Nonlinear model predictive control of MIMO system with Least squares support vector machines and Particle swarm optimization

M. Germin Nisha*, G. N. Pillai**

** Research Scholar, Department of Electrical Engineering, Indian Institute of Technology Roorkee, Roorkee-247667, Uttarakhand, India, Email: germinnisha@gmail.com*
*** Professor, Department of Electrical Engineering, Indian Institute of Technology Roorkee, Roorkee-247667, Uttarakhand, India, Email: gopinath.pillai@gmail.com*

Abstract: This paper demonstrates control accuracy and computational efficiency of nonlinear model predictive control (NMPC) strategy which utilizes a deterministic sparse kernel learning technique called Support vector regression (SVR) and particle swarm optimization with controllable random exploration velocity (PSO-CREV). An accurate reliable nonlinear model is first identified by SVR with a radial basis function (RBF) kernel and then the optimization of control sequence is speeded up by PSO-CREV. An improved system performance is guaranteed by an accurate sparse predictive model and an efficient and fast optimization algorithm. To compare the performance, model predictive control (MPC) using neural network (NN) model is done on a highly nonlinear distillation column with severe interacting process variables. SVR based MPC shows improved tracking performance with very less computational effort which is much essential for real time control.

Keywords: Least squares support vector machines; Nonlinear model predictive control; PSO-CREV.

1. INTRODUCTION

Model predictive control (MPC) is recognized as one of the advanced control technique which has been very successful in practical applications (Qin and Badgwell, 2003). This acknowledgment is due to its ability to handle constraints imposed on process inputs and outputs, interactions between process variables, process nonlinearities, dead times, and model uncertainties. MPC algorithm has the capability of controlling multi input, multi output (MIMO) nonlinear processes with significant time-delays and process interactions more efficiently.

In earlier times linear model predictive controllers were repeatedly used in practice. But linear model predictive controllers fail to experiment the inevitable nonlinear behaviour of chemical processes. Linear model predictive controller is inadequate for highly nonlinear processes and moderately nonlinear processes which have large operating regimes. This shortcoming coupled with increasingly stringent demands on throughput and product quality has spurred the development of nonlinear model predictive control (Henson, 1998). Two challenging tasks in nonlinear model predictive controller are acquiring an accurate nonlinear model and solving nonlinear optimization problem online.

The performance of nonlinear model predictive controller depends on model accuracy. For a highly tuned controller a very accurate model is necessary (Rossiter, 2003). Thus precise nonlinear model is expected for better controlled performance. Neural networks were widely believed for estimation of nonlinear system dynamics due to its simplicity besides its poor extrapolation, poor generalization. Moreover, training a neural network is too lengthy and the number of training data required is more (Liu et al., 2010). Several scholars (Bhat and Mcavoy, 1990; Psichogios and Ungar 1991; Hunt et al., 1992) have approximated nonlinear models by neural networks which paid acceptable performance. Despite the existence of many nonlinear control strategies in theory, designing a suitable controller for complex process is still a challenge in practice (Liu et al., 2010).

The sparse kernel learning is a nonlinear modeling method originally proposed in the machine learning area (Taylor and Cristianini, 2004; Bishop, 2006). A deterministic nonlinear modeling method, support vector machines (SVM) which overwhelms the over fitting and poor generalization ability of neural network with less number of training data and less training time providing better tracking performance is introduced in (Vapnik, 1998). The guaranteed model accuracy, better extrapolation and generalization capability of SVR model are explicitly acknowledged by many researchers (Zhang and Wang, 2006; Kulkarni et al., 2003; Zhong et al., (2005a, b); Yue-hua et al., 2007; Xue-Cheng et al., 2007 which highlights its significance. The complexity of developing an accurate model for the distillation column and the nonlinearities of its dynamics, make very attractive the use of support vector machine.

Despite of accurate approximation of nonlinear dynamics it suffers from computational burden as model predictive controller does prediction and optimization at each sampling instant. Zhong et al., (2005a, b) has solved the cost function of SVM based MPC by Nelder-Mead simplex direct search method. (Yue-hua et al., 2007) has optimized the performance index by genetic algorithm which has more computational effort when compared with particle swarm

optimization. (Xue-Cheng et al., 2007) obtained the control sequence by dynamic programming in which selection of sub problems and ordering are tough tasks.

The particle swarm optimization is an attractive tool owing to its simplicity and high performance, it has been proven to be a powerful competitor to other evolutionary algorithms (Eberhart and Kennedy, 1995; Kennedy and Eberhart, 1995) and been widely used in many optimization processes (Yoshida et al., 1999; Messerschmidt and Engelbrecht, 2004). It is a computationally efficient method since it is a derivative free method.

Chen and Li, (2007a, b) developed 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.

In this paper, a nonlinear model predictive controller combining support vector regression model and particle swarm optimization with controllable random exploration velocity (PSO-CREV) is presented; which merges the advantage of accurate prediction and less computational effort. Simulation results of a highly nonlinear multi input multi output (MIMO) distillation column process with severe interacting process variables illustrates the better tracking performance of SVM based MPC when compared to neural network based MPC.

This paper encompasses five sections commencing with the introduction as the first section followed by the second section which describes least squares support vector machines. The third section explains MPC based on LS-SVM and particle swarm optimization. The fourth section shows a comparative study of a highly nonlinear distillation column process with suitable simulation results of LS-SVM based MPC and NN based MPC and the fifth section concludes the paper.

2. LEAST SQUARES SUPPORT VECTOR MACHINES

Support Vector learning is based on simple ideas which originated in statistical learning theory (Bishop, 2006). 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 characterized by usage of kernels, absence of local minima, sparseness of the solution and capacity control obtained by acting on the number of support vectors. Although SVM's are being used mainly for classification tasks, recently SVM's have been successfully extended to solve regression problems (Karatzoglou and Meyer, 2006).

Consider a given 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 actual plant and $y_i \in R$ is the output data of the actual plant.. In high dimensional feature space *Z*, LS-SVM model is,

$$y(x) = w^T \ (x) + b \text{ where } \quad w \in Z, b \in R \tag{1}$$

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

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

This is subjected to the following constraints

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

where

γ - regularization parameter.

E_i - error between the actual output and predicted output of the i[th] data.

The Lagrange function for equation (2) is

$$L = \frac{1}{2} w^T w + \frac{\gamma}{2} \sum_{i=1}^{M} E_i^2 - \sum_{i=1}^{M} \alpha_i [w^T \varphi(x_i) + b + E_i - y_i] \tag{4}$$

where α_i -Lagrange multiplier.

According to Karush–Kuhn–Tucker conditions,

$$\begin{aligned} &\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{M} \alpha_i \varphi(x_i) \\ &\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{M} \alpha_i = 0 \\ &\frac{\partial L}{\partial \alpha_i} = 0 \Rightarrow w^T \varphi(x_i) + b + E_i = y \\ &\frac{\partial L}{\partial E_i} = 0 \Rightarrow \alpha_i = \gamma E_i, i = 1,2,...., M \end{aligned} \tag{5}$$

All the above equations in Equation (5) are first transformed into a matrix form and then substituting the values of E and w results in the following matrix equation,

$$\begin{bmatrix} 0 & \vec{1}_M^{\,T} \\ \vec{1}_M & \varphi(x_i)^T \varphi(x_j) + \gamma^{-1} I_M \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \tag{6}$$

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 are* any kernel function satisfying the Mercer condition (Smola and Scholkopf, 2004).

The parameters α and b can be obtained as a solution of Equation (5) and hence LS-SVM predicted model of the given dataset is as follows,

$$\hat{y}(x) = \sum_{i=1}^{M} \alpha_i K(x_i, x) + b \tag{7}$$

In the present work Radial basis function (RBF) in Equation (8) is selected as kernel function because of its ability to reduce computational complexity of the training process and to improve generalization ability of LS-SVM.

$$k(x_i, x_j) = \exp\{ \|x - x_i\|^2 / \sigma^2 \} \tag{8}$$

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 (De Brabanter et al., 2011). 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. MPC BASED ON RVM AND PARTICLE SWARM OPTIMIZATION

3.1 LS-SVM Based MPC Principle

The basic structure of LS-SVM based nonlinear model predictive controller is shown in Fig. 1. It includes three important blocks, the actual plant to be controlled with output *y(k)*. The LS-SVM model of the actual plant to be controlled with predicted output $\hat{y}(k) = [\hat{y}(k+1) / k \ ,\ldots \ \hat{y}(k+N_p) / k]$ here, N_p 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 $u(k)=[u(k/k),\ldots u(k+ N_u -1 /k)]$ where N_u is the control horizon of MPC which dictates the number of control moves used to attain the future control trajectory, subjected to the specified constraints which is required for the plant to achieve the desired trajectory $ref(k)=[ref(k+1) \ \ldots . ref(k+Np)]$. 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 minimize the formulated performance index in (11) i.e. the difference between the predicted output of the model and the desired reference trajectory over the specified prediction horizon N_p.

The number of manipulated variable in the sequence is decided by the control horizon value N_u and only the first manipulated variable is applied to the actual plant. This course is repeated at each sampling instant.

The basic structure of neural network based nonlinear model predictive control is obtained by replacing LS-SVM model by neural network model in Fig. 1.

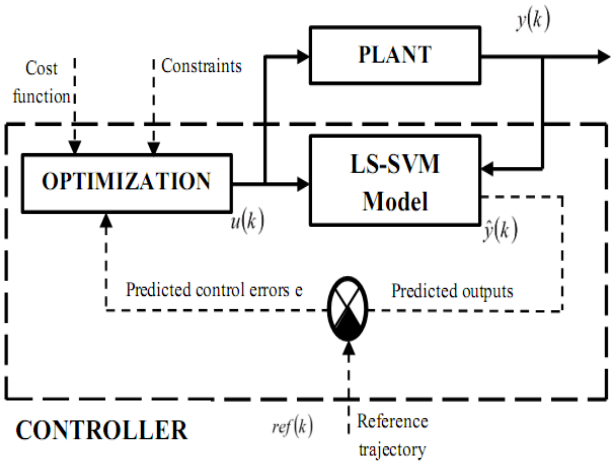

Fig. 1. Basic structure of LS- SVM based nonlinear model predictive control.

3.2 Performance index formulation

For a MIMO $n \times m$ nonlinear process the predicted outputs of LS-SVM model from Equation (7) 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, $U(k-1)=[u_1(k-1)\ldots u_1(k-n_u+1),u_2(k-1)\ldots u_2(k-n_u+1),\ldots.,u_n(k-1)\ldots u_n(k-n_u+1)]$. 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 following discrete time model,

$$\begin{aligned} \hat{y}_1(k+1) &= f[Y_1(k), u(k), U(k-1)], \\ \hat{y}_2(k+1) &= f[Y_2(k), u(k), U(k-1)], \\ &\ldots\ldots\ldots \\ \hat{y}_m(k+1) &= f[Y_m(k), u(k), U(k-1)]. \end{aligned} \tag{9}$$

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 perform a linear regression in this feature space. Thus after system identification with the regression data set, prediction of each output could be formulated as

$$\hat{y}_j(k+1) = \sum_{i=1}^{M} \alpha_i K(x_i, x) + b \tag{10}$$

where $j=1\ldots m$ and *M* is the number of subsets of training samples.

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

$$J = \sum_{j=1}^{m}\sum_{i=N_1}^{N_2} q_j\left[ref_j(k+i) - \hat{y}_j(k+i)\right]^2 + \sum_{j=1}^{n}\sum_{i=1}^{N_u} \lambda_j\left[\Delta u_j(n+i)\right]^2 \quad (11)$$

where

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 LS-SVM model
$\Delta u_j(.)$	-	change of j^{th} control input defined as $u_j(k+i)- u_j(k+i-1)$
k	-	current sampling instant
q_j, λ_j	-	time independent weighting coefficients.

In the performance index formulated in Equation (11), $\hat{y}$ depends on the kernel function which in turn is a function of manipulated variable u, which is optimized and applied to the actual plant in order to minimize the deviation between the reference value and controlled variable.

3.3 Conventional Particle swarm optimization

Although nonlinear predictive controller is good at controlling unknown nonlinear systems, it does not mean that practical implementation is without difficulties. The primary shortage results from its computational cost (Chen and Li, 2007a, b). Usage of evolutionary algorithm for MPC optimization overcomes this difficulty. Inspired by the foraging behaviour of birds, American psychologist Kennedy and electrical engineer Eberhart developed the particle swarm optimization algorithm (Kennedy and Eberhart, 1995). This evolutionary algorithm has the capability of universality and global optimization.

If in an n dimensional search space, the swarm $X=[X_1,...,X_2,...,X_m]$ is composed of m particles. Let the position and velocity of i^{th} individual particles be $X_i=[x_{i1},x_{i2},...x_{in}]^T$ and $V_i=[v_{i1},v_{i2},...v_{in}]^T$ respectively and the best position be $P_i=[P_{i1},P_{i2},...,P_{in}]^T$. Let the global best position, gbest be $P_g=[p_{g1},p_{g2},...p_{gn}]^T$. Then the updated velocity and position of particle X_i will be as in Equation (12) and Equation (13).

$$v_{id}^{(t+1)} = \omega v_{id}^{(t)} + C_1 r_1 (P_{id}^{(t)} - X_{id}^{(t)}) + c_2 r_2 (P_{gd}^{(t)} - X_{id}^{(t)}) \quad (12)$$

$$x_{id}^{(t+1)} = x_{id}^{(t)} + v_{id}^{(t+1)} \quad (13)$$

where $d=1,2,...,n,\ i=1,2,...m,$

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.

Usually the learning factors c_1 and c_2 ranges between [0, 4].

3.4 Disadvantages of Conventional PSO

From Equation (18) and Equation (19) it is understood that the strength of exploration performance is merely determined by the degrading rate 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. Hence if a swarm converges to a local minimal solution, the algorithm may not have the capability to neglect it and hence the strength of exploration behaviour of the conventional PSO algorithm needs improvement. This task of improving the exploration strength is achieved in a modified novel algorithm PSO-CREV.

3.5 PSO-CREV Algorithm

The intensity of exploration capability of conventional PSO was improved significantly by Chen and Li, (2007a, b), after incorporating some modifications in the position and velocity equations as shown in Equation (14) and Equation (15) respectively.

$$v_{id}^{(t+1)} = \varepsilon^{(t)}[\omega v_{id}^{(t)} + c_1 r_1 (P_{id}^{(t)} - X_{id}^{(t)}) + c_2 r_2 (P_{gd}^{(t)} - X_{id}^{(t)}) + \xi_{id}^{(t)}] \quad (14)$$

$$x_{id}^{(t+1)} = \alpha x_{id}^{(t)} + v_{id}^{(t+1)} + \frac{1-\alpha}{\phi_{id}^{(t)}}[c_1 r_1 P_{id}^{(t)} + c_2 r_2 P_{gd}^{(t)}] \quad (15)$$

$\xi_{id}^{(t)}$ - Bounded random variable with continuous uniform distribution,

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

α - ranges between 0 and 1.

In order to achieve the global optimal solution the random velocity $\xi_{id}^{(t)}$ is introduced to enhance the particles to reach the strange solution space which might be very close to the global optimal solution. On the other hand, a time-varying bond of random search velocity 'ξ' can meet strong exploration ability and fast convergence.

Hence,

$$\xi(n) = w(n)\bar{\xi}(n)$$

Here the time varying positive coefficient $w(n)$ is the one to be adjusted according to the requirements of optimization problem.

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

N_b	-	Total number of iterations.
λ_1, λ_2	-	Positive constants less than one which makes a decreasing random velocity.

The values chosen for N_b ,λ_1, λ_2 are 60, 0.99 and 0.98 respectively. Hence during iterations 1 to 15, the intension of ξ (n) is strong, so that particles have more opportunities to reach unknown solution space. And after that, the bound of ξ(n) decreases iteration by iteration on different rates within different periods, especially in the last quarter of iterations, ξ (n) has trivial effect on the convergence of PSO-CREV. Such a time-varying bound of random search velocity makes PSO-CREV meet both the requirements of strong exploration ability and fast convergence.

A dynamic strategy is selected for $\varepsilon(n)$ as $\varepsilon(n)=\frac{a}{(n+1)^b}$ where a and b are scalars. To balance the exploration capability and convergence speed the value of a and b must be properly selected (Chen and Li, 2007b). The larger the value of a is, the stronger the divergence behaviour of PSO-CREV. That means a large a makes particle disperse widely. At the same time, b plays a very important role for determining the convergence speed. A small b makes PSO-CREV converge slowly, because $\varepsilon(n)$ decreases slowly. Hence, there exists a dilemma in choosing b. Choosing a small b makes PSO-CREV with strong exploration ability but weak convergent speed, while large value of b makes PSO-CREV with the reverse character. To balance exploration and convergence speed, the values of a and b are chosen as 3.5 and 0.4 respectively.

The learning factors c_1 and c_2 are chosen as 2. The variables r_1 and r_2 are random numbers which ranges between 0 and 1. Due to the better extrapolation capability of PSO-CREV even 30 swarm size is able to reach good convergence. The inertia weight w, controls the convergence behaviour of PSO whose value is gradually reduced from 1 for refined solutions.

The nonlinear model predictive control algorithm incorporated here utilizes this PSO-CREV optimization technique. The experiment is performed for 15 trials (the performances were not much varying) and the average of the 15 results were taken.

4. APPLICATION ON BINARY DISTILLATION COLUMN PROCESS

This section describes the better accuracy and less computational demand of LS-SVM based nonlinear model predictive control (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. 2. Two conventional controllers denoted by *LC* are used to maintain the levels in the reflux tank and bottom product tank. 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 critical controller performance attributes of set point tracking and disturbance rejection are presented through simulations.

The binary distillation column considered is under LV – configuration (Skogestad and Morari, 1988). 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.

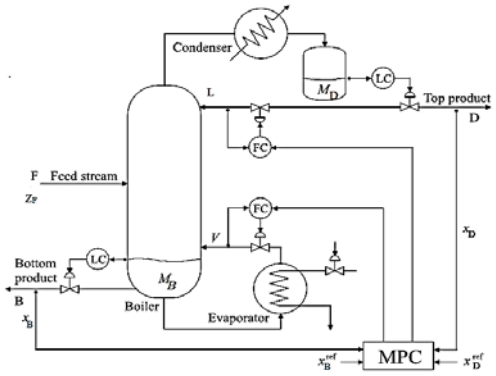

Fig. 2. Schematic of the binary distillation column process.

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	-	total number of theoretical trays
N_F	-	Feed tray location 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;

$$i = 2, N \ (i \neq N_{F,} \ i \neq N_F + 1)$$

$$M_i \dot{x}_i = L_{i+1} x_{i+1} + V_{i-1} y_{i-1} - L_i x_i - V_i y_i \quad (16)$$

above feed location $i = N_F + 1$

$$M_i \dot{x}_i = L_{i+1} x_{i+1} + V_{i-1} y_{i-1} - L_i x_i - V_i y_i + F_v y_F \quad (17)$$

below feed location, $i = N_F$

$$M_i \dot{x}_i = L_{i+1} x_{i+1} + V_{i-1} y_{i-1} - L_i x_i - V_i y_i + F_L x_F \quad (18)$$

reboiler, $i = 1$

$$M_B \dot{x}_i = L_{i+1} x_{i+1} - V_i y_i - B x_i, \qquad x_B = x_1 \quad (19)$$

total condenser, $i = N + 1$

$$M_D \dot{x}_i = V_{i-1} x_{i-1} - L_i x_i - D x_i, \qquad y_D = x_{N+1} \quad (20)$$

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

$$y_i = \alpha x_i / (1 + (\alpha - 1) x_i) \quad (21)$$

Flow rates above and below feed trays assuming constant molar flows are,

$i > N_F$ above feed, $L_i = L, \ V_i = V + F_V$ (22)

$i \leq N_F$ below feed, $L_i = L + F_L, V_i = V$ (23)

$$F_L = q_F F \ , F_V = F - F_L \quad (24)$$

condenser holdup is kept constant,

$$D = V_N - L = V + F_V - L \quad (25)$$

reboiler holdup is kept constant,

$$B = L_2 - V_1 = L + F_L - V \quad (26)$$

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

$$F Z_F = F_L x_F + F_V y_F \quad (27)$$

$$y_F = \alpha x_F / (1 + (\alpha - 1) x_F) \quad (28)$$

The nonlinear differential equations described form equation (16 – 28) are based on first principle model. The binary distillation column model considered under *LV*- configuration contains a total of 41 stages including the reboiler and total condenser. Thus 41 nonlinear differential equations are used to describe the system dynamics. Development of such a model is usually costly, time consuming and effort demanding.

The model derived in such a way is of very high order because of thorough modeling and hence if such a model is used for prediction in NMPC, optimization problem in NMPC becomes a complex task.

An upright support vector regression (SVR) model developed for this process has a specific advantage of sparseness of the solution. Which means SVM's solution depends on the support vectors and not on the whole data set. As a result the overall computation in least square support vector machine (LS-SVM) based NMPC is made simple with very less time consumption.

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 $2.5 \leq u_1(t) \leq 2.9$ and $3 \leq u_2(t) \leq 3.5$ respectively. (33) In order to capture the dynamics of binary distillation column model using SVR model, two past outputs and past inputs are sufficient hence the following second order model is chosen.

$$y_1(k) = f(y_1(k-1), y_1(k-2), u_1(k-1), u_1(k-2), u_2(k-1), u_2(k-2)) \quad (29)$$

$$y_2(k) = f(y_2(k-1), y_2(k-2), u_1(k-1), u_1(k-2), u_2(k-1), u_2(k-2)) \quad (30)$$

A sequence of 100 samples with two delay regression vector format is used to train the SVR model offline using the leave one out method. Leave one out method is one in which the function approximator is trained on all the data except for one point and the prediction is made for that point. This procedure is repeated for each data point. The average error is computed by combining the different estimate of the performance and used to evaluate the model. The assumption is made that the input data is distributed independent and identically over the input space ((De Brabanter et al., 2011).

In the case of 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 SVR model and NN model are assessed by the root mean square error (RMSE) performance function.

$$RMSE = \left\{ \sum_{k=1}^{N} [\hat{y}(k) - y(k)]^2 \Big/ N \right\}^{1/2} \quad (31)$$

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

Here the process inputs u_1, u_2, process outputs y_1, y_2, predicted output $\hat{y}$ and sampling instant *k* are dynamic variables which depend on time. *N* represents the total number of samples used for simulation which is time independent

The identification performance of SVR model and NN model are assessed by the root mean square error (RMSE) performance function. The input variables, u_1, u_2, output variables y_1, y_2 predicted output $\hat{y}$ and the sampling instant *K* are time dependent dynamic variables. *N* represents the total number of samples used for simulation which is time independent.

Table 1. Accuracy of SVR and NN model of binary distillation column process.

Model	$RMSE_{training}$		$RMSE_{testing}$	
	x_D	x_B	x_D	x_B
SVR	0.0027	0.0028	0.0028	0.0030
NN	0.0026	0.0028	0.0153	0.0134

Fig. 3 and Fig. 4 correspond to the modeling results of SVR and NN methods. While modeling the training set, NN model and SVR model attains almost same identification performance. But, for the test data which are beyond the training data, the SVR model can achieve much better performance than NN. The comparative graph of prediction errors of SVR model and NN model for test data are shown in Fig. 4, which explores the better extrapolation capability of SVR model than NN model. Accuracy of the model in terms of RMSE (31) is tabulated in Table 1. Thus one can conclude that the SVR based empirical modeling can prevail over the poor extrapolation capability and over fitting problem of NN modeling.

The offline trained and validated SVR model or NN model is then used as the nonlinear model for nonlinear MPC. Fig. 5 illustrates the random set point tracking performances of SVR based MPC and NN based MPC.

Certainly the tracking performance of SVR based MPC is much better with less oscillations and faster settling time when compared with NN based MPC even in the presence of severe interacting process variables. 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 SVR-PSO-CREV and NN-PSO-CREV are with very less fluctuations as shown in Fig.6 presenting the index of control performance.

The unmeasured disturbance rejection capability of SVM-PSO-CREV based MPC and NN-PSO-CREV based MPC are compared 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 at different sampling instance are shown in Fig. 7.

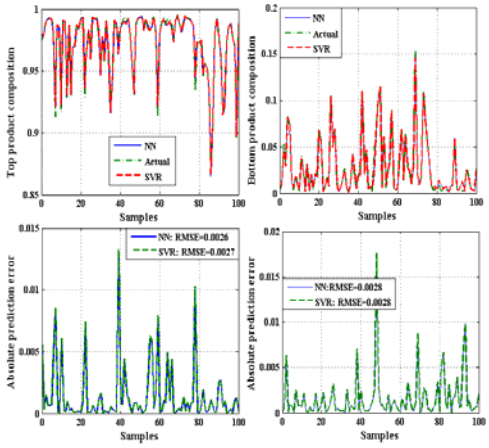

Fig. 3. Training performance comparison of SVR and NN models.

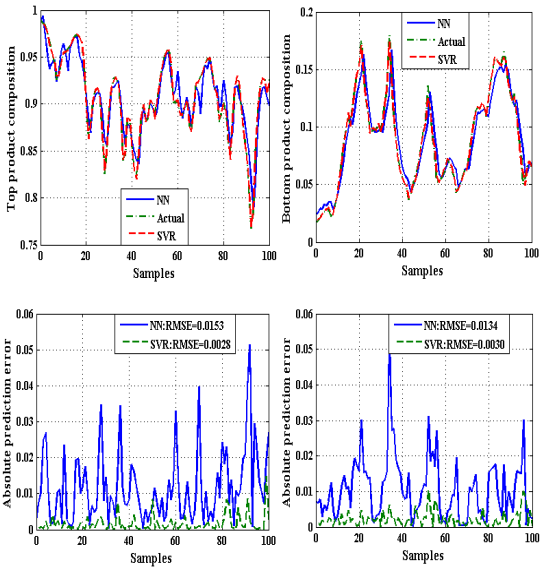

Fig. 4. Testing performance comparison of SVR and NN models.

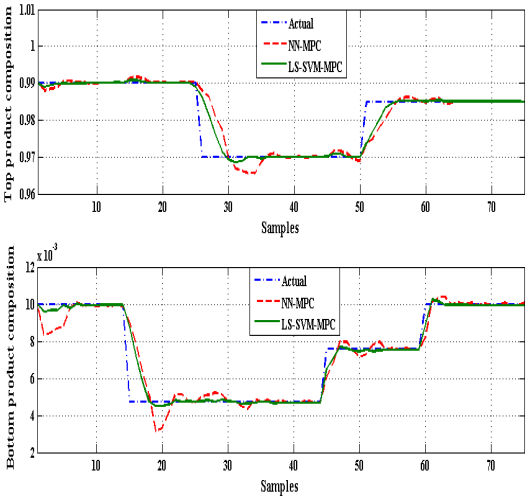

Fig. 5. Set point tracking performance of distillation column process by LS-SVM-MPC and NN-MPC.

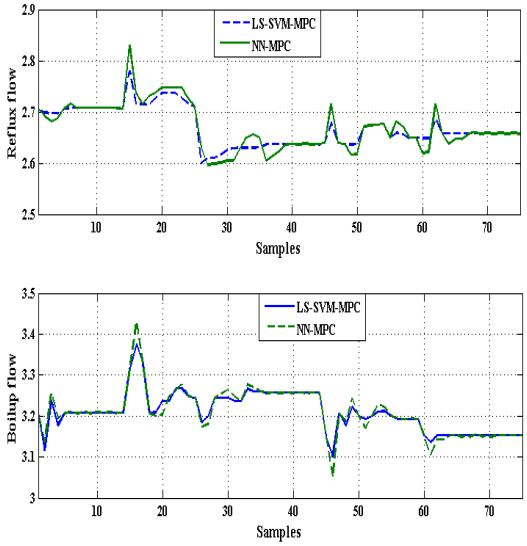

Fig. 6. Changes in the process variables for tracking x_D and x_B of distillation column process.

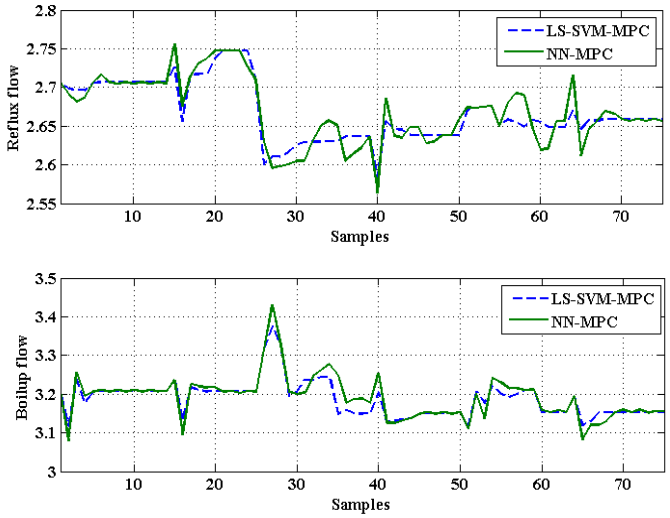

Fig. 7. Changes in the process variable to show unmeasured disturbance.

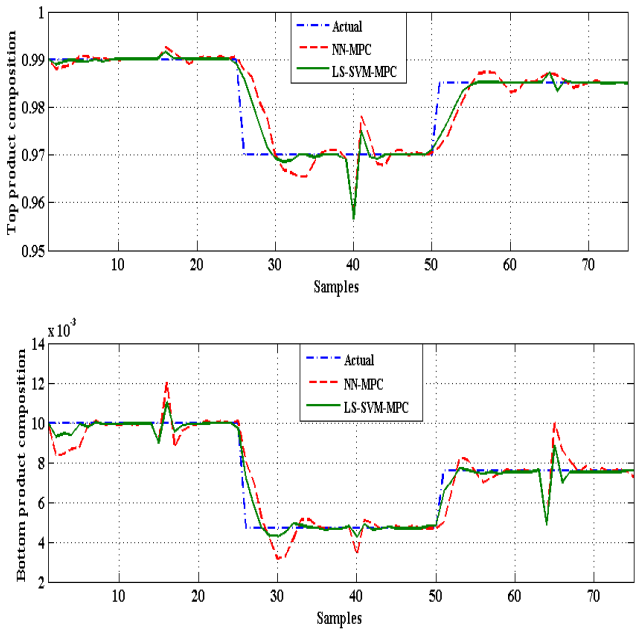

Fig. 8. Performance comparison of unmeasured disturbance rejection.

Certainly the unmeasured disturbance rejection performance of SVR-PSO-CREV based MPC is better when compared to NN-PSO-CREV based MPC as shown in Fig. 8. Thus the better capability of LS-SVM based MPC; in overcoming the interaction among process variables are vibrant from the simulation results. Accordingly SVR-PSO-CREV based MPC behaves suitably for process control industrial applications.

4.2 Tabulation of performance indices for different controlling techniques

This section enunciates the performance indices and computational cost of the controllers discussed in previous section. Integral absolute error (IAE) is the performance criteria which quantifies the accuracy of all controllers. Table.2 shows the IAE value and computational time related to each controller for the simulation results carried out for 75 samples.

The distillation column model under simulation has very slow time constants on the order of minutes. The sparseness property of SVR model sharply reduces the computational time of SVR-MPC to 31.06 seconds for 75 samples (ie., nearly 0.414 Seconds for each sample), which is much shorter than the sampling time of the distillation column process. Instead in NN based NMPC the computation times for 75 samples is 65.51 seconds with the sampling period of 0.874 seconds. Hence, it is clear that NN based MPC is the one which consumes more time with more IAE and SVR-PSO-CREV model predictive controller is the better controller with less computational load and less IAE.

Thus SVM based MPC performs better based on various attributes like usage of less number of training data, less training time, better prediction accuracy, better generalization and extrapolation capability, excellent set point tracking performance, better unmeasured disturbance rejection capability. Hence it is well suitable for industrial process control applications.

Table 2. Performance Indices of various control strategies.

Conditions	Control tactics	Number of Training Samples	IAE		computational time (Seconds)	Sampling period (Seconds)
			Top product	Bottom product		
No Disturbance	SVR-PSO-CREV	100	0.0825	0.0177	31.06	0.4141
	NN-PSO-CREV	1000	0.1337	0.1517	65.51	0.874
Disturbance	SVM-PSO-CREV	100	0.1011	0.0199	32.07	0.4276
	NN-PSO-CREV	1000	0.2341	0.1194	67.20	0.896

5. CONCLUSIONS

A viable solution to the problem of nonlinear model predictive control is proposed in this paper. A deterministic sparse kernel learning technique, SVM is used to create an accurate for prediction model and a derivative free optimization method, PSO-CREV is used to achieve faster convergence. Based on the simulation results of highly nonlinear distillation column process, the tracking performance of SVM- PSO-CREV based MPC is better than NN-PSO-CREV 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 better prediction accuracy, better generalization capability and sparse nature of SVM model and fast accurate convergence of PSO-CREV algorithm.

REFERENCES

Bhat, N., and Mcavoy, T. J. (1990). Use of neural nets for dynamic modeling and control of chemical process systems. *Comput. Chem. Eng*., 14(5), 573-583.

Bishop, C.M. (2006). *Pattern Recognition and Machine Learning.* Springer-Verlag, New York .

Chen, X., and Li, Y. (2007). Neural network predictive control for Mobile Robot Using PSO with Controllable Random Exploration Velocity. *International journal of intelligent control and systems*, 12(3), 217-229.

Chen, X., and Li, Y.(2007). A Modified PSO Structure Resulting in High Exploration Ability With Convergence Guaranteed. *IEEE Transactions on systems, man and cybernetics*, 37(5), 1271-1289.

De Brabanter, K., Karsmakers, P., Ojeda, F., Alzate, C., De Brabanter, J., Pelckmans, K., De Moor, B., Vandewalle, J., Suykens, J.A.K. (2011). *LS-SVM lab Toolbox User's Guide version 1.8.*

Eberhart, R.C., and Kennedy, J. (1995). A New Optimizer Using Particle Swarm Theory. *In Proc of 6th International Symposium on Micro Machine and Human Science*, 39-43, Nagoya, Japan.

Henson, M.A. (1998). Nonlinear Model predictive control: current status and future directions. *Computers and chemical engineering*, 23,187-202.

Hunt, k. J., Sbarbaro, k., Zbikowski, R., and Gawthrop, P. J. (1992). Neural Network for control systems – a survey. *Automatica,* 28(6), 1083-1120.

Kennedy, J., and Eberhart, R.C.(1995).Particle Swarm Optimization. *In Proc of IEEE International Conference on Neural Network*, 1995. 1942-1948, Perth, Australia.

Karatzoglou, A., and Meyer, D. (2006). Support Vector machine in R. *Journal of statistical software*, 15(9), 1-32.

Kulkarni, A., Jayaraman, V. K., and Kulkarni, B. D. Control of chaotic dynamical systems using support vector machines. *Phys. Lett. A*, 317, 429–435.

Liu, Y; Gao, Y., Gao , Z., Wang, H., and Li P. (2010). Simple nonlinear predictive control strategy for chemical processes using sparse kernel learning with polynomial form. *Industrial & Engineering Chemistry Research*, 49, 8209–8218.

Messerschmidt, L., and Engelbrecht, A.P.(2004). Learning to Play Games Using a PSO-Based Competitive Learning Approach. *IEEE Transactions on Evolutionary Computation*, 8(3), 280-288.

Psichogios, D. C., and Ungar, L. H.(1991). Direct and indirect model based control using artificial neural network. *Industrial & Engineering Chemistry Research*, 30(12), 2564-2573.

Qin, S.J. and Badgwell, A.T. (2003).A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7), 733-764.

Rossiter, J.A. (2003). *Model Based predictive control: a practical approach.* CRC presses Baco Paton London, New York, Washington.

Skogestad, S., and Morari, M. (1988).Understanding the Dynamic Behaviour of Distillation Columns. *Industrial Engineering Chemistry Research*, 27, 1848-1862.

Smola A. J., and Scholkopf, B. (2004). A Tutorial on support vector regression. Statistics and computing, 14(3), 199-222.

Taylor, J.S. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, UK.

Vapnik, V. (1998). *Statistical Learning Theory*, Wiley, New York.

Yoshida, H., Kawata, K., Fukuyama, Y., and Nakanishi, Y.A. (1999). Particle Swarm Optimization for Reactive Power and Voltage Control Considering Voltage Stability. *In Proc of International Conference on Intelligent System Application to Power Systems*, 117-121, Rio de Janeiro, Brazil.

Yue-hua, C., Guang-yi, C., and Xin-jian Z.(2007). LS-SVM model based nonlinear predictive control for MCFC system. *Journal of Zhejiang University Science A*, 8(5), 748-754.

Zhang, H., and Wang, X. (2006). Nonlinear systems modeling and control using support vector machine technique. *Lecture Notes in Computer Science Springer-Verlag:* New York, 3967, 660 – 669.

Zhong, W., Pi, D., and Sun, Y. (2005). An approach of nonlinear model multistep ahead predictive control based on SVM. *Lecture notes in Computer Science Springer-Verlag Berlin Heidelberg*, 3516, 1036-1039.

Zhong, W., Pi, D., and Sun, Y. (2005). Support vector machine based nonlinear model multistep ahead optimizing predictive control. *Journal of Cent. South university of technology,* 12(5), 519-595.