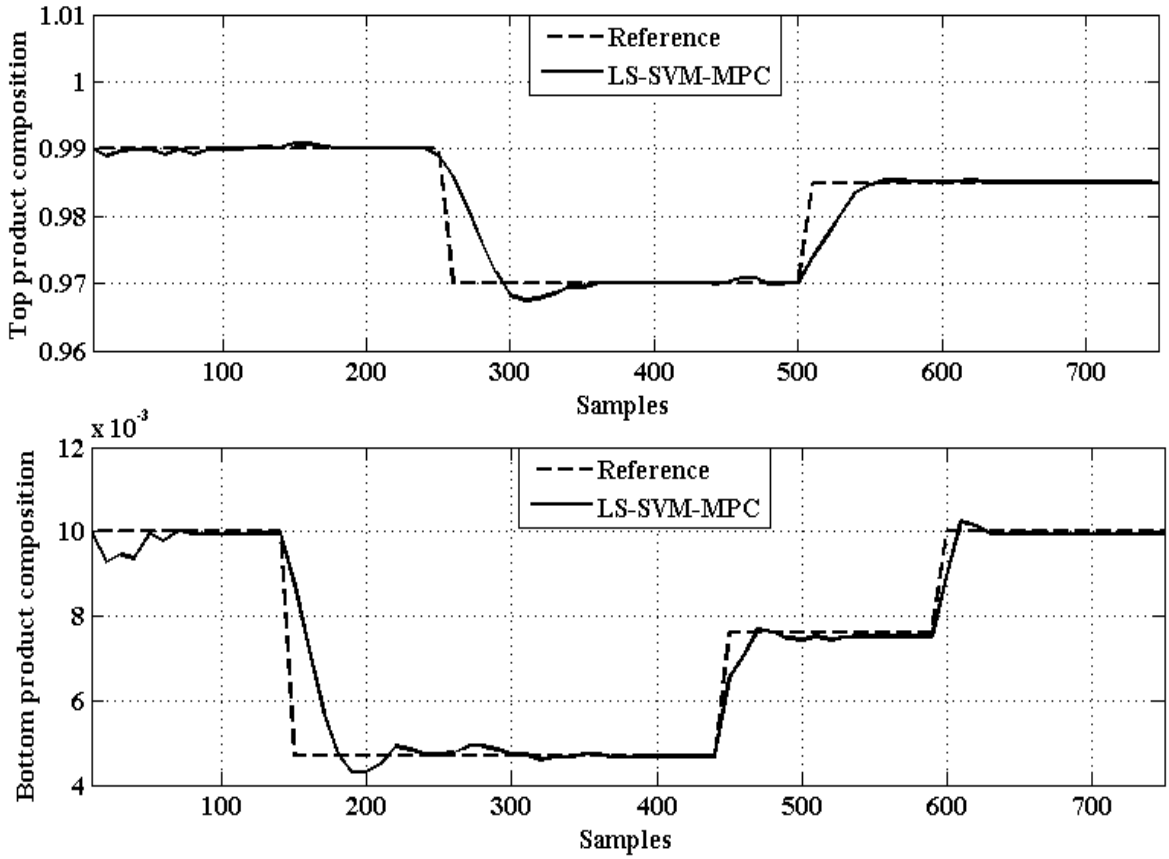

Fig. 6.11 Set point tracking performance of distillation column process by SVM-MPC

Also as the PSO-CREV algorithm converges to the finest solution at each sampling instant the manipulated variables reflux flow rate, *L* and boilup flow rate, *V* corresponding to SVM-PSO-CREV-MPC are with very less fluctuations as shown in Fig.6.12 presenting the index of control performance.

The unmeasured disturbance rejection capability of SVM-PSO-CREV based MPC is examined by subjecting the distillation column process with dissimilar magnitudes of disturbances at different sampling instants.

The control variables, boilup flow rate, V and Reflux flow rate, L with disturbances (indicated by dashed circles) at different sampling instant are shown in Fig. 6.13. Both the controlled variables, top and bottom product compositions settle down smoothly with very less oscillations after getting affected by unmeasured disturbances as shown in Fig. 6.14.

Thus the better capability of SVR based MPC; in overcoming the interaction among process variables is vibrant from the simulation results. Accordingly SVR-PSO-CREV based MPC behaves suitably for process control industrial applications.

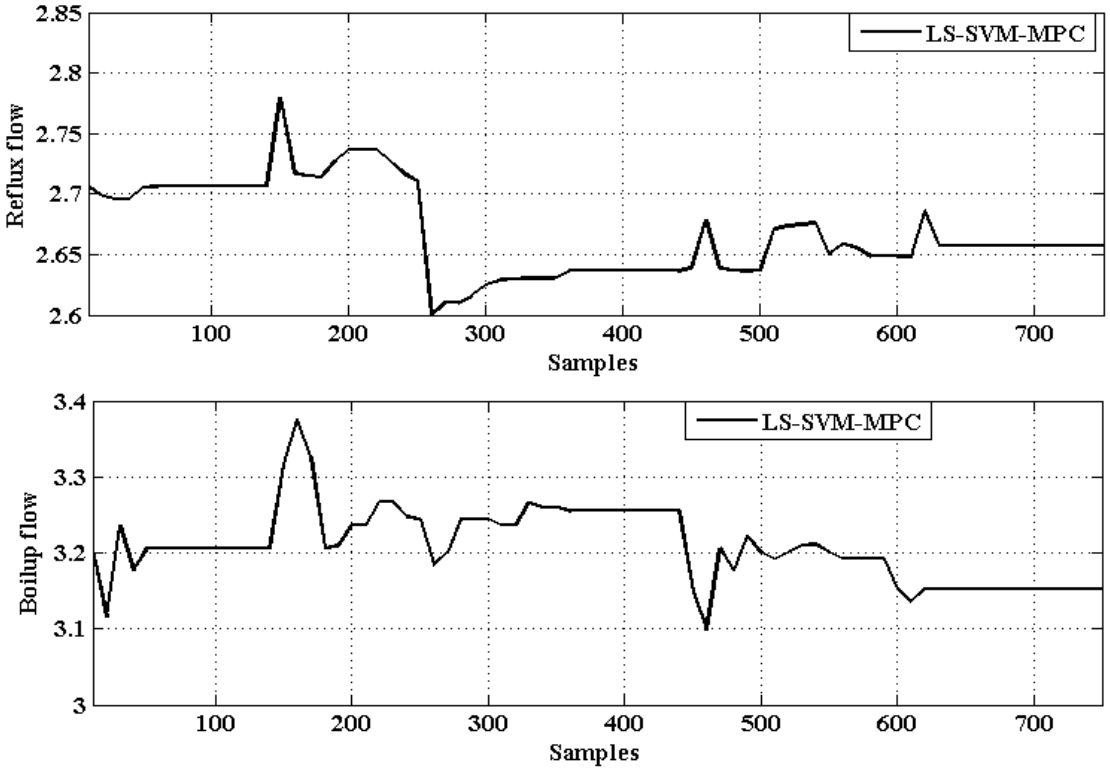

Fig. 6.12 Modifications in the process variables for tracking the top product and bottom product compositions of distillation column process

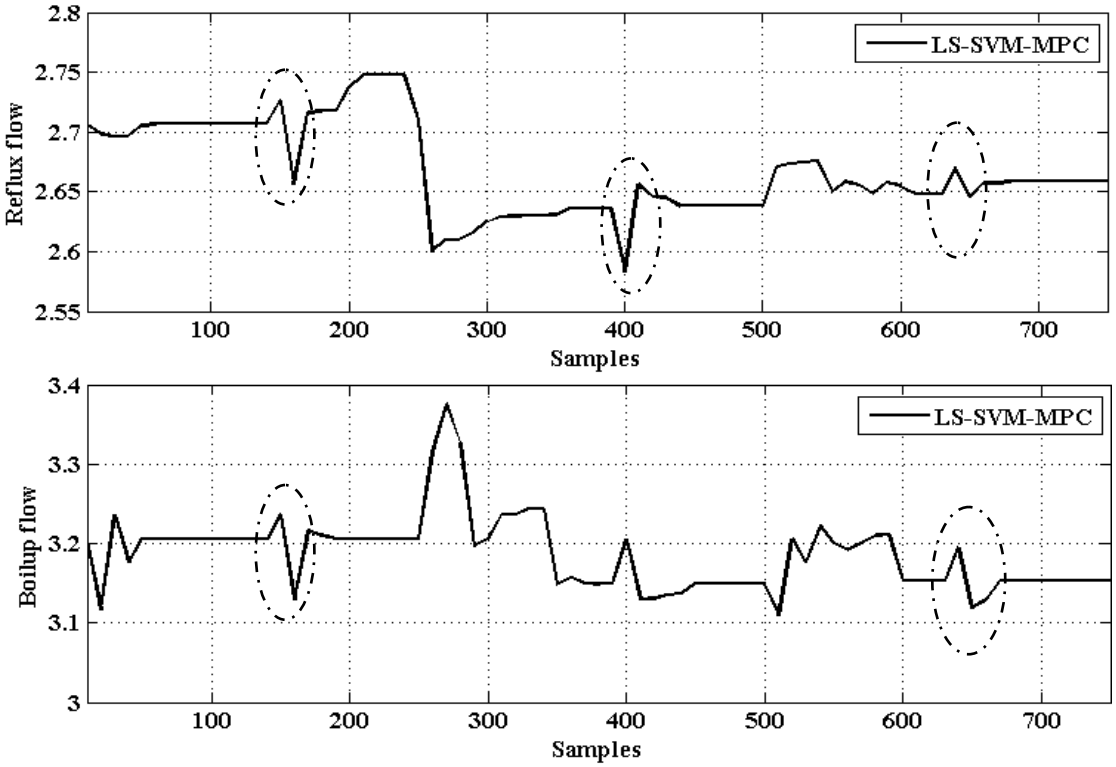

Fig. 6.13 Modifications in the process variable to illustrate unmeasured disturbance

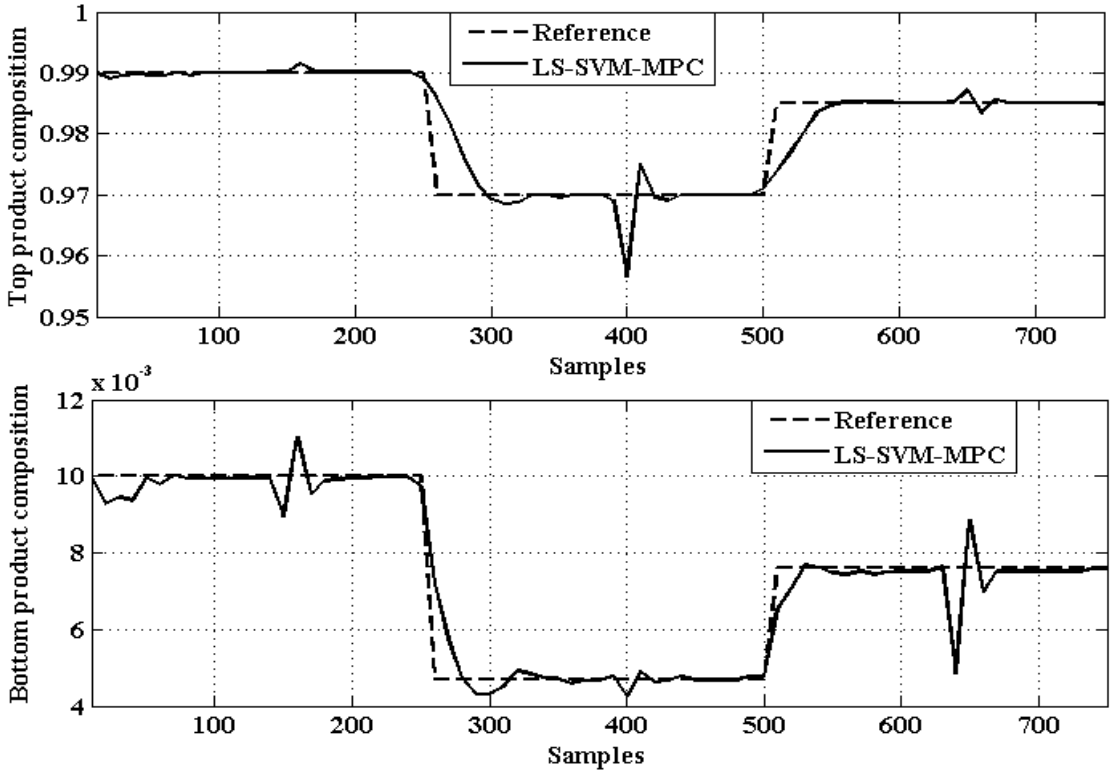

Fig. 6.14 Performance evaluation of unmeasured disturbance rejection

6.6 RVM BASED MPC OF BINARY DISTILLATION COLUMN PROCESS

This section describes the significance of RVM based MPC using PSO-CREV optimization algorithm for the multi input multi output, binary distillation column process with highly interacting process variables.

6.6.1 Training and testing the model

A random sequence of 100 samples are generated using the Matlab simulink model of binary distillation column process and is used to train the sparse Bayesian RVR model offline as discussed in chapter 3. RBF kernel is used in the RVM model which improves generalization ability and minimizes the computational complexity of the training process.

The identification performance of RVR model is assessed by RMSE performance function of equation (6.20). Fig. 6.15 and Fig. 6.16 correspond to the modeling results of RVR method. The RMSE value corresponding to the training errors of top and bottom product compositions are 0.0021 and 0.0024 respectively. The trained RVR model is further tested with 100 samples of random inputs which are beyond the training data and there corresponding absolute prediction errors are calculated. The graph of prediction errors of RVR model shown in Fig. 6.16 explores the better generalization capability of RVR model. The RMSE value corresponding to the testing errors of top and bottom product compositions are 0.0023 and 0.0025 respectively. Thus one can conclude that RVR based empirical modeling can prevail better with accurate model having good generalization capability.

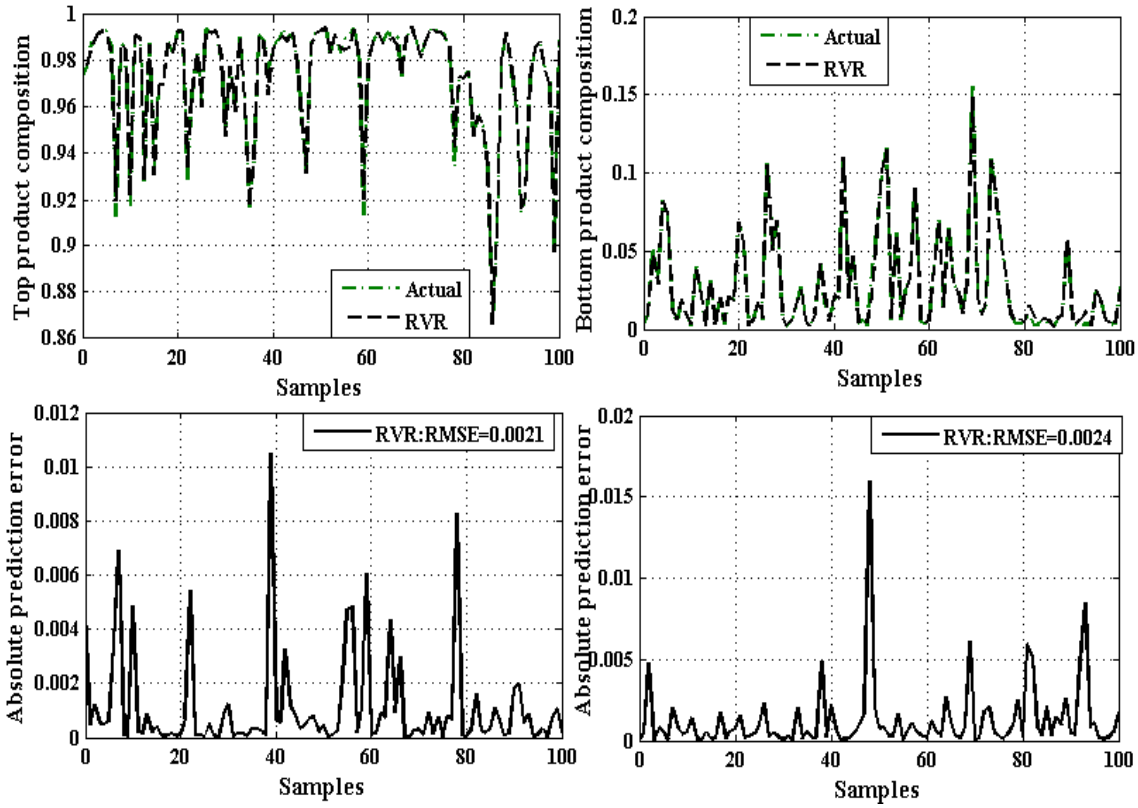

Fig. 6.15 Training performance of RVR models

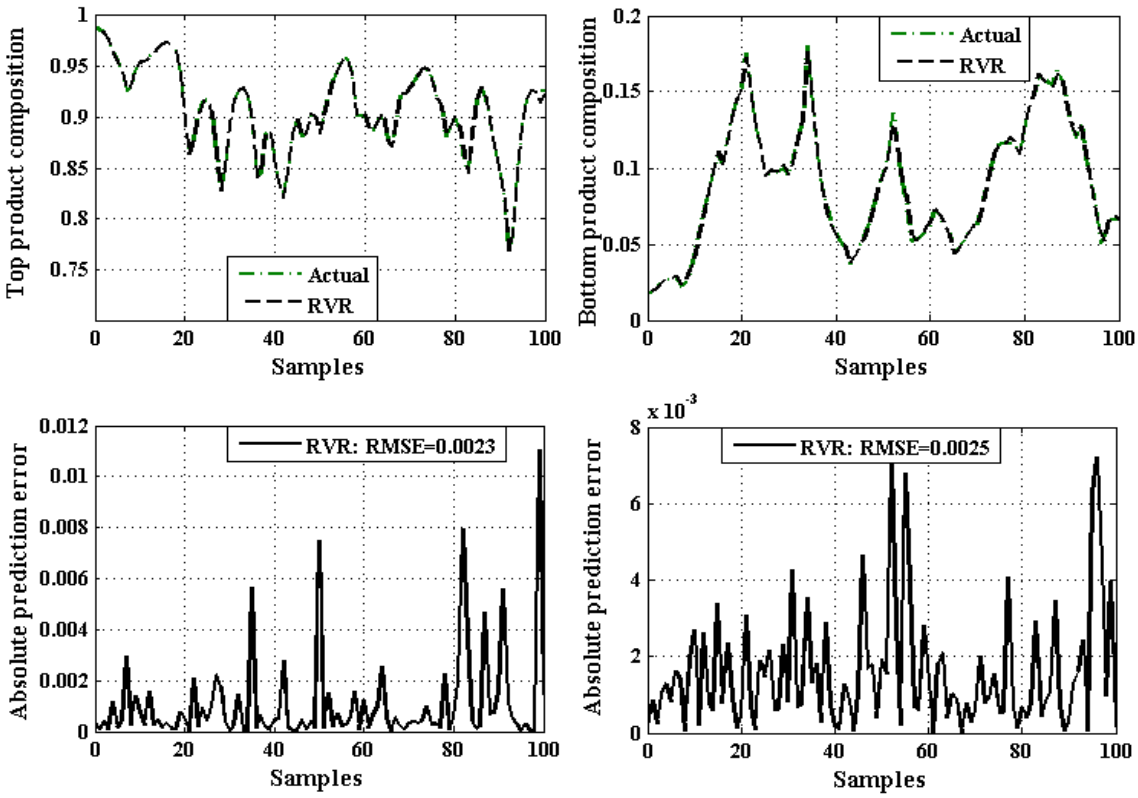

Fig. 6.16 Testing performance of RVR models

6.6.2 Performances of RVM-PSO-CREV-MPC

The offline trained and tested RVR model is then used as the nonlinear model for nonlinear MPC. Fig. 6.17 illustrates the random set point tracking performances of RVR based MPC. The RVR based MPC performs well, with no oscillations even in the presence of severe interacting process variables.

Also as the PSO-CREV algorithm congregate to the finest solution at each sampling instant the manipulated variables reflux flow rate, *L* and boilup flow rate, *V* corresponding to SVR-PSO-CREV and NN-PSO-CREV are with very less fluctuations as shown in Fig.6.18 presenting the index of control performance.

The unmeasured disturbance rejection capability of RVM-PSO-CREV based MPC is examined by subjecting the distillation column process with dissimilar magnitudes of disturbances at different sampling instants.

The control variables, Reflux flow rate, L and boilup flow rate, V with disturbances (indicated by dashed circles) at different sampling instants are shown in Fig. 6.19. Certainly the unmeasured disturbance rejection performance of RVM-PSO-CREV based MPC is very good with very less oscillations as shown in Fig. 6.20. The controlled variable, product concentration settles down faster after getting affected by unmeasured disturbances. Thus RVM-PSO-CREV based MPC behaves suitably for process control industrial applications.

Thus the better capability of RVR based MPC; in overcoming the interaction among process variables are vibrant from the simulation results. Accordingly RVR-PSO-CREV based MPC behaves suitably for process control industrial applications.

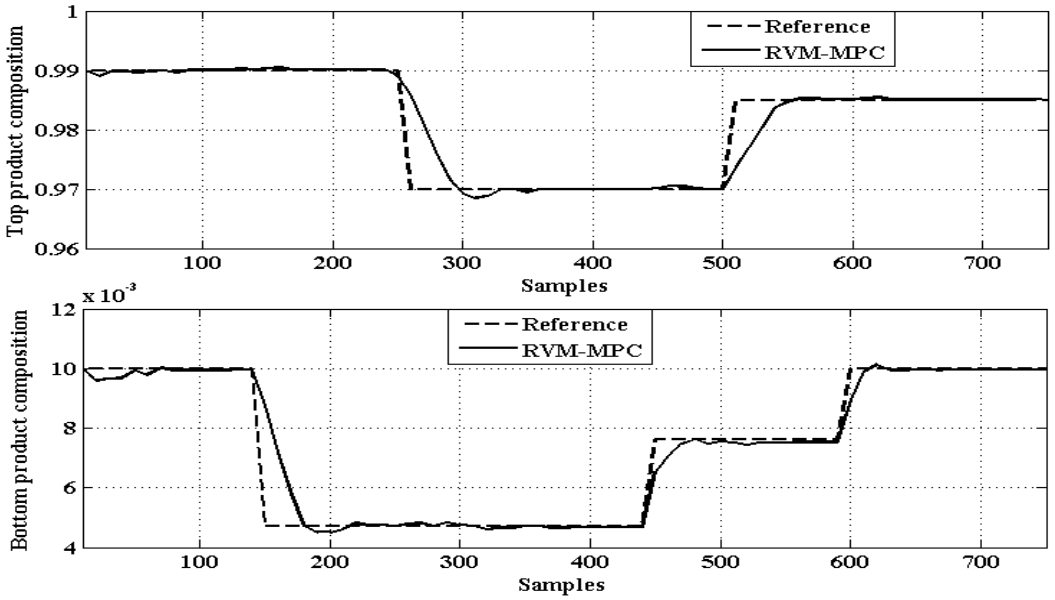

Fig. 6.17 Set point tracking performance of distillation column process by RVM-MPC

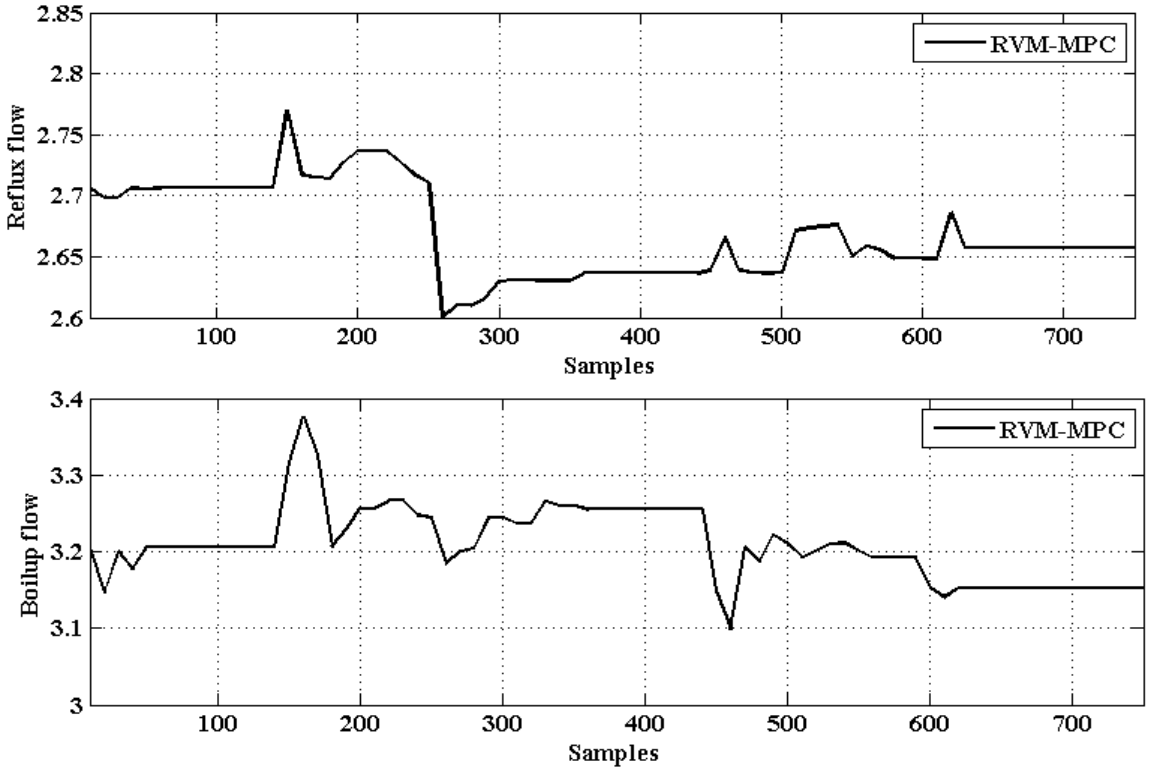

Fig. 6.18 Modifications in the process variables to trail the top product and bottom product compositions of distillation column process

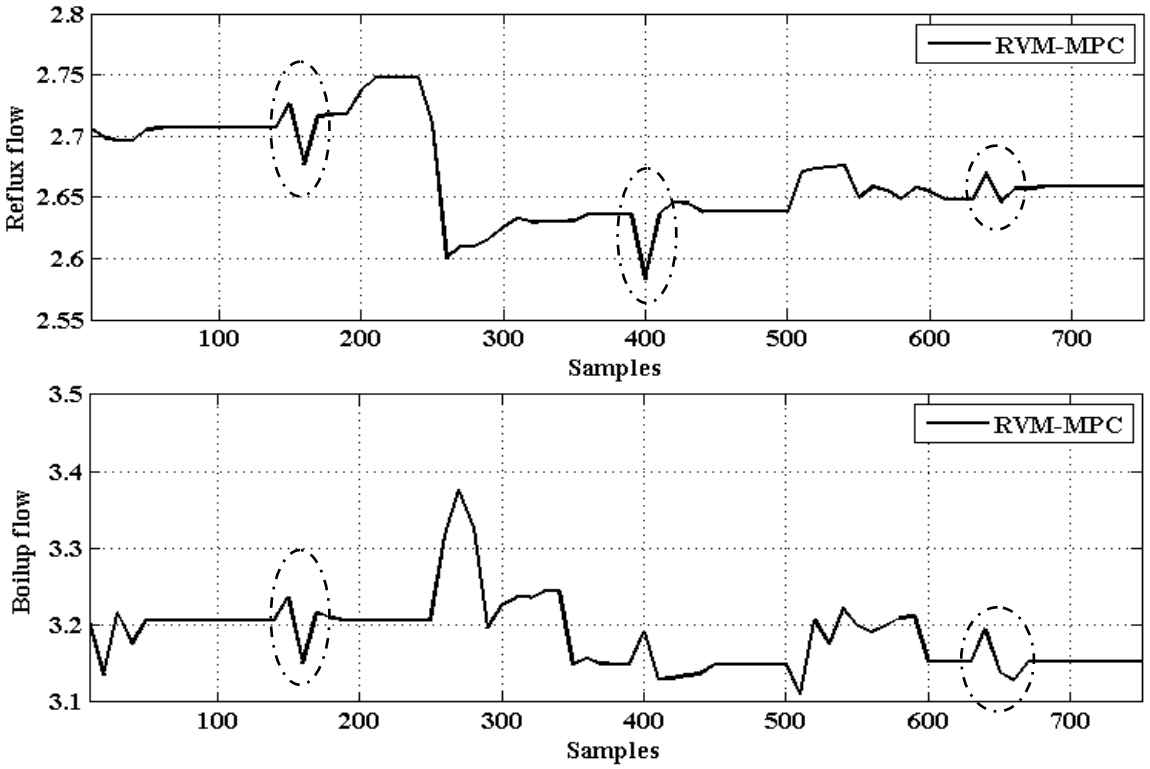

Fig. 6.19 Modification in the process variable to illustrate unmeasured disturbance

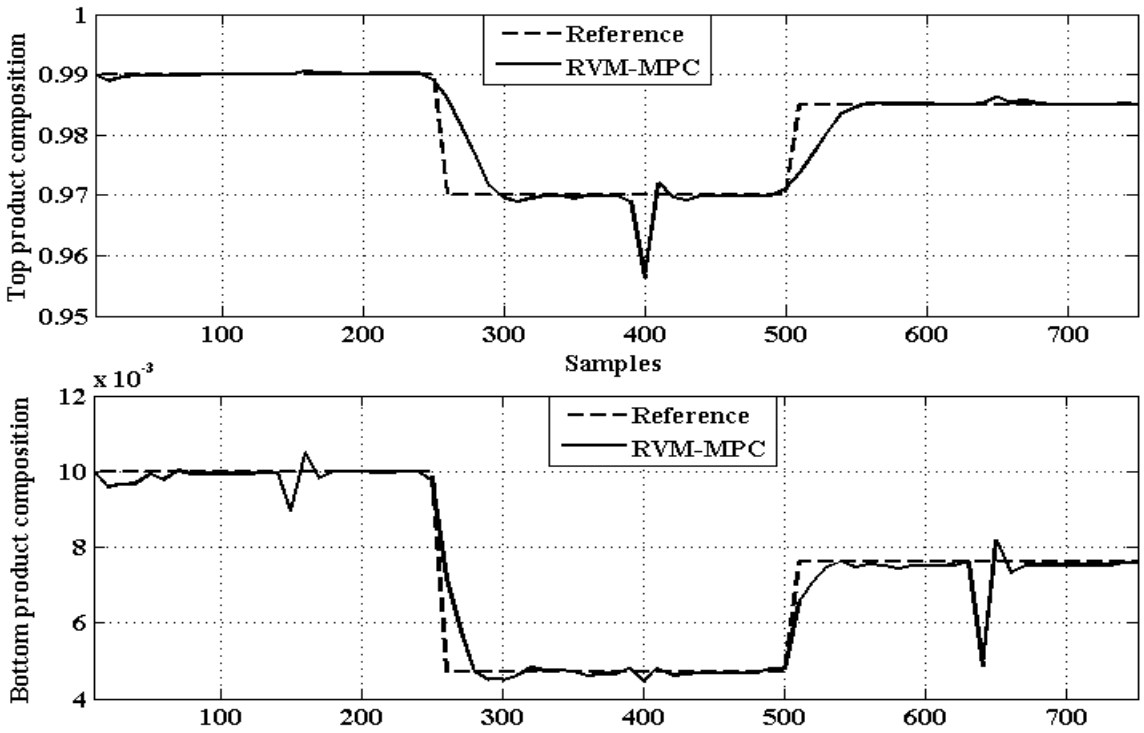

Fig.6.20 Performance evaluation of unmeasured disturbance rejection

6.7 EXTREME ANFIS BASED MPC OF BINARY DISTILLATION COLUMN PROCESS

This section describes the proposed novel Extreme ANFIS model based MPC using PSO-CREV optimization algorithm for binary distillation column process.

6.7.1 Training and Testing the Model

A sequence of 100 samples with two delay regression vector formats is used to train the Extreme ANFIS model offline. The six inputs to the network for reflux flow output are $y_1(k)$, $y_1(k-1)$, $u_1(k)$, $u_1(k-1)$, $u_2(k)$ *and* $u_2(k-1)$, Here, $y(k)$ and $y(k-1)$ corresponds to top product composition. Also $u_1(k)$ and $u_1(k-1)$ corresponds to input1, reflux flow and Also $u_2(k)$ and $u_2(k-1)$ corresponds to input 2, boilup flow. Three bell shaped membership functions with 729 rules are used for adapting parameters of ANFIS. This novel Extreme ANFIS model has good generalization ability with less training time. The final membership function after learning the nonlinear binary distillation column process with Extreme ANFIS algorithm is shown in Fig.6.21 and Fig. 6.22 respectively. Fig. 6.21 corresponds to the model for output 1, top product composition and Fig. 6.22 corresponds to the model for output 2, bottom product composition.

The identification performance of Extreme ANFIS model is assessed by the root mean square error performance function in (6.21).

$$RMSE = \left\{ \sum_{k=1}^{N} [\hat{y}(k) - y(k)]^2 \Big/ N \right\}^{1/2} \tag{6.21}$$

where $\hat{y}(k)$ signifies the predicted output of the model for the sampling instant k, $y(k)$ represents the output of the plant for the sampling instant k and N represents total number of samples.

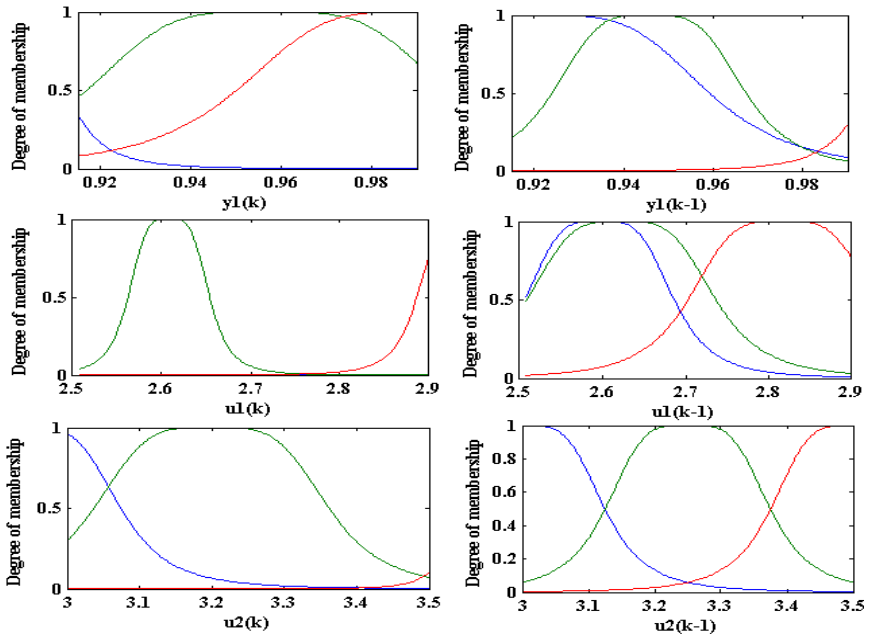

Fig. 6.21 Final membership functions of top product composition after learning with Extreme ANFIS algorithm

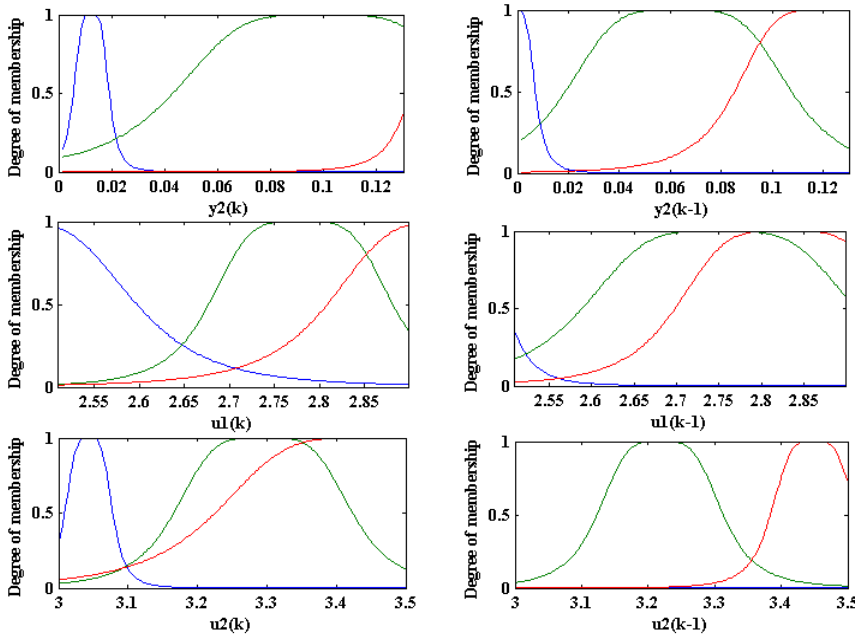

Fig. 6.22 Final membership functions of bottom product composition after learning with Extreme ANFIS algorithm

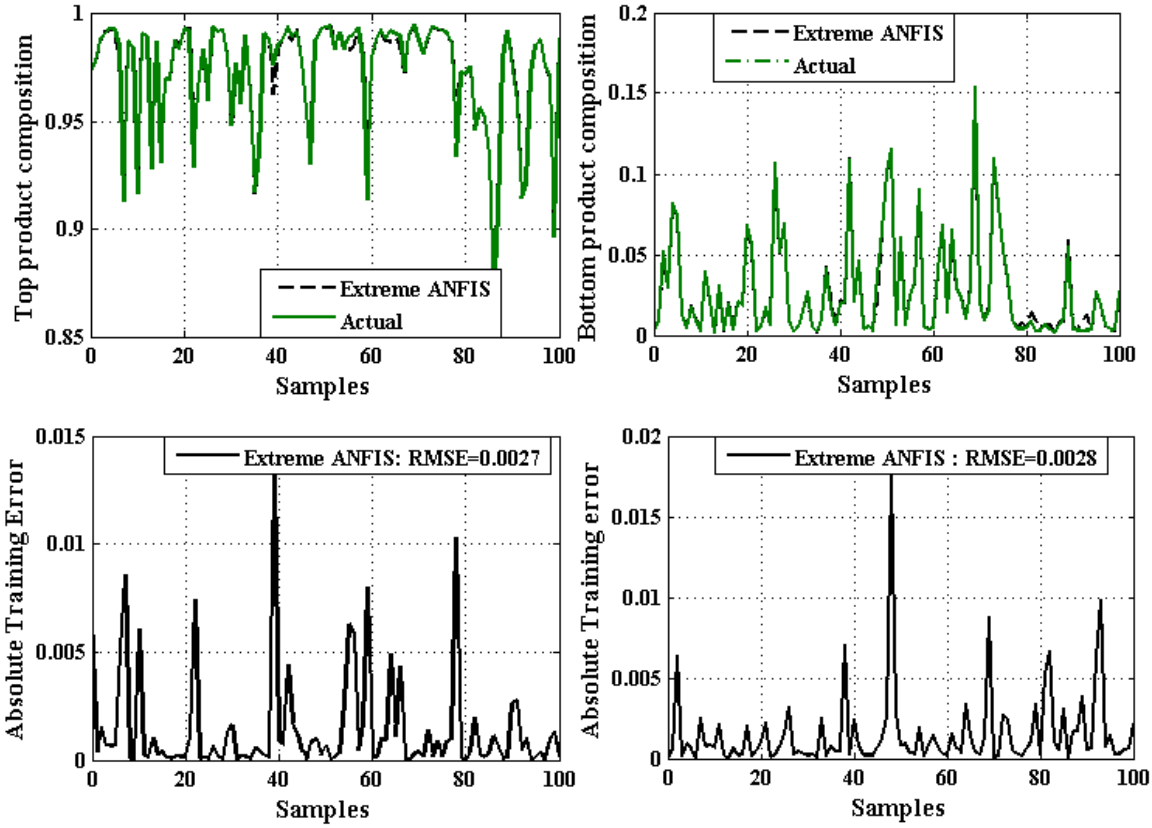

Fig. 6.23 Training performance of Extreme ANFIS model

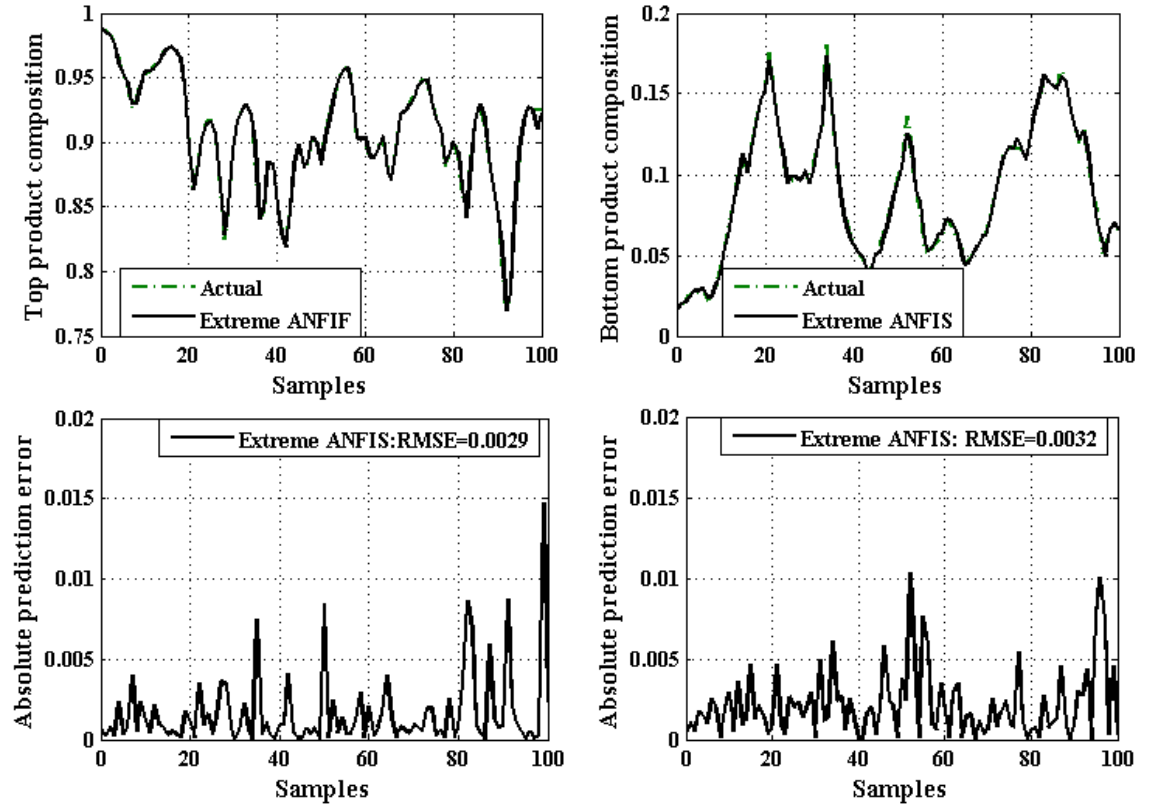

Fig. 6.24 Testing performance of Extreme ANFIS model

Figure 6.23 and Figure 6.24 corresponds to the modeling results of Extreme ANFIS model. The RMSE values corresponding to training errors of Extreme ANFIS models are 0.0027 and 0.0028 for top product and bottom product composition respectively. The trained Extreme ANFIS models are further tested with 100 samples of random inputs which are beyond the training data and there corresponding absolute prediction errors are calculated. The RMSE value corresponding to testing errors of Extreme ANFIS models are 0.0029 and 0.0032 for top product and bottom product composition respectively.

The graph for prediction error of Extreme ANFIS model is shown in Figure 6.24, which explores the better generalization capability of Extreme ANFIS model. Thus one can conclude that Extreme ANFIS based empirical model performs better with good generalization capability.

6.7.2 Performances of Extreme ANFIS-MPC

The offline trained and tested Extreme ANFIS model is then used as the nonlinear model for nonlinear MPC. Fig. 6.25 illustrates the random set point tracking performances of Extreme ANFIS based MPC. It performs well even in the presence of severe interacting process variables.

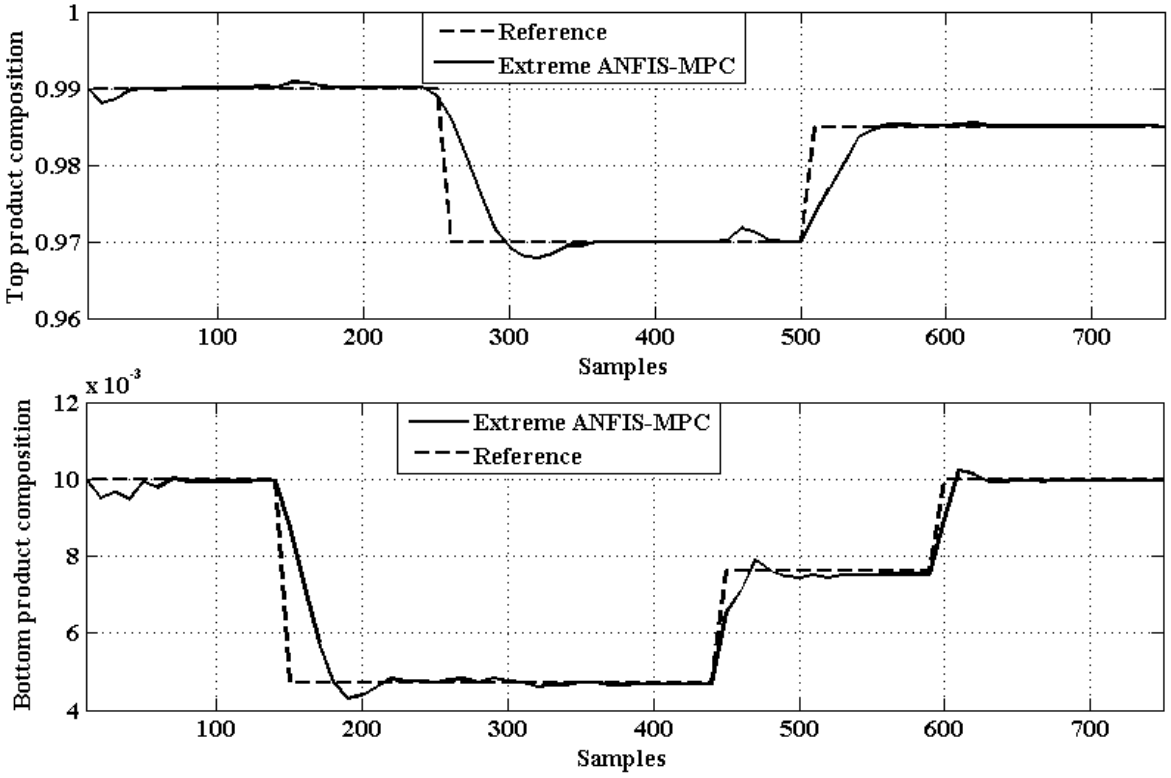

Fig. 6.25 Set point tracking performance of distillation column process by Extreme ANFIS-MPC

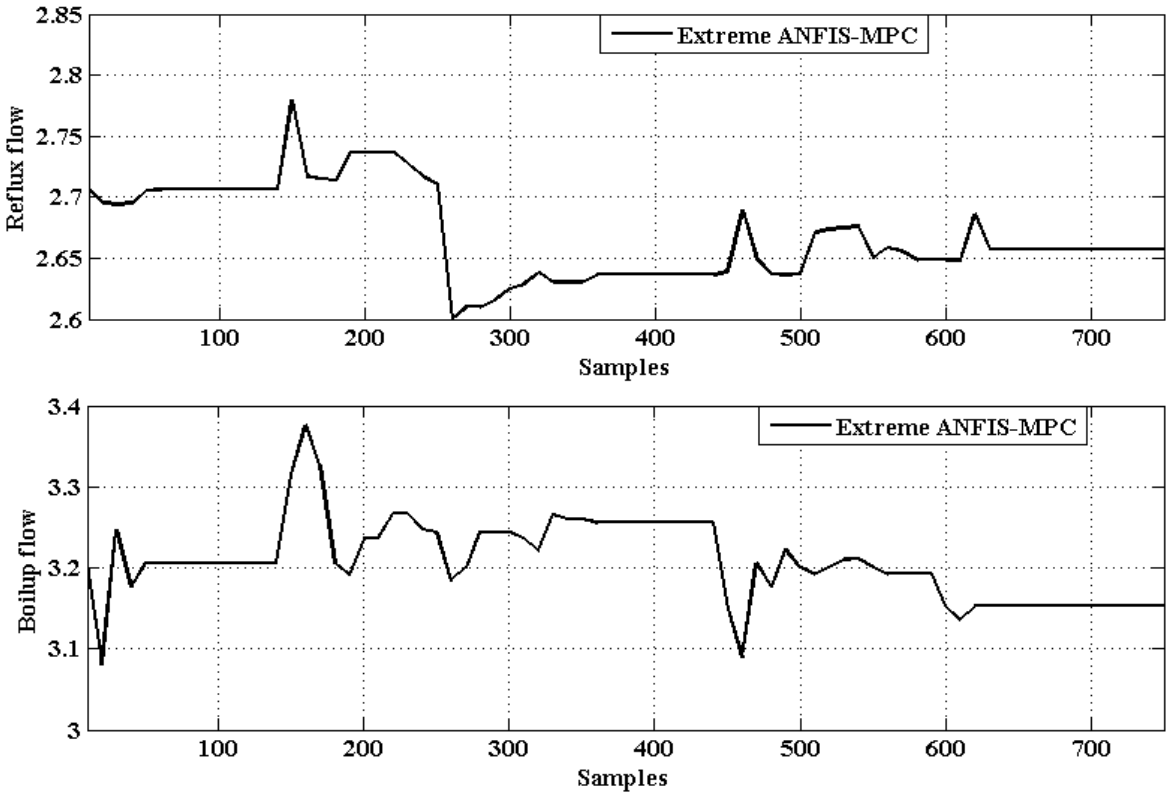

Fig. 6.26 Modifications in the process variables to trail the top product and bottom product compositions of distillation column process

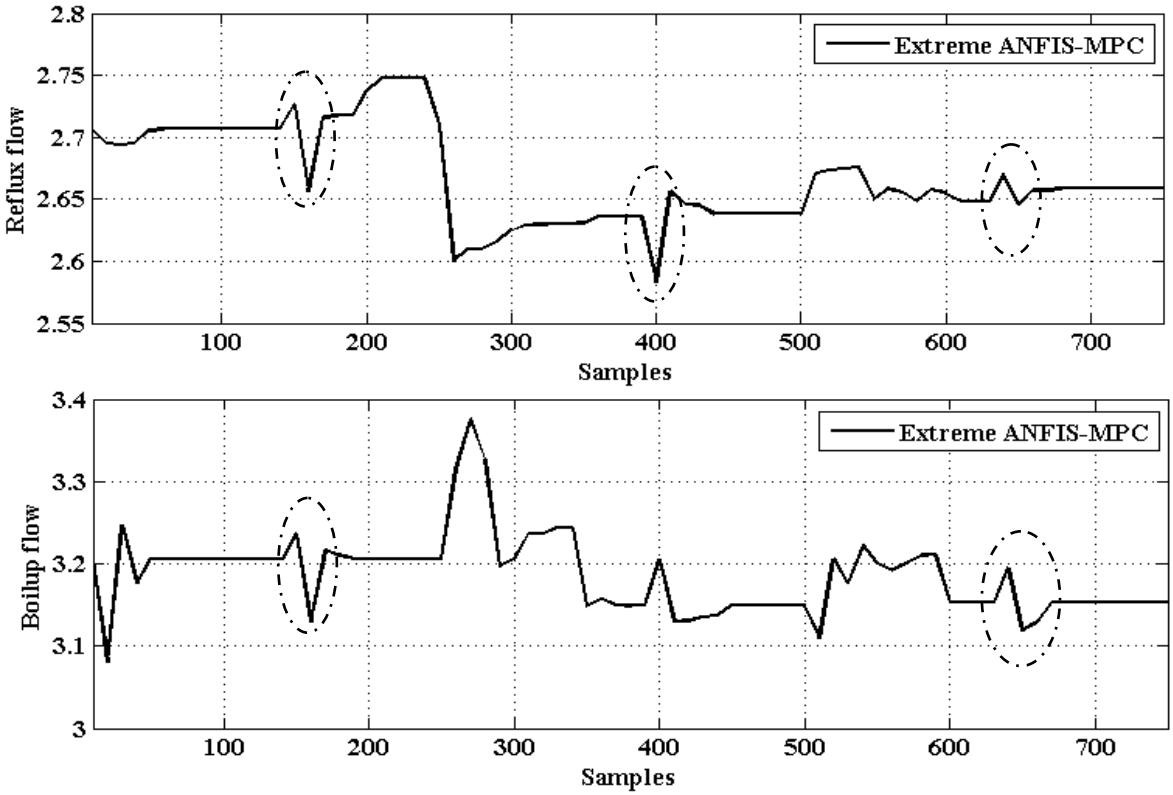

Fig. 6.27 Changes in the process variable to show unmeasured disturbance

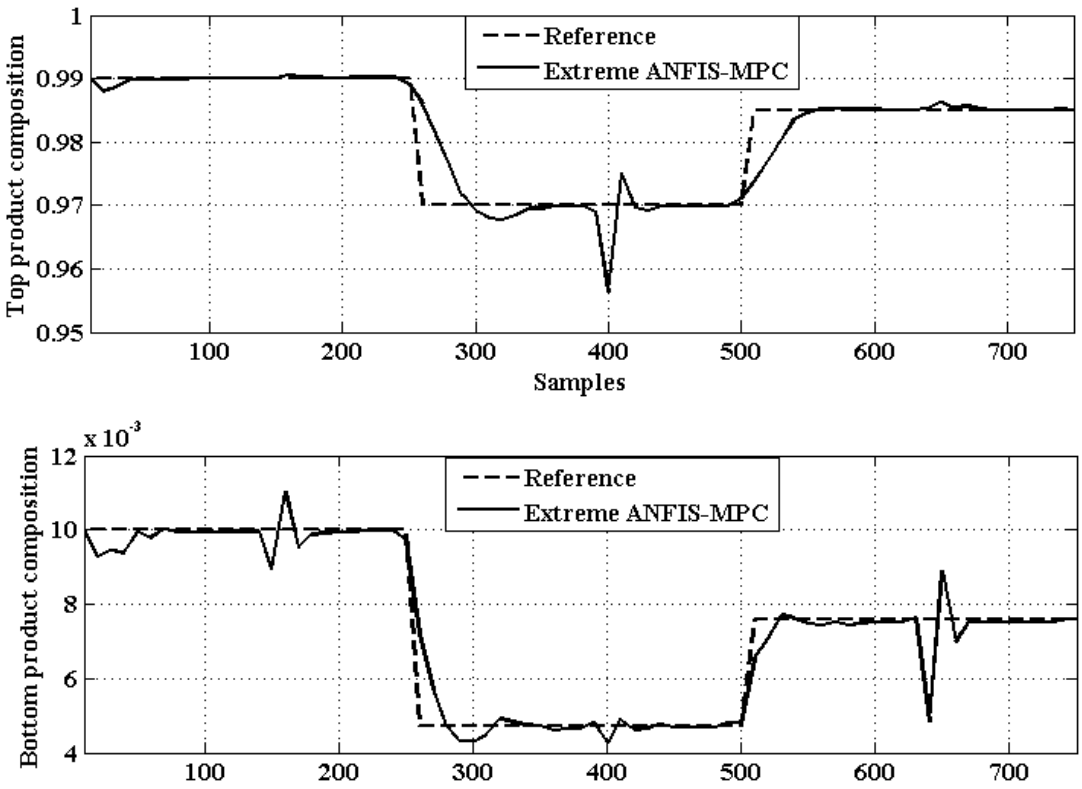

Fig. 6.28 Performance of unmeasured disturbance rejection

Also as the PSO-CREV algorithm converges to the best solution at each sampling instant the manipulated variables reflux flow rate, *L* and boilup flow rate, *V* corresponding to Extreme ANFIS-PSO-CREV-MPC are with very less fluctuations as shown in Fig.6.26 presenting the index of control performance.

The unmeasured disturbance rejection capability of Extreme ANFIS-PSO-CREV based MPC is analyzed by subjecting the distillation column process with dissimilar magnitudes of disturbance at different sampling instants. The control variables, Reflux flow rate, *L* and boilup flow rate, *V* with disturbances (indicated by dashed circles) at different sampling instant are shown in Fig. 6.27. The controlled variables, top product composition and bottom product composition settles down smoothly with very less oscillations after getting affected by unmeasured disturbances, and is shown in fig. 6.28.

6.8 TABULATION OF PERFORMANCE INDICES OF DIFFERENT CONTROLLING TECHNIQUES

This section enunciates the model accuracy of different empirical models developed and the performance indices and computational cost of all the controllers designed in the previous sections.

Table 6.1 shows the training accuracy, testing accuracy of various models with their corresponding number of training data. Even though the NN model is trained using 1000 training data, its testing error is more. But the Extreme ANFIS model, SVR model and RVR model uses less number of training data and has behaves with good generalization capability. This is clear from their corresponding prediction errors for unseen test data.

Table 6.1 Accuracy of different empirical models

Model	**Number of Training data**	**$RMSE_{training}$**		**$RMSE_{testing}$**	
		x_D	x_B	x_D	x_B
NN	1000	0.0026	0.0028	0.0153	0.0134
Extreme ANFIS	100	0.0027	0.0028	0.0029	0.0032
SVR	100	0.0027	0.0028	0.0028	0.0030
RVR	100	0.0021	0.0024	0.0023	0.0025

Table 6.2 enunciates the performance indices and computational cost of the controllers discussed in previous sections. Integral absolute error (IAE) is the performance criteria which quantifies the accuracy of all controllers. Fig. 6.29 and Fig. 6.30 shows the tracking performance of all controllers discussed in previous sections. Table 6.2 shows the IAE value and computational time related to each controller for the simulation results carried out for 750 samples.

The distillation column model under simulation has time constants of one minute. Here, NN-PSO-CREV-MPC is the one which consumes more time and RVR-PSO-CREV-MPC and Extreme ANFIS based MPC consumes very less time. Also the IAE value is more for NN-PSO-CREV and is very less for RVM-PSOCREV. Which highlights the efficient performance of RVR-PSE-CREV based MPC.

The sparseness property of SVR model reduces the computational time of SVR-MPC to 1451.1 seconds for 750 samples (ie. nearly 1.9 Seconds for sample). But the sparse nature of RVR model sharply reduces the computational time of RVR-MPC to 769.80 seconds for 750 samples (i.e., nearly 1.02 Seconds for sample), which is much shorter than the sampling time of the distillation column process.

Table 6.2 Performance Indices of different control strategies based on LS-SVM, RVM, Extreme ANFIS and NN models.

Conditions	Control tactics	Number of Training Samples	IAE		Number of Support Vectors/ Relevance Vectors		Comput-ational time (Seconds)
			Top product	Bottom product	Model1	Model2	
No Disturbance	RVM-PSO-CREV	100	0.0749	0.0157	12	11	769.80
	LS-SVM-PSO-CREV	100	0.0825	0.0165	61	59	1451.1
	Extreme ANFIS-PSO-CREV	100	0.0829	0.0167	-	-	770.9
	NN-PSO-CREV	1000	0.1337	0.1517	-	-	3750.51
Disturbance	RVM-PSO-CREV	100	0.0922	0.0158	12	11	771
	LS-SVM-PSO-CREV	100	0.1010	0.0199	61	59	1453
	Extreme ANFIS-PSO-CREV	100	0.1050	0.0200	-	-	772
	NN-PSO-CREV	1000	0.2341	0.1194	-	-	3756

The number of support vectors and relevance vectors used while modeling SVR and RVR model respectively are tabulated in Table 6.2 which signifies the sparseness of the models. The Extreme ANFIS model based MPC consumes only 770.9 seconds (i.e., nearly 1.02 Seconds for sample), due to its simple structure and simple algorithm. Thus the time consumption in Extreme ANFIS based MPC is same as RVM based MPC. But the dynamic NN model based MPC consumes 3750.51 Seconds which is much larger than all the above mentioned controllers.

Also the integral average error corresponding to NN model based MPC is very large compared to that of LS-SVM model based MPC, Extreme ANFIS model based MPC and RVM model based MPC. Hence, it is clear that NN based MPC is the one which consumes more time with more integral average error and RVR-PSO-CREV, SVR-PSO-CREV and Extreme ANFIS-PSO-CREV model predictive controllers are better than NN based model predictive controller with less computational load and less integral average error.

Also, RVR-PSO-CREV and Extreme ANFIS-PSO-CREV based MPC's consumes very less time with good tracking performances, which is much essential for real time applications. Therefore we conclude RVR-PSO-CREV-MPC and Extreme ANFIS-PSO-CREV-MPC to be the best controller based on various attributes like usage of less number of training data, better prediction accuracy, better generalization capability, excellent set point tracking performance, better unmeasured disturbance rejection capability. Hence it is well suitable for industrial process control applications.

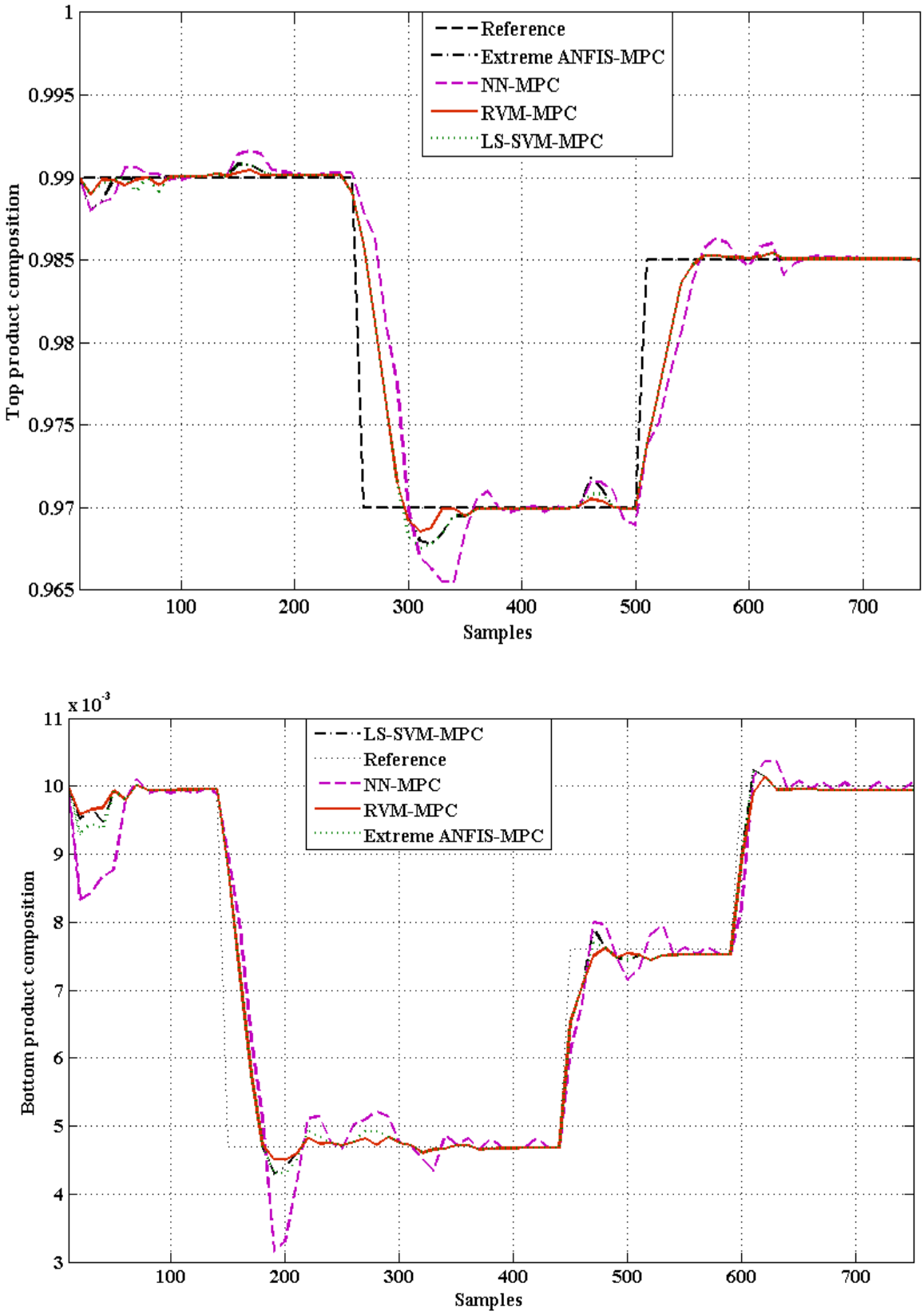

Fig.6.29 Tracking performance comparison of Top product composition and bottom product composition of binary distillation column process.

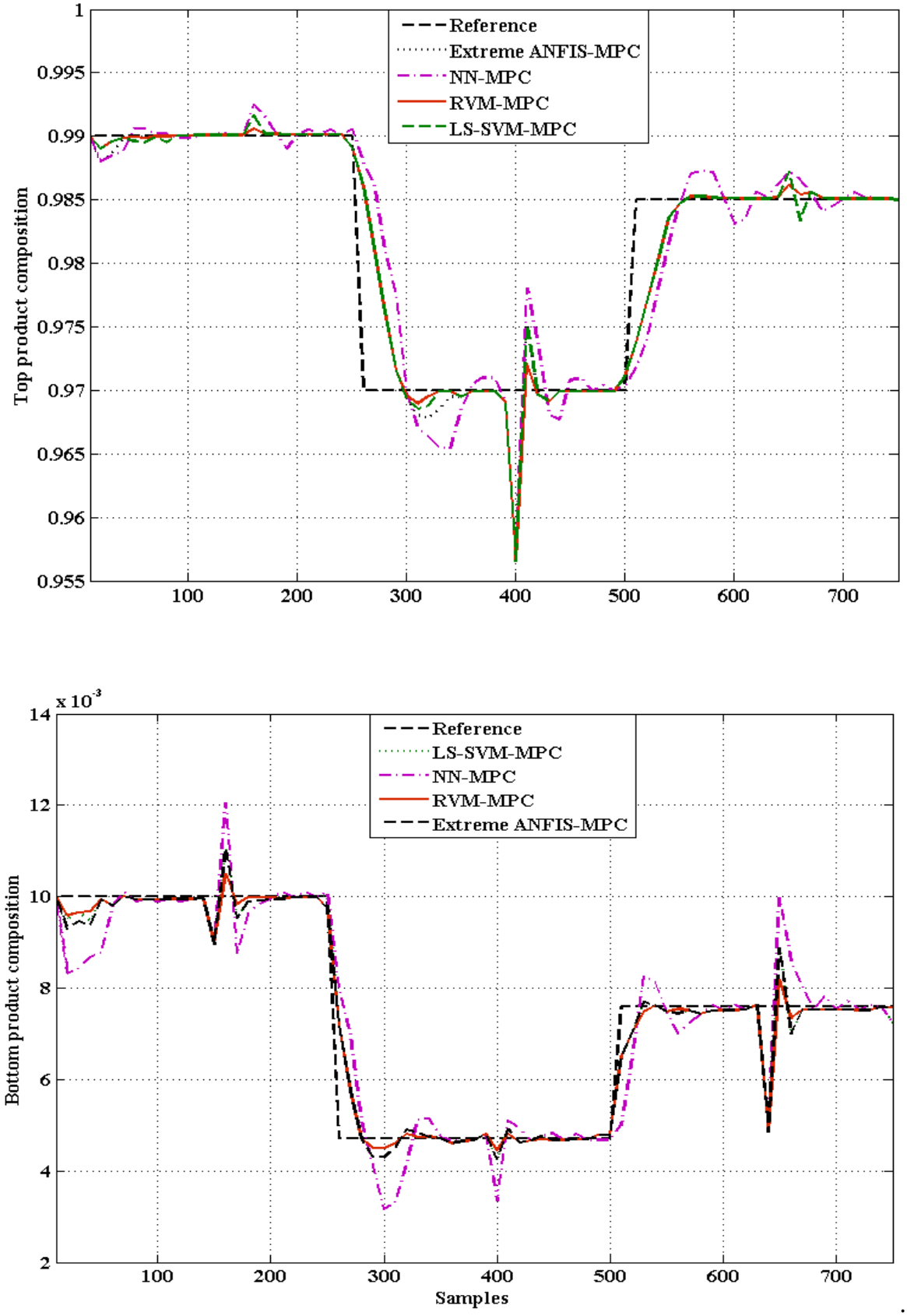

Fig. 6.30 Performance comparison of unmeasured disturbance rejection

6.9 CONCLUSION

A viable solution to the problem of nonlinear model predictive control is proposed in this chapter. Different machine learning technique is used to create an accurate for prediction model and a derivative free optimization algorithm, PSO-CREV is used to achieve faster convergence. Based on the simulation results of highly nonlinear distillation column process, the tracking performance of RVM- PSO-CREV based MPC , LS-SVM-PSO-CREV based MPC, Extreme ANFIS based MPC are much better than NN-PSO-CREV based MPC with better unmeasured disturbance rejection capability which confirms its feasibility. Simulation results convey that such better performance is due to their better prediction accuracy and better generalization capability. Also the computation time of Extreme ANFIS based MPC and RVM based MPC are very less and equal which makes them well suitable for real time control applications.

Chapter 7
CONCLUSIONS AND SCOPE FOR FUTURE WORK

7.1 CONCLUSIONS

Linear model predictive controllers were popularly practiced since 1970's. But after 1990's control theoreticians and control practitioners have shown increasing interest towards nonlinear model predictive controllers. This is because of the necessity of today nonlinear processes to operate under rigid, quality performance specifications with more and more constraints in a wide operating region. These demands could be fulfilled only if the process nonlinearities are considered explicitly while designing a controller. Nonlinear model predictive controller operates suitably to fulfill today's demands by directly using the nonlinear model for prediction, with necessary constraints and by solving the nonlinear optimal performance function online. The two demanding tasks to achieve quality performance of NMPC are an accurate nonlinear model and fast accurate convergence of performance function.

This research focused in achieving the above tasks by incorporating accurate nonlinear models and reducing the computational cost related to nonlinear model predictive controller. The conclusions of this thesis are summarized as the following:

- Learning machines as accurate models of plants give better MPC performance compared to classical models.
- Accuracy of the model influences the performance of model predictive controllers.
- Derivative free optimization technique like PSO can be used to speed up the optimization process in model predictive control.
- Model predictive controller could be applied to systems with faster dynamics by incorporating FCS-MPC principle. The control of photovoltaic array Maximum Power Point Tracker through Nonlinear Model Predictive Control strategy using fast predicting model and FCS-MPC technique is successfully done.
- The proposed Extreme ANFIS model, LS-SVM model and RVM model have very good generalization capability and hence they outperform in overcoming severe interaction, severe nonlinearity and disturbances in processes.

- But the time consumption for prediction is very less in proposed Extreme ANFIS model and sparse RVM model than LS-SVM model and NN based NMPC.
- Hence Extreme ANFIS based MPC and RVM based NMPC are the best suitable for industrial applications since it provides accurate tracking performance with very less computation time.
- The control performances of faster dynamics, photovoltaic system using Extreme ANFIS model, RVM model, and SVM model based FCS-MPC algorithm are much better than the performance of state space model based NMPC.
- Simulation results convey that such better performance is due to the better prediction accuracy of Extreme ANFIS /RVM /SVM model which is because of its good generalization capability.
- Also, RVM model based controller and the proposed Extreme ANFIS model based controller requires very less computations and hence considered to be the best suitable for implementation.

7.2 SCOPE FOR FUTURE WORK

The research work presented in this thesis can be further extended as below:

- Stability issues and robustness related to nonlinear model predictive control using learning machine could be studied.
- Since the learning time of novel Extreme ANFIS model is very less. An online trained Extreme ANFIS model based NMPC could be developed. An online trained model will have the capability to eliminate model mismatch even under unmeasured disturbance condition.
- Different optimizations techniques could be developed and applied to reduce the computation costs further in turn the NMPC could be applied for systems with faster dynamics.
- Nonlinear MPC for MPPT of PV array is simulated in which temperature is kept constant and irradiation alone is varied. As a future work both temperature and irradiation could be kept variable to design an effective control of MPPT of PV array.

PUBLICATIONS FROM THIS WORK

International Journals and International Conference - Published/Accepted

1. M. Germin Nisha, G. N. Pillai, "Nonlinear model predictive control with relevance vector regression and particle swarm optimization", Journal of control theory and applications, 2013, 11(4), 563-569. (Published by Springer available online).
2. M. Germin Nisha, G. N. Pillai, "Nonlinear model predictive control of MIMO system with relevance vector regression and particle swarm optimization", Journal of control engineering and applied informatics, (Accepted).
3. M. Germin Nisha, G. N. Pillai, "Nonlinear model predictive control using neural networks and particle swarm optimization" International conference on advanced computing methodologies, ICACM-2011, 1-6. (Published by Elsevier).
4. M. Germin Nisha, G. N. Pillai, "Nonlinear Model Predictive Control using Relevance Vector Machine for Maximum Power Point Tracking of Photovoltaic Arrays", ISA Transactions (Elsevier journal) submitted after revision.

International Journals under review

1. M. Germin Nisha, G. N. Pillai, "Nonlinear model predictive control of a MIMO system with a novel neuro-fuzzy learning machine and Particle swarm optimization", Applied soft computing (communicated).
2. M. Germin Nisha, Pushpak Jagtap, G. N. Pillai, "Nonlinear model predictive control with a new neuro-fuzzy learning machine and Particle swarm optimization", Soft computing (communicated).

BIBLIOGRAPHY

1. R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems". *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

2. M. Athans, "The role and use of the stochastic Linear-Quadratic-Gaussian problem in control system design". *IEEE Transaction on Automatic Control*. AC-16, vol. 6, pp. 529–552, 1971.

3. J. Richalet, A. Rault, J.L.Testud and J. Papon, "Model Predictive Heuristic Control: Applications to Industrial Processes", *Automatica*, vol. 14, pp. 413-428, 1978.

4. C. R. Cuttler and B.L. Ramaker, "Dynamic matrix control- a computer control algorithm" *in Proc. Joint automatic control conference*, San Francisco, 1980.

5. S. J. Qin and A. T. Badgwell, "A survey of industrial model predictive control technology", *Control Engineering Practice*, vol. 11, no. 7, pp. 733-764, 2003.

6. M. A. Henson, "Nonlinear Model predictive control: current status and future directions", *Computers and chemical engineering*, vol. 23, pp. 187-202, 1998.

7. A. Rossiter, "Model Based predictive control: a practical approach", CRC press Baco Paton London, New York, Washington, D.C.2003.

8. N. Bhat and T. J. Mcavoy, "Use of neural nets for dynamic modeling and control of chemical process systems", *Comput. Chem. Eng*., vol. 14, no. 5, pp. 573-583, 1990.

9. D. C. Psichogios and L. H. Ungar, "Direct and indirect model based control using artificial neural network", *Industrial & Engineering Chemistry Research*, vol. 30, no. 12, pp. 2564-2573, 1991.

10. K. J. Hunt, K. Sbarbaro, R. Zbikowski , and P. J. Gawthrop, "Neural Network for control systems – a survey", *Automatica*, vol. 28, no. 6, pp. 1083-1120, 1992.

11. Y. Liu, Y. Gao, Z. Gao, H. Wang, and P. Li, "Simple nonlinear predictive control strategy for chemical processes using sparse kernel learning with polynomial form", *Industrial & Engineering Chemistry Research*, vol. 49, pp. 8209–8218, 2010.

12. J. S. Taylor, and N. Cristianini, "Kernel Methods for Pattern Analysis", Cambridge University Press, Cambridge, UK, 2004.

13. C. M. Bishop, “Pattern Recognition and Machine Learning”, Springer-Verlag, New York, 2006.

14. V. Vapnik, “Statistical Learning Theory”, Wiley, New York, 1998.

15. H. Zhang and X. Wang, “Nonlinear systems modeling and control using support vector machine technique”, *Lecture Notes in Computer Science*, Springer-Verlag: New York, pp. 660 – 669, 2006.

16. A. Kulkarni, V. K. Jayaraman and B. D. Kulkarni , “Control of chaotic dynamical systems using support vector machines”, *Phys. Lett. A*, vol. 317, pp. 429–435, 2003.

17. W. Zhong, D. Pi and Y. Sun, “An approach of nonlinear model multistep ahead predictive control based on SVM”, *Lecture notes in Computer Science*, Springer-Verlag Berlin Heidelberg, vol. 3516, pp. 1036-1039, 2005.

18. W. Zhong, D. Pi, Y. Sun, “Support vector machine based nonlinear model multistep ahead optimizing predictive control”, *Journal of Cent. South university of technology*,vol. 12, no. 5, pp. 591-595, 2005.

19. C. Yue-hua, C. Guang-yi, Z. Xin-jian , “ LS-SVM model based nonlinear predictive control for MCFC system”, *Journal of Zhejiang University Science A*, vol. 8, no.5, pp. 748-754, 2007.

20. X. C. Xi, A. N. Poo, S. K. Chou, “Support vector regression model predictive control of a HVAC plant”, *Control Engineering Practice*,vol. 15, pp. 897-908, 2007.

21. M. E. Tipping “The relevance vector machine”, *Advances in Neural Information Processing Systems*, vol. 12, pp. 652–658, 2000.

22. M. E. Tipping, “Sparse Bayesian Learning and the Relevance Vector Machine”, *Journal of Machine Learning Research*, vol. 1, 2001, pp. 211–244.

23. G.-B. Huang, Q.-Y. Zhu and C.-K. Siew, “Extreme learning machine: theory and applications”, *Neurocomputing*, vol. 70, pp. 489–501, 2006.

24. G.-B. Huang, H. Zhou, X. Ding, R. Zhang, "Extreme learning machine for regression and multiclass classification0", *IEEE Transactions on Systems. Man Cybernetics B Cyber*, vol. 42, no. 2, pp. 513–529, 2012.

25. J. Rodriguez, M. P. Kazmierkowski, J. R. Espinoza , P. Zanchetta, H. Abu-Rub, H. A. Young, and C. A. Roja, "State of the Art of Finite Control Set Model Predictive Control in Power Electronics" *IEEE Transactions on Industrial Informatics*, vol. 9 no. 2, pp.1003-101, 2013.

26. S. Purwar, I.N. Kar, A.N. Jha, "On-line system identification of complex systems using Chebyshev neural networks", *Applied Soft Computing*, vol. 7, pp. 364–372, 2007.

27. A. A. Patwardhan and T. F. Edgar, "Nonlinear model predictive control of apacked distillation column", *Industrial Engineering and Chemistry Research*, vol. 32, no. 10, pp. 2345- 2356, 1990.

28. H. Chen and F. Allgower, "A quasi infinite horizon nonlinear model predictive control scheme with guaranteed stability", Automatica, vol. 34, no. 10, pp. 1205-1218, 1997.

29. N. L. Ricker and J. H. Lee, "Nonlinear Model predictive control of the Tennesse Eastman challenge process", Computer and Chemical Engineering, vol. 19, pp. 961-981, 1995.

30. A. Zheng, "A Computational efficient nonlinear linear model predictive control algorithm", *in Proc. American Control Conference*. Albuquerque, NM, 1997.

31. A. A. Padtwardhan, T. T. Wright and T. E. Edgar, "Nonlinear model predictive Control of distributed parameter systems", *Chemical Engineering Science*. vol. 47, no. 4, pp. 721-735, 1992.

32. J. H. Lee, "Modeling and identification for nonlinear model predictive control: requirement,current status and future research needs, Editors: Allogower, F. and Zheng, A. *Nonlinear model predictive control*, Birkhauser. pp. 269-293, 2000.

33. K. P. Fruzzetti, A. Palazoglu and K. A. MacDonald , "Nonlinear model predictive control using Hammerstein models", *Journal of Process Control*, vol.7, no. 1, pp. 31-41, 1997.

34. B. R. Maner, F. J. Doyle, B. A. Ogunnaike and R. K. Pearson, “Nonlinear model predictive control of a simulated multivariable polymerization reactor using second order Volterra models, *Automatica,* vol. 32, pp. 1285-1301,1996.

35. S. S. Jang and L. S. Wang, “Experimental study of rigorous nonlinear model predictive control for a packed distillation column”, *Journal of Chinese Institute of Chemical Engineer*, vol. 28, no. 3, pp. 151-162, 1997.

36. M. Asohi, “Modeling and Control of a Continuous Crystallization Process Using Neural network and Model Predictive Control” Ph. D dissertation, University of Saskatchewan, 1995.

37. F. Allgower, A. Zeng, “progress in systems and control theory: Nonlinear model predictive control”, vol. 26, Berelin, Birkhauser, 2000.

38. F. Alonge, F. D Ippolito, F.M. Raimondi and S, Tumminaro, “Identification of nonlinear systems described by Hammerstein models” in *Proc. of the 42 IEEE conference on decision and control*, Hawali ,USE December 2003.

39. Y. Chen, B. Yang, J. Dong, A. Abraham, “Time-series forecasting usingflexible neural tree model”, *Information Sciences*, vol. 174,pp. 219-235, 2005.

40. K. J. Hunt, D. Sbarbaro, R. Zbikowski and P. J,Gawthrop, “Neural Network for control systems”, *Automatica,* vol. 28, no. 6, pp. 1083-1112, 1992.

41. K. S. Narendra and K. Parthasarathy, “Identification and control of Dynamic Systems using Neural Networks”, *IEEE Transactios on Neural networks.* vol. 1, pp. 4-27, 1990.

42. Arumugasamy, S. Kumar and A. Zainal, “Elevating Model Predictive Control Using Feedforward Artificial Neural Networks: A Review”, Chemical Product and Process Modeling: vol. 4, no. 1, Article 45, 2009.

43. J. V. Desai, B. Bandyopadhyay and C. D. Kane, “Neural network based fabric classification and blend composition analysis”, *in Proc. IEEE international conference on industrial technology*, 2000.

44. P. Georgieva and F. D. A. Segbastio, “Application of feed forward neural networks in modeling and closed loop control of a fed-batch crystallization process",

Transactions on Engineering, Computing and Technology, vol. 12, pp. 65-70, 2006.

45. M. A. Hussain, "Review of the applications of neural networks in chemical process control - simulation and online implementation", *Artificial Intelligence in Engineering*, vol.13 no. 1, pp. 55- 68, 1999.

46. E. Al-Gallaf, "Artificial Neural Network Based Nonlinear Model Predictive Control Strategy", *Information Technology Journal*, vol. 1 no. 2, 173-179, 2002.

47. V. Rankovic and I. Nikolic, "Identification of Nonlinear Models with Feedforward Neural Network and Digital Recurrent Network", *FME Transactions*, vol. 36, no. 2, pp. 87-92, 2008.

48. S. Chidrawar and B. Patre, "Generalized Predictive Control and Neural Generalized Predictive Control", *Leonardo Journal of Sciences*, vol. 7, no.13, 133-152, 2008.

49. P. Kittisupakorn, P. Thitiyasook and Hussain, "Neural network based model predictive control for a steel pickling process", *Journal of Process Control*, vol. 19, no. 4, pp. 579-590, 2009.

50. J. Z. Chu, "Multistep model predictive control based on Artificial Neural Networks", *Industrial Engineering Chemistry.Research*, vol. 42, pp. 5215-5228, 2003.

51. A. S. Kamalabady, K. Salahshoor, "New SISO and MISO Adaptive Nonlinear Predictive Controllers based on Self Organizing RBF Neural Networks", in proc. 3rd IEEE International conference on communication control and signal processing, 2008.

52. W. Guo, M. Han, "Generalized predictive controller based on RBF neural network for a class of nonlinear system", *in Proc. IEEE American control conference*, 2006.

53. R. Ahmad and H. Jamaluddin, "Radial basis function for nonlinear dynamic system identification", *Jurnal Teknologi*, vol. 36, no. A, pp. 39–54, 2002.

54. http://www.statsoft.com/textbook/neural-networks/#radial (accessed on 10/9/2013)

55. L. M. Saini and M. K. Soni, “Artificial Neural Network-Based Peak Load Forecasting Using Conjugate Gradient Methods”, *IEEE Transactions on power systems*, vol. 17, no. 3, pp. 907-912, 2002.

56. P. R. Patnaik, “Neural control of an imperfectly mixed fed-batch bioreactor for recombinant ß –galactosidase”, *Biochemical Engineering Journal*, vol. 3, no. 2, pp. 113-120, 1999.

57. F. Declercq and R. D. Keyser, “Comparative study of neural predictors in model based predictive control”, *in Proc. of International Workshop on Neural Networks for Identification, Control, Robotics, and Signal/Image Processing*, Bratislava, pp.20-28, 1996.

58. C. Cortes and V. Vapnik, "Support-Vector Networks", Machine Learning, vol.20, 1995.

59. A. Karatzoglou and D. Meyer, “Support Vector machine in R”, *Journal of statistical software*, vol. 15, no. 9, pp. 1-32, 2006.

60. U. Thissen, R. V. Brakel, A.P. D. Weijer, W.J. Melssen and L.M.C. Buydens,“ Using support vector machines for time series prediction” *Chemometrics and Intelligent Laboratory Systems* ,vol.69, pp.35-49, 2003.

61. W. Wang, C. Men, W. Lu, “Online prediction model based on support vector machine”, *Neurocomputing*, vol. 71, pp. 550-558, 2008.

62. S. R. Kolla, “Identifying faults in three phase induction motors using support vector machines”, *in proc. Electrical manufacturing and coil winding expo,* pp: 109-114, 2010-2013.

63. D. G. Tzikas , L. Wei , A. Likas , Y. Yang , and N. P. Galatsanos, “A Tutorial on Relevance Vector Machines for Regression and Classification wit Applications”, University of Ioannina, Ioanni, GREECE, Illinois Institute of Technology, Chicago, USA, 2006.

64. G. Camps-Valls, M. Martinez-Ramon, J. L. Rojo-Alvarez and J. Munoz-Mari. “Nonlinear System Idenfication With Composite Relevance Vector Machines”, *IEEE Signal Processing Letters*, vol.14, no. 4, pp. 279-282, 2007.

65. I. Psorakis, T. Damoulas and A. M. Girolami, “Multiclass Relevance Vector Machines: Sparsity and Accuracy”, *IEEE Transactions on neural networks*, vol. 21, no. 10, pp. 1588-1598, 2010.

66. J. Q. Candela and L. K. Hansen,“Time serier prediction based on the relevance vector machine with adaptive kernels”, in *Proc. IEEE International Conference on Aquostics, Speech and Signal Processing*, pp. 985-988, 2003.

67. A. M. Nicolaou, H. Gunes, and M. Pantic “Output-associative RVM regression for dimensional and continuous emotion prediction”, *Image and Vision Computing*, vol. 30, no. 3, pp.186-196, 2012.

68. S. TAI. “An Annealing Dynamical Learning-based Relevance Vector Regression Algorithm for Housing Price Forecasting”, *Journal of Information & Computational Science*, vol. 8, no. 14, pp.3313–3319, 2011.

69. P. Wong, Q. Xu, C. Vong and H. Wong, “Rate-Dependent Hysteresis Modeling and Control of a Piezostage Using Online Support Vector Machine and Relevance Vector Machine”, *IEEE Transactions on industrial Electronics*, vol. 59, no. 4, pp. 1988-2001, 2012.

70. A. Abraham, B. Nath, “A neuro-fuzzy approach for modeling electricity demand in Victoria”, Applied Soft Computing, vol. 1, pp. 127–138, 2001.

71. S. Yadav, J. P. Tiwari and S. K. Nagar, “Digital Control of Magnetic Levitation System using Fuzzy Logic Controller”, *International Journal of Computer Applications*, vol. 41, no. 21, pp. 27-31. 2012.

72. S. Harish and M. K. Mishra, “Fuzzy Logic based Supervision of DC link PI Control in a DSTATCOM”, *in Proc. IEEE India conference INDICON*, vol. 2, 2008

73. J. S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference systems," *IEEE Trans. Svst., Man, Cybern.*, vol. 23, no. 3, pp. 665-685, 1993.

74. M. Sugeno and G. T. Kang, “Structure identification of fuzzy model”, IEEE transactions on fuzzy sets and systems, vol.28, No. 4, pp.15-33, 1988.

75. 7J. S. Jang, C. T. Sun and E. Mizutani, “Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence”, NJ: Prentice-Hall, 1997.

76. C. Grosan and A. Abraham, “Multiple Solutions for a System of Nonlinear Equations”, *International Journal of Innovative Computing, Information and Control*, vol. 4, no. 9, pp. 2161–2170. 2008.

77. D. Soloway and P. J. Haley, “Neural generalized predictive control: A Newton-Raphson Implementation”, *in Proc. IEEE International Symposium on Intelligent Control Dearborn*, MI September 15-18, pp. 277-282, 1996.

78. S. K. Chidrawar, S. Bhaskarwar and B. M. Patre, “Implementation of neural network for generalized predictive control A Comparison between a Newton Raphson and Levenberg Marquardt implementation”, *World congress on Computer science and information engineering*, vol. 1, pp. 669-673, 2009.

79. M. Norgaard, “Neural Networks for Modelling and Control of Dynamic Systems: A Practitioner's Handbook” Springer verlag, London, 2000.

80. C. Grosan and A. Abraham, “A New Approach for Solving Nonlinear Equations Systems”, IEEE Transactions on systems man and cybernetics, vol. 38, no. 3, pp. 698-714, 2008.

81. X. C. Xi, A. N. Poo and S. K. Chou, “ Support vector regression model predictive control of a HVAC plant”, *Control Engineering Practice*,vol.15, pp. 897-908, 2007.

82. D. K. Kumar, S. K. Nagar and S.K. Bharadwaj, “Model order reduction based on SISO and MIMO systems based on genetic algorithm”, *in Proc. International conference on automation robotics and control systems*, pp.97-104, 2010.

83. C. Yue-hua, C. Guang-yi and Z. Xin-jian, “LS-SVM model based nonlinear predictive control for MCFC system”, *Journal of Zhejiang University Science A*, vol.8, no.5, pp.748-754, 2007.

84. E. Elbeltagi, T. Hegazy and D. Grierson, “Comparison among five evolutionary - based optimization algorithms”, *Advanced engineering informatics*, vol. 19, pp. 43-53, 2005.

85. C. Grosan, A. Abraham and M. Nicoara, "Performance Tuning of Evolutionary Algorithms Using Particle Sub Swarms", *in Proc. 7th IEEE International Symposium on Symbolic and Numeric Algorithms for Computing Scientific*, 2005.

86. R. C. Eberhart and J. Kennedy, "A New Optimizer Using Particle Swarm Theory", in *Proc. 6th International Symposium on Micro Machine and Human Science*, Nagoya, Japan, pp. 39-43, 1995.

87. J. Kennedy and R. C. Eberhart "Particle Swarm Optimization", in *Proc. IEEE International Conference on Neural Network*, Perth, Australia, pp. 1942-1948,1995.

88. H. Yoshida, K. Kawata, Y. Fukuyama and Y. A. Nakanishi, " Particle Swarm Optimization for Reactive Power and Voltage Control Considering Voltage Stability", in *Proc. International Conference on Intelligent System Application to Power Systems*, Rio de Janeiro, Brazil, pp. 117-121,1999.

89. L. Messerschmidt and A. P. Engelbrecht, "Learning to Play Games Using a PSO-Based Competitive Learning Approach", *IEEE Transactions on Evolutionary Computation*, vol 8, no. 3, pp. 280-288, 2004.

90. B. K. Kumar, M. K. Mishra, K. S. Bhaskar and P. H. Vardhana, " PSO-based feedback controller design of DSTATCOM for load compensation with non-stiff sources", *International Journal of Power Electronics*, vol.1, no.2, pp.191 – 205, 2008.

91. X. Chen and Y. Li, "Neural network predictive control for Mobile Robot Using PSO with Controllable Random Exploration Velocity", *International journal of intelligent control and systems*, vol. 12, no. 3, , 2007, pp. 217-229.

92. X. Chen and Y. Li, "A Modified PSO Structure Resulting in High Exploration Ability with Convergence Guaranteed", *IEEE Transactions on systems, man and cybernetics*, vol. 37, no.5, pp. 1271-1289, 2007.

93. R. J. Pontt , C. A. Silva, P. Correa, P. Lezana, P. Cortes and U. Ammann, " Predictive current control of a voltage source inverter", *IEEE Transactions on Industrial Electronics*, vol. 54,no. 1,pp. 495-503, 2007.

94. S. Muller, U. Ammann and S. Rees, “New time-discrete modulation scheme for matrix converters”, *IEEE Transactions on industrial Electronics*, vol. 52,no. 6, pp.1607-1615, 2005.

95. J. C. P. Rodriguez, P. Antoniewicz and M. Kazmierkowski, “ Direct Power Control of an AFE Using Predictive Control”, *IEEE Transactions on Power Electronics*,vol. 23, no. 5,pp. 2516-2523, 2008.

96. P. Cortes, J. Rodriguez, D. E. Quevedo and C. Silva, “ Predictive current control strategy with imposed load current spectrum”, *IEEE Transactions on Power Electronics*, vol. 23, no. 2, pp. 612-618, 2008.

97. D. E. Quevedo and F. C. Goodwin, “ Multistep optimal analog to digital Conversion”, *IEEE Transactions on Circuits and Systems I.- Regular Papers*, vol. 52, no. 3, pp. 503-515, 2005.

98. R. Vargas, P. Cortes, U. Ammann, J. Rodriguez and J. Pontt , “ Predictive control of a three-phase neutral-point-clamped inverter”, *IEEE Transactions on Industrial Electronics*, vol. 54, no. 5, pp. 2697-2705, 2007.

99. S. Kouro, P. Cortes, R. Vargas, U. Ammann and J. Rodriguez, “Model Predictive Control—A Simple and Powerful Method to Control Power Converters”, *IEEE Transactions on Industrial Electronics, vol.* 56, no. 6, pp. 1826-1838, 2009.

100. P. Cortes, M. P. Kazmierkowski, R. M. Kennel, D. E. Quevedo and J. Rodriguez, “Predictive Control in Power Electronics and Drives”, *IEEE Transactions on Industrial Electronics* vol. 55, no. 12, pp. 4312-4324, 2008.

101. P. E. Kakasimos and A. G. Kladas, “Implementation of photovoltaic array MPPT through fixed step predictive control technique”, *Renewable Energy*, vol. 36, pp. 2508-2514, 2011.

102. M. Kumar and I. N. Kar, “Design of Model-Based Optimizing Control Scheme for an Air Conditioning System”, *HVAC&R Research*, vol. 16, no 5, pp. 565-597**,** 2010.

103. K.S. Narendra and K. parthasarathy, “Identification and control of dynamical system using neural networks”, *IEEE transactions on neural networks*, vol. 1, pp. 4-27, 1990.

104. S.I. Sudharsanan, I. Muhsin and M. K. sundareshan, “ Self tuning adaptive control of Multi input multi output nonlinear systems using multilayer recurrent neural networks with application to synchronous power generators”, *in Proc. IEEE international conference on neural networks*, Piscataway, NJ., USA, pp. 1301-1306, 1993.

105. S. R Chu and R. Shoureshi, “ Neural based identification of continuous nonlinear systems”, *in Proc. American control conference*, San Fransisco, California, pp. 1440,1444, 1993.

106. R. Adomaitis, R.M. Farber, J.L. Hudson, I.G. Kevrekidis, M. Kube, and A.S. Lapedes, “Application of neural nets to system identification and bifurcation analysis of real world experimental data”, *Neural networks: Biological computers or Electronic brains*, Springer –Verlag, Paris France, pp. 87-97, 1990.

107. S. Chen, S. A. Billings, C. F. N. Cowan and P.M. Grant, “ Practical identification of NARMAX models using Radial basis function”, *International journal of control*, vol. 52, no. 6, pp. 1327- 1350, 1990.

108. E. Levin, Gewirtzman and G.F. Inbar, “Neural network architecture for adaptive system modeling and control”, *Neural networks* , vol. 4, pp. 185-191, 1991.

109. S. Haykin, “Neural Networks a Comprehensive Foundation”, Englewood Cliffs, NJ: Prentice-Hall, 1999.

110. J. L. Elman, “Finding structure in time,” *Cognitive Science*, vol. 14, pp. 179–211, 1990.

111. A. F. Konar, Y. Becerikli, and T. Samad, “Trajectory tracking with dynamic neural networks,” in *Proc. IEEE International Symposium Intelligent Control*, Ístanbul, Turkey, pp. 173–180, 1997.

112. Z. G. Hou, M. M. Gupta, P. N. Nikiforuk, M. Tan, and L. Cheng, “A recurrent neural network for hierarchical control of interconnected dynamic systems”, *IEEE Transactions on Neural Networks*, vol. 18, no.2, pp.466–481, 2007.

113. M. S. Ballal, H. M. Suryawanshi, M. K. Mishra, "ANN based real time incipient fault detection and protection system for induction motor", *International Journal of Power and Energy Conversion*, vol. 1, no.2/3, pp.125 – 142, 2009.

114. H. Jack, D. M. A. Lee, R. O Buchal, W. H. Elmaraghy, " Neural networks and the inverse kinematics problem", *Journal of intelligent manufacturing*, vol. 4, pp-43-66, 1993.

115. A. Guez and J. Selinsky, "A neuromorphic controller with a human teacher", *in Proc. IEEE international conference on neural networks Icnn'88*, san Diego, California, vol. 2, pp. 595- 602,1988.

116. T. Troudet, W. C. Merrill, "Neuromorphic learning of continuous valued mapping in the presence of noise: application to real time adaptive control", *in Proc. IEEE international symposium on intelligent control*, Washington, D.C, National Aeronautics and Space Administration, pp.312- 319, 1989.

117. M. Norgaard, O. Ravn, N. K. Poulsen and L.K. Hansen, "Neural networks for modeling and control of dynamic systems: A practitioner's handbook", springer, Boston, MA,2000.

118. A. Olurotimi. Dahunsi and J. O. Pedro, "Neural network based identification and approximate predictive control of a servo hydraulic vehicle suspension system", Engineering letters, vol. 8, no. 4, 2010.

119. H. Demuth and M. Beale, "Neural networks toolbox user guide: For use with MATLAB", The Math Works, Inc., Natick, Massachusetts, 2002.

120. S. Haykin, "Neural Networks and learning machines", Pearson Education,Inc., New Jersey, 2009.

121. B. M. Akesson, H. T. Toivonen, "A Neural Network Model Predictive controller", *Journal of Process control*, vol.16, pp 937-946, 2006.

122. J.Taheri, A. Y. Zomaya, P. Bouvry and S. U. Khan, "Hopfield neural network for simultaneous job scheduling and data replication in grids", *Future Generation Computer Systems*, vol. 29, pp. 1885–1900, 2013.

123. R. Thangaraj, T. Chelliah, M. Pant, A. Abraham and P. Bouvry, “Applications of Nature Inspired Algorithms for Electrical Engineering Optimization Problems”, Editors: I. Zelinka, V. Snasel, A. Abraham, *Handbook of Optimization*, Springer-Verlag Berlin Heidelberg, pp. 991–1024, 2012.

124. R. Thangaraj, M. Pant, A. Abraham and V. Snasel, “ Modified particle swarm optimization with time varying velocity vector”**,** *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 1(A) , pp. 201-218, 2012.

125. Y. Shi and R. Eberhart, “Modified particle swarm optimizer,” in Proc. IEEE Conference on Evolutionary Computation, ICEC, pp: 69-73, 1998.

126. D. Zhao and P. Liang, “Support Vector Machine Predictive Control for Superheated Steam Temperature Based on Particle Swarm Optimization”, in proc. IEEE International conference, South China University of Technology, 2010.

127. W. Sun and Y. X. Yuan, “Optimization theory and methods: nonlinear programming”, Springer, 2006.

128. T. Howley, M. G. Madden, “The Genetic kernel Support vector machine: Description and Evaluation”, *Artificial Intelligence review*, vol. 24, pp. 379-395, 2005.

129. J. Stolfa, O. Koberskay, P. Kromer, S. Stolfa, M. Kopka and V. Snasel, “Comparison of Fuzzy Rules and SVM Approach to the Value Estimation of the Use Case Parameters”, *in Proc. IEEE, IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS),* pp. 789-794, 2013.

130. S. K. Aggarwal, L. M. Saini and A. Kumar, “Day-ahead Price Forecasting in Ontario Electricity Market Using Variable-segmented Support Vector Machine-based Model”, *Electric Power Components and Systems*, vol. 37, pp.495–516, 2009.

131. N. Cristianini, J. S. Taylor, “An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods”, Cambridge, U.K., Cambridge University Press, 2000.

132. M. Kumar and I.N. Kar, "Fault Diagnosis of an Air-Conditioning System Using LS-SVM", *Pattern recognition and machine intelligence-Lecture notes in computer science*, vol. 5909, pp. 555-560, 2009.

133. A. J. Smola and B. Scholkopf, "A Tutorial on support vector regression", *Statistics and computing*, vol. 14, no.3, pp. 199-222, 2004.

134. K. D. Brabanter, P. Karsmakers, F. Ojeda, C. Alzate, J. D. Brabanter, K. Pelckmans, B. De Moor, J. Vandewalle and J. A. K. Suykens, *"LS-SVM lab Toolbox User's Guide* ,version1.8, 2011.

135. C. Cercignani , "The Boltzmann equation and its applications", Berlin: Springer-Verlag; 1988.

136. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth and A. H. Teller, " Equations of state calculations by fast computing machines", *Journal of Chemical Physics*, vol. 21,no. 6, pp. 1087–1092, 1953.

137. S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, "Optimization by simulated annealing", Science, vol. 220, no. 4598, pp.671–680,1983.

138. P. F. Pai, W. C. Hong, "Support vector machine with simulated annealing algorithms in electricity load forecasting", *Journal of Energy conversion and management*, vol.46, no.17, pp.2669-2688, 2005.

139. M. E. Tipping, "Bayesian inference: An introduction to Principles and practice in Machine learning." Editors: O. Bousquet, U. V. Luxburg, and G. Ratsch , *Advanced Lectures on Machine Learning*, pp. 41-62. Springer, 2004.

140. C. M. Bishop, M. E.Tipping, "Variational Relevance Vector Machines", *in Proc. Uncertainity in Artificial Intelligence*, pp. 46-53, 2000.

141. J. O. Berger, "Statistical Decision Theory and Bayesian Analysis", Springer-Verlag, New York, Inc. 1980.

142. Z. Aydin, A. Singh, J. Bilmes, and W. S. Noble, "Learning sparse models for a dynamic Bayesian network classifier of protein secondary structure", *Bioinformatics*, vol.12, no. 154, pp.1-21, 2011.

143. Y. Chen, B. Yang, A. Abraham and L. Peng, “Automatic Design of Hierarchical Takagi–Sugeno Type Fuzzy Systems Using Evolutionary Algorithms”, *IEEE Transactions on fuzzy systems*, vol. 15, no. 3, 2007.

144. S. Bansal, L. M. Saini, D. Joshi, “ Design of PI and Fuzzy Controller for High-Efficiency and Tightly Regulated Full Bridge DC-DC Converter” *World Academy of Science, Engineering and Technology, International Journal of Electrical, Electronic Science and Engineering*, vol:7, no:4, pp. 100-106, 2013.

145. S. R. Kolla and L. Varatharasa, “Identifying three-phase induction motor faults using artificial neural networks” *ISA Transactions*, vol. 39, pp: 433- 439, 2000.

146. S. R. Kolla, "Fuzzy logic control of an electric motor drive system," in Proc. of Electrical Manufacturing & Coil Winding Conference," Cincinnati, OH, October 6-8, 1998.

147. R. Thangaraj, M. Pant, A. Abraham and P. Bouvry, “Particle swarm optimization: Hybridization perspectives and experimental illustrations”, *Applied Mathematics and Computatio*n, vol. 217, no.12, pp. 5208–5226, 2011.

148. I. H. Altas, A. M. Sharaf, “A novel on-line MPP search algorithm for PV arrays”, *IEEE Transactions on Energy Conversion*, vol. 11, no. 4, pp. 748–754, 1996.

149. C. Hua, J. Lin and C. Shen, “ Implementation of a DSP-controlled photovoltaic system with peak power tracking”, *IEEE Transactions on industrial Electronics*, vol. 45,no. 1,pp. 99–107, 1998.

150. O. Wasynczuk, “Dynamic behavior of a class of photovoltaic power Systems”, *IEEE Transactions on Power Apparatus and Systems*, vol.102, no.9, pp. 3031–3037, 1983.

151. S. Azadeh and S. Mekhilef, “ Simulation and Hardware Implementation of Incremental Conductance MPPT with Direct Control Method Using Cuk Converter”, *IEEE Transaction on Industrial Electronics*, vol. 58, no.4, pp.1154-1161, 2011.

152. I. Houssamo, F. Locment and M. Sechilariu, “Maximum power tracking for photovoltaic power system: development and experimental comparison of two algorithms”, Renewable Energy, vol. 35, no.10, pp. 2381 -2387, 2010.

153. P. Kromer, V. Snasel, J. Platos, A. Abraham, L. Prokop and S. Misak, "Genetically Evolved Fuzzy Predictor for Photovoltaic Power Output Estimation", in Proc. 3rd IEEE International Conference on Intelligent Networking and Collaborative Systems, pp. 41-46, 2011.

154. T. Hiyama, S. Kouzuma, T. Iimakudo, "Identification of optimal operating point of PV modules using neural network for real time maximum power tracking control", *IEEE Transactions on Energy Conversion*,vol. 10, no.2, pp. 360–367, 1995.

155. T. Hiyama, S. Kouzuma, T. Imakubo, T. H. Ortmeyer, " Evaluation of neural network based real time maximum power tracking controller for PV system", *IEEE Transactions on Energy Conversion*, vol. 10, no. 3, pp. 543–548, 1995.

156. T. Hiyama and K. Kitabayashi, "Neural network based estimation of maximum power generation from PV module using environmental information", *IEEE Transactions on Energy Conversion*, vol. 12, no. 3, pp. 241–247, 1997.

157. J. Shi, W. Lee, Y. Liu, Y. Yang and P. Wang, "Forecasting power output of photovoltaic systems based on weather classification and support vector machines", *IEEE Transactions on industry applications*, vol. 48,no.3, pp. 1064-1069, 2012.

158. A. D. Karlis, T. L. Kottas, Y. S. Boutalis, " A novel maximum power point tracking method for PV systems using fuzzy cognitive networks (FCN)",. *Electric Power Systems Research* vol. 77, pp. 315–327, 2007.

159. C. Chiu, "T-S fuzzy maximum power point tracking control of solar power generation systems", *IEEE Transactions on Energy Conversion*, vol. 25, no. 4, pp.1123-1132, 2010.

160. N. Gounden, S. Annpeter, H. Nallandula and S. Krithiga, "Fuzzy logic controller with MPPT using line-commutated inverter for three-phase grid-connected photovoltaic systems", *Renewable Energy*, vol. 34, no.3, pp. 909 -915, 2009.

161. C. Larbes, S. A. Cheikh, T. Obeidi and A. Zerguerras, " Genetic algorithms optimized fuzzy logic control for the maximum power point tracking in photovoltaic system", *Renewable Energy*, vol. 34, pp. 2093 -2100, 2009.

162. L. Chen, C. Tsai, Y. Lin and Y. Lai, “A biological swarm chasing algorithm for tracking the PV maximum power point”, *IEEE Transactions on Energy Conversion*, vol. 25, no.2, pp. 484-493, 2010.

163. S. L. Brunton, C. W. Rowley, S. R. Kulkarni and C. Clarkson, “Maximum power point tracking for photovoltaic optimization using ripple based extremum seeking control”, *IEEE transactions on power electronics*, vol. 25, no.10,pp. 2531-2540, 2010.

164. Y. H. Liu, C. L. Liu, J. W. Huang and G. H. Chen, “Neural network based maximum power point tracking methods for photovoltaic systems operating under fast changing environments”, *Solar energy*, vol. 89, pp. 42-53, 2013.

165. M. Khalid and A. V. Savkin, “A model predictive control approach to the problem of wind power smoothing with controlled battery storage”, *Renewable Energy*, vol. 35, pp. 1520-1526, 2010.

166. C. Hua, C. Wu and C. Chuang, “A digital predictive current control with improved sampled inductor current for cascaded inverters”, *IEEE Transactions on Industrial Electronics*, vol. 56, no. 5, pp. 1718-1726, 2009.

167. J. Rodriguez, M. P. Kazmierkowski, J. R. Espinoza, P. Zanchetta, H. Abu-Rub, H. A. Young, and C. A. Rojas, “State of the Art of Finite Control Set Model Predictive Control in Power Electronics”, *IEEE Transactions on Industrial Informatics*, vol. 9, no. 2, pp. 1003-1016, 2013.

168. M. V. Cleef, P. Lippens and J. Call, “Superior energy yields of UNI-SOLAR_ triple junction thin film silicon solar cells compared to crystalline silicon solar cells under real outdoor conditions in Western Europe,” *in proc. 17th European Photovoltaic Solar Energy Conference and Exhibition*, Munich, 2001.

169. E. Bum, N. Cereghetti, D. Chianese, A. Realini, and S. Reuonieo, “PV module behavior in real conditions: Emphasis on thin film modules”, Ph. D. dissertation, CH-Testing Centre for PV-Modules, University of Applied Sciences of Southern Switzerland.

170. M. G. Villalva, J. R. Gazoli and E. R. Filho, “Comprehensive approach to modeling and simulation of photovoltaic arrays”, *IEEE Transactions on Power Electronics*, vol. 24, no. 5, pp. 1198-1208, 2009.

171. S. Chowdhury , G. A. Taylor, S. P. Chowdhury, A. K. Saha, Y. H. Song, “Modelling, simulation and performance analysis of a PV array in an embedded environment”, *in Proc. 42nd International universities power engineering conference. (UPEC)*, Brunel University, London, pp. 781–785, 2007.

172. A.D. Hansen, P. Sorensen, L.H. Hansen, and H. Bindner, “Models for a stand-alone PV system”, Riso National Laboratory, Roskilde, 2000.

173. T. Esram, and P.L. Chapman, “Comparison of photovoltaic array maximum power point tracking techniques,” *IEEE Transactions on Energy Conversion*, vol. 22, no. 2, pp. 439–449, 2007.

174. A.K. Sahani, S.K. Nagar, “Design of digital controllers for multivariable systems via time-moments matching”, *Computers & Electrical Engineering*, vol. 24, pp. 335-347, 1998.

175. B. Bandyopadhyay, H. Unbehauenu and B. M. Patre, “Control of higher order system via its reduced model”, *in Proc. 10th IEEE international conference on global connectivity in energy, computer communication and control*, vol. 1, pp. 226-229, 1998.

176. B. Bandyipdhyay, F. Deepak, K. S. Kim, “Sliding mode control using novel sliding surfaces Lecture Notes in Control and Information Sciences”, Editors: M. Thoma, F. Allgower, M. Morari, Springer-Verlag Berlin Heidelberg, 2009.

177. F.G.Sinskey, “Process control systems: Applications, Design and tuning”, 4th edition, McGraw-Hill, New York,1979.

178. E. Eskinat, S. H. Johnson and W. L. Luyben, “Use of Auxillary information in system identification”, *Industrial Engineering and Chemistry Research*, vol. 32, pp. 1981-1992, 1993.

179. W.T. Wu, J.W. Ko and H.G.Lee, “Decoupling control of Multivariable system with a Desensitizer”, *Industrial and Engineering Chemistry research*, vol. 32, pp. 2937-2941, 1993.

180. S. Skogestad and M. Morari, “Understanding the Dynamic Behavior of Distillation Columns”, *Industrial Engineering Chemistry Research*, vol. 27, pp. 1848-1862, 1988.