Nonlinear Model Predictive Control of MIMO System with Relevance Vector Machines and Particle Swarm Optimization

M. Germin Nisha, G. N. Pillai

Department of Electrical Engineering, Indian Institute of Technology Roorkee, Roorkee-247667, Uttarakhand, India
E-mail: germinnisha@gmail.com; gopinath.pillai@gmail.com

Abstract: This paper demonstrates control accuracy and computational efficiency of nonlinear model predictive control (NMPC) strategy which utilizes a probabilistic sparse kernel learning technique called Relevance vector regression (RVR) and particle swarm optimization with controllable random exploration velocity (PSO-CREV). An accurate reliable nonlinear model is first identified by RVR 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 a deterministic sparse kernel learning technique called Least squares support vector machines (LS-SVM) regression 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: Relevance vector regression; 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).

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). 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 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, 2000) introduced relevance vector machine (RVM) 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 (SVM). 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 (Vapnik, 1998; Tipping, 2001). 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 (Tipping, 2000) demonstrates the comparable generalization performance of RVM than SVM with intensely fewer kernel functions.

Nonlinear system identification using RVM is successfully discussed in many literatures (Gustavo *et al.*, 2007; Psorakis *et al.*, 2010; Mihalis *et al.*, 2012; Wong *et al.*, 2012) which highlights its significance. To the author's best knowledge, combinations of RVM model and MPC approach are less reported in literatures. The complexity of developing an accurate model for the distillation column and the nonlinearities of its dynamics, make very attractive the use of relevance vector machine.

Despite of accurate approximation of nonlinear dynamics by RVR model it suffers from computational burden as model predictive controller does prediction and optimization at each sampling instant. 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 relevance 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 RVM based MPC when compared to LS-SVM based MPC.

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

2. RELEVANCE VECTOR REGRESSION

RVM is a probabilistic model whose functional form is equivalent to that of SVM. It achieves comparable recognition accuracy to the SVM, yet provides a full predictive distribution, and also requires substantially fewer kernel functions (Bishop and Tipping, 2000). RVM is based on Bayesians approach in which a prior is introduced over the model weights and each weight is administrated by one hyperparameter. The most probable value of each hyper parameter is iteratively evaluated from the data. The model is sparser since the posterior distributions of some proportion of the weights are set to 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 expressed as a linear combination of the response of a set of M basis functions, as follows:

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

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

$$\hat{y}(x,w) = \sum_{m=1}^{M} w_m \varphi_m(x) = w^T \varphi \quad \text{where } w \in Z \tag{2}$$

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 $w = [w_1, ...w_M]^T$ and the responses of all kernel function $\varphi(x) = [\varphi_1(x) ... \varphi_M(x)]^T$ which maps the input data into a high dimensional feature space z.

Hence the obtained error signal could be stated as

$$\varepsilon_m = y_m - \hat{y}_m = N(0, \sigma^2) \tag{3}$$

The objective of relevance vector regression is to find the finest value of w such that $\hat{y}(x,w)$ makes good predictions for unknown input data. For the RVM model in equation (2) let $\alpha = [\alpha_1 .....\alpha_M]^T$ 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,

$$p(w_m / \alpha_m) = \prod_{m=1}^{M} (\alpha_m / 2\pi)^{1/2} \exp\{-\alpha_m {w_m}^2 / 2\} \tag{4}$$

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,

$$p(y / w, \sigma^2) = \prod_{m=1}^{M} p(y_m / w, \sigma^2) = \frac{e^{-\left\|(y-\hat{y})^2\right\| / 2\sigma^2}}{\sqrt{(2\pi\sigma^2)^M}} \tag{5}$$

The above likelihood function is enhanced by the prior in equation (4) defined over each weight to reduce the complexity of the model and to avoid over fitting.

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

$$p(w/y,\alpha,\sigma^2)=\frac{p(y/w,\sigma^2)p(w/\alpha)}{p(y/\alpha,\sigma^2)} \quad (6)$$

The posterior distribution in equation (6) is a Gaussian distribution function,

$$p(w/y,\alpha,\sigma^2)=N(\mu,\sigma^2) \quad (7)$$

Whose covariance and mean are respectively given by

$$\Sigma=(\sigma^{-2}\varphi^T\varphi+A)^{-1} \quad (8)$$

$$\mu=\sigma^{-2}\Sigma\varphi^T y \quad (9)$$

with $A=diag(\alpha)$

Marginalization of the likelihood distribution over the training targets given by equation (5) can be obtained by integrating out the weights to acquire the marginal likelihood for the hyperparameters.

$$p(y/\alpha,\sigma^2)=\int p(y/w,\sigma^2)p(w/\alpha)dw=N(0,C) \quad (10)$$

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

In equation (8) and equation (9) the only unknown variables are the hyperparameters α. The values of these hyperparameters are estimated using the framework of type II maximum likelihood (Berger, 2010).

$$\log p(y/\alpha,\sigma^2)=-1/2(M\log 2\pi+\log|C|+y^T C^{-1}y) \quad (11)$$

Logarithm is included in equation (11) to reduce computational complexity. Maximization of the logarithmic marginal likelihood in equation (11) over α leads to the most probable value α_{MP} which provides the maximum a posteriori (MAP) estimate of the weights.

The ambiguity about the optimal value of the weights, given by (6), is used to express ambiguity about the predictions made by the model, i.e., given an input x^*, the probability distribution of the corresponding output y^* is given by the predictive distribution

$$p(y^*/x^*,\hat{\alpha},\hat{\sigma}^2)=\int p(y^*/x^*,w,\hat{\sigma}^2)p(w/y,\hat{\alpha},\hat{\sigma}^2)dw \quad (12)$$

which has the Gaussian form

$$p(y^*/x^*,\hat{\alpha},\hat{\sigma}^2)=N(Y^*,\sigma^{*2}) \quad (13)$$

The mean and variance of the predicted model are respectively,

$$Y^*=\varphi^T(x^*)\mu \quad \text{and} \quad \sigma^{*2}=\hat{\sigma}^2+\varphi^T(x^*)\Sigma\varphi(x^*) \quad (14)$$

Maximizing the logarithmic marginal likelihood in (11) leads the optimal values of many of the hyperparameters α_m typically infinite yielding a posterior distribution in (6) 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. MPC BASED ON RVM AND PARTICLE SWARM OPTIMIZATION

3.1 RVM Based MPC Principle

The basic structure of RVM 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 RVM model of the actual plant to be controlled with predicted output $\hat{y}(k)=[\hat{y}(k+1)/k\ ,\ldots\ \hat{y}(k+Np)/k]$ 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 $ref(k)=[ref_1(k)\ \ldots ref_{Np}(k)]$. Here *k* stands for the current sampling instant.

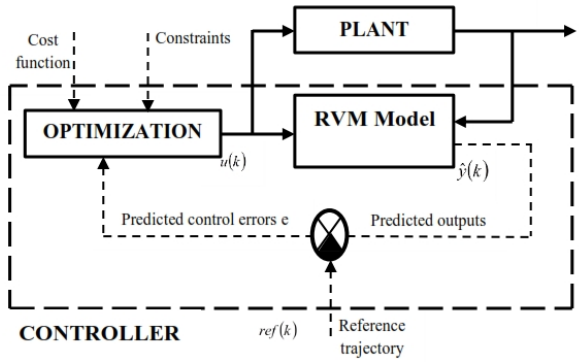

Fig. 1. Basic structure of RVM based nonlinear model Predictive control.

Thus at each sampling instant a sequence of manipulated variable *u(k)* is calculated in order to minimize the formulated performance index 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 LS-SVM based nonlinear model predictive control is obtained by replacing RVM model by LS-SVM model in Fig. 1.

3.2 Performance index formulation

For a MIMO $n \times m$ nonlinear process the predicted outputs of RVM model is a function of past process outputs, $Y(k)=[y_1(k)\ldots..y_1(k-n_y+1),y_2(k)\ldots..y_2(k-n_y+1),\ldots.$

$,y_m(k).....y_m(k-n_y+1)]$ and past process inputs, $U(k-1)=[u_1(k-1)...u_1(k-n_u+1),\ u_2(k-1)...u_2(k-n_u+1),.... ,u_n(k-1)...u_n(k-n_u+1)]$. Which could be compactly rewritten as $Y(k)=[Y_1(k),Y_2(k)...,Y_m(k)]$ *and* $U(k-1) = [U_1(k-1),U_2(k-1)....,U_n(k-1)]$. 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,

$$\begin{aligned}
&\hat{y}_1(k+1) = f[Y_1(k),u(k),U(k-1)] \\
&\hat{y}_2(k+1) = f[Y_2(k),u(k),U(k-1)] \\
&\text{……..} \\
&\hat{y}_m(k+1) = f[Y_m(k),u(k),U(k-1)]
\end{aligned} \tag{15}$$

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

$$\hat{y}_j(k+1) = \sum_{m=1}^{M} w_m \varphi_m(x) = w^T \varphi \tag{16}$$

where $j=1...m$ 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,

$$J = \sum_{j=1}^{m}\sum_{i=N_1}^{N_2} q_j \left[ref_j(k+i) - \hat{y}_j(k+i)\right]^2 + \sum_{j=1}^{n}\sum_{i=1}^{N_u} \lambda_j \left[\Delta u_j(n+i)\right]^2 \tag{17}$$

In the performance index formulated in Equation (17) $\hat{y}$ 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.

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 RVM 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.

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 (18) and Equation (19).

$$v_{id}^{(t+1)} = \omega v_{id}^{(t)} + C_1 r_1 (P_{id}^{(t)} - X_{id}^{(t)}) + c_2 r_2 (P_{gd}^{(t)} - X_{id}^{(t)}) \tag{18}$$

$$x_{id}^{(t+1)} = x_{id}^{(t)} + v_{id}^{(t+1)} \tag{19}$$

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.

Learning factors c_1 and c_2 usually equals to 2. However, other settings were also used. But usually c_1 equals to c_2 and ranges from [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 behavior 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 (20) and Equation (21) respectively.

$$v_{id}^{(t+1)} = \varepsilon^{(t)}[\omega v_{id}^{(t)} + c_1 r_1 (P_{id}^{(t)} - X_{id}^{(t)}) + c_2 r_2 (P_{gd}^{(t)} - X_{id}^{(t)}) + \xi_{id}^{(t)}] \quad (20)$$

$$x_{id}^{(t+1)} = \alpha x_{id}^{(t)} + v_{id}^{(t+1)} + \frac{1-\alpha}{\phi_{id}^{(t)}}[c_1 r_1 P_{id}^{(t)} + c_2 r_2 P_{gd}^{(t)}] \quad (21)$$

$\xi_{id}^{(t)}$ - 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.

4. APPLICATION ON 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. 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.

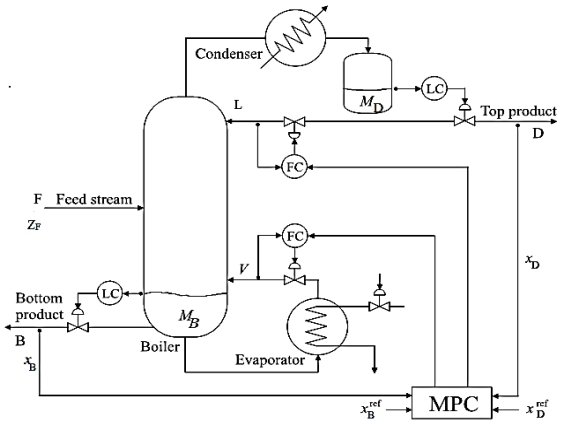

Fig. 2. Schematic of the binary distillation column process.

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;

$$i = 2, N \ (i \neq N_F, \ i \neq N_F + 1)$$

$$M_i \dot{x}_i = L_{i+1} x_{i+1} + V_{i-1} y_{i-1} - L_i x_i - V_i y_i \quad (22)$$

above feed location $i = N_F + 1$

$$M_i \dot{x}_i = L_{i+1} x_{i+1} + V_{i-1} y_{i-1} - L_i x_i - V_i y_i + F_v y_F \quad (23)$$

below feed location, $i = N_F$

$$M_i \dot{x}_i = L_{i+1} x_{i+1} + V_{i-1} y_{i-1} - L_i x_i - V_i y_i + F_L x_F \quad (24)$$

reboiler, $i = 1$

$$M_B \dot{x}_i = L_{i+1} x_{i+1} - V_i y_i - B x_i, \qquad x_B = x_1 \quad (25)$$

total condenser, $i = N + 1$

$$M_D \dot{x}_i = V_{i-1} x_{i-1} - L_i x_i - D x_i, \qquad y_D = x_{N+1} \quad (26)$$

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

$$y_i = \alpha\, x_i / (1 + (\alpha - 1)\, x_i) \tag{27}$$

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

$$i > N_F \text{ above feed, } L_i = L,\ V_i = V + F_V \tag{28}$$

$$i \le N_F \text{ below feed, } L_i = L + F_L,\ V_i = V \tag{29}$$

$$F_L = q_F F \quad , F_V = F - F_L \tag{30}$$

condenser holdup is kept constant,

$$D = V_N - L = V + F_V - L \tag{31}$$

reboiler holdup is kept constant,

$$B = L_2 - V_1 = L + F_L - V \tag{32}$$

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

$$FZ_F = F_L x_F + F_V y_F \tag{33}$$

$$y_F = \alpha\, x_F / (1 + (\alpha - 1)\, x_F) \tag{34}$$

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 $2.5 \le u_1(t) \le 2.9$ and $3 \le u_2 t) \le 3.5$ 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 SVR model, two past outputs and past inputs are sufficient hence the following second order model is chosen.

$$y_1(k) = f(y_1(k-1), y_1(k-2), u_1(k-1), u_1(k-2), u_2(k-1), u_2(k-2)) \tag{35}$$

$$y_2(k) = f(y_2(k-1), y_2(k-2), u_1(k-1), u_1(k-2), u_2(k-1), u_2(k-2)) \tag{36}$$

A sequence of 100 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 (Tipping, 2001). For this RVR model RBF kernel is used with the width parameter estimated automatically by the learning procedure (Tipping, 2001) this 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.

The SVR model is also trained offline using a sequence of 100 samples 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). The identification performance of RVR model and SVR model are assessed by the root mean square error (RMSE) performance function.

$$RMSE = \left\{ \sum_{k=1}^{N} [\hat{y}(k) - y(k)]^2 \Big/ N \right\}^{1/2} \tag{37}$$

where $\hat{y}(k)$ represents the 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. Fig. 3 and Fig. 4 correspond to the modeling results of RVR and SVR methods. While modelling the training set, RVR model can attain a slightly better identification performance than SVR. Similarly, for the test data which are beyond the training data, the RVR model can achieve better performance than SVR. The comparative graph of prediction errors of RVR model and SVR model for test data are shown in Fig. 4, which explores the better extrapolation capability of RVR model than SVR model.

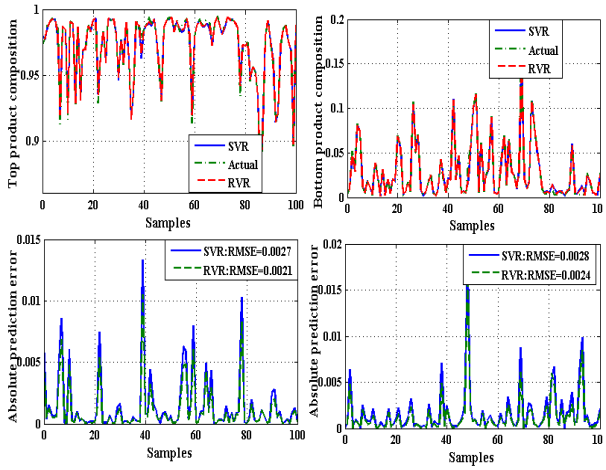

Fig. 3. Training performance comparison of RVR and SVR models.

Accuracy of the model in terms of RMSE (37) is tabulated in Table 1.Thus one can conclude that the RVR based empirical modeling can accomplish better accuracy and extrapolation capability than LS-SVM based modeling.

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

Model	$RMSE_{training}$		$RMSE_{testing}$	
	x_D	x_B	x_D	x_B
RVR	0.0021	0.0024	0.0023	0.0025
SVR	0.0027	0.0028	0.0028	0.0030

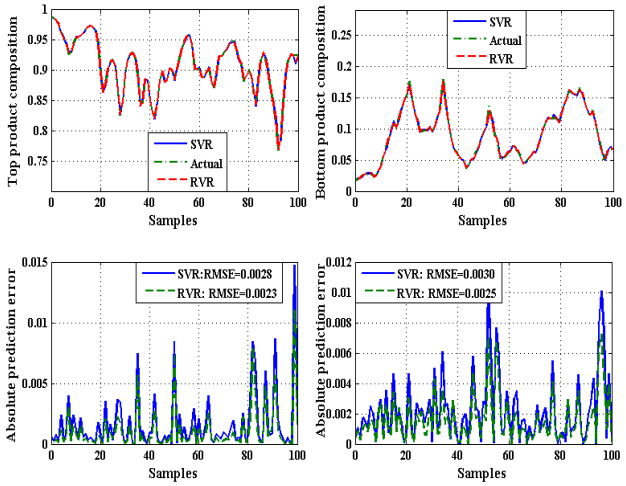

Fig. 4. Testing performance comparison of RVR and SVR model.

The offline trained and validated RVR model or LS-SVM model is then used as the nonlinear model for nonlinear MPC. Fig. 5 illustrates the random set point tracking performances of RVR based MPC and SVR based MPC. The tracking performance of RVR based MPC is better when compared with LS-SVM 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 boil up 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.

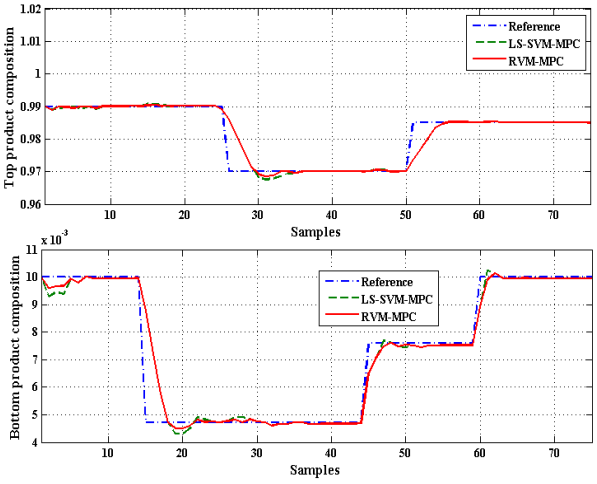

Fig. 5. Set point tracking performance of distillation column process by RVM-MPC and LS-SVM-MPC.

The unmeasured disturbance rejection capability of RVM-PSO-CREV based MPC and LS-SVM-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.

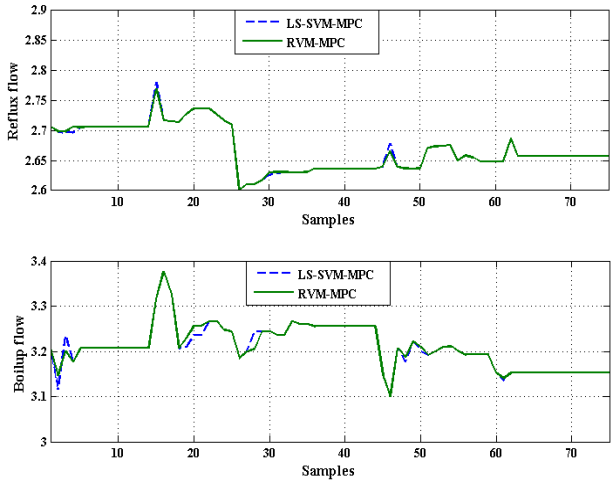

Fig. 6. Changes in the process variables for tracking the top product and bottom product compositions of distillation column process.

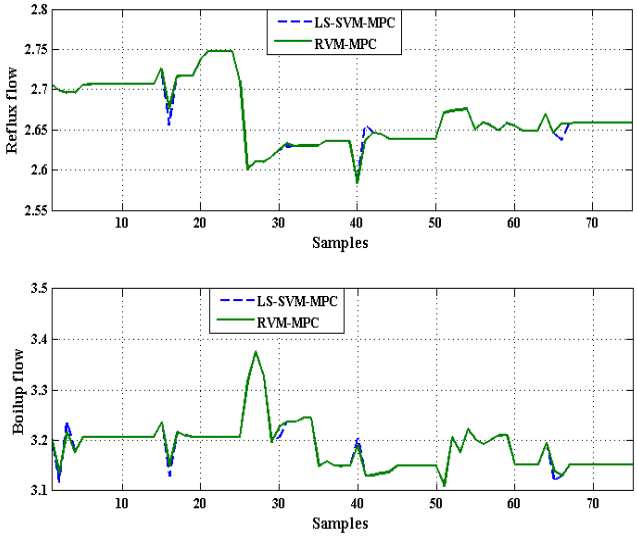

Fig. 7. Changes in the process variable to show unmeasured disturbance.

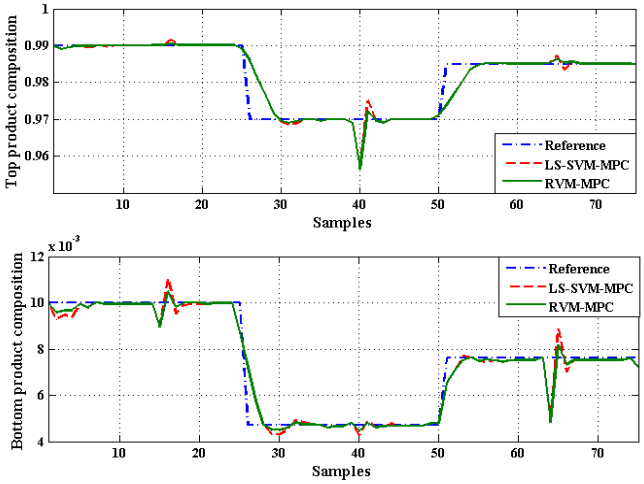

Fig. 8. Performance comparison of unmeasured disturbance rejection.

Certainly the unmeasured disturbance rejection performance of RVR-PSO-CREV based MPC is better when compared to LS-SVM-PSO-CREV based MPC as shown in Fig. 8.
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.

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 RVR model sharply reduces the computational time of RVR-MPC to 31.06 seconds for 75 samples (ie., nearly 0.414 Seconds for sample), which is much shorter than the sampling time of the distillation column process. Hence, it is clear that LS-SVM based MPC is the one which consumes more time with more integral average error and RVR-PSO-CREV model predictive controller is the better controller with less computational load and less integral average error.

Thus when compared to LS-SVM-PSO-CREV based MPC , RVM-PSO-CREV based MPC is the best controller based on various attributes like better prediction accuracy, better generalization capability, better set point tracking performance, better unmeasured disturbance rejection capability with very less computation time due to its sparse model. 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**	**Integral Absolute Error**		**Number of Support Vectors/ Relevance Vectors**		**computational time (Seconds)**
			Top product	**Bottom product**	**Model1**	**Model2**	
No Disturbance	RVM-PSO-CREV	100	0.0749	0.0157	12	11	16.80
	LS-SVM-PSO-CREV	100	0.0825	0.0177	61	59	31.06
Disturbance	RVM-PSO-CREV	100	0.0922	0.0158	12	11	16.94
	LS-SVM-PSO-CREV	100	0.1010	0.0199	61	59	32.07

5. CONCLUSIONS

A viable solution to the problem of nonlinear model predictive control is proposed in this paper. A probabilistic sparse kernel learning technique, RVM 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 RVM- PSO-CREV based MPC is better than LS-SVM-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 more sparseness property of RVM model and fast accurate convergence of PSO-CREV algorithm.

REFERENCES

Berger, J.O. (2010). *Statistical Decision Theory and Bayesian Analysis*. Second ed. Springer-Verlag, New York.

Bishop, C.M., Tipping, M.E. (2000). Variational Relevance Vector Machines. *In Proc of 16th conference on Uncertainity in Artificial Intelligence*, 46-53.

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*, 1995. 39-43, Nagoya, Japan.

Gustavo, C.V., Manel, M.R., Jose Luis, R.A., and Jordi M.M. (2007). Nonlinear System Identification with Composite Relevance Vector Machines. *IEEE Signal Processing Letters*, 14(4), 279-282.

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

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

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.

Mihalis, A., Nicolaou, Gunes, H., and Pantic, M.(2012). Output-associative RVM regression for dimensional and continuous emotion prediction. *Image and Vision Computing, article in press*, 30(3),186-196.

Psorakis, I., Damoulas, T., and Girolami, A.M. (2010). Multiclass Relevance Vector Machines: Sparsity and Accuracy. *IEEE Transactions on neural networks,* 21(10), 1588-1598.

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 Behavior of Distillation Columns. *Industrial Engineering Chemistry Research*, 27, 1848-1862.

Tipping, M.E. (2000). The relevance vector machine. *Advances in Neural Information Processing Systems*, 12, 652–658.

Tipping, M.E. (2001). Sparse Bayesian Learning and the Relevance Vector Machine. *Journal of Machine Learning Research*, 1, 211–244.

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.

Wong, P., Xu, Q., Vong, C., and Wong, H. (2012). Rate-Dependent Hysteresis Modeling and Control of a Piezostage Using Online Support Vector Machine and Relevance Vector Machine. *IEEE Transactions on industrial Electronics*, 59(4), 1988-2001.

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.