Neural network and support vector machine predictive control of tert-amyl methyl ether reactive distillation column

Neha Sharma and Kailash Singh

Department of Chemical Engineering, Malaviya National Institute of Technology, JLN Marg, Malaviya Nagar, Jaipur 302017, India

(Received 20 November 2013; final version received 10 May 2014)

An algorithm of model predictive control based on artificial neural network and least-square support vector machine method is presented for a class of industrial process with strong nonlinearity such as tert-amyl methyl ether (TAME). Integral constant is added to improve the performance of the controller. In the present work, two different control methodologies neural network predictive control (NNPC) and support vector machine-based predictive control (SVMPC) are implemented and compared with a conventional proportional-integral-derivative (PID) control methodology to a TAME reactive distillation column. The simulation result shows that both NNPC and SVMPC gives better control performance than PID for set-point change as well as for load change of ±10% in methanol feed flow rate and molar ratio of methanol to isoamylene in reactor effluent feed.

Keywords: reactive distillation column; neural network predictive control; SVMPC; TAME

1. Introduction

Model predictive control (MPC) has become one of the most successful control algorithms in process industries such as chemical plants and oil refineries (Garcia & Morshedi, 1986; Qin & Badgwell, 1997, 2003; Rewagad & Kiss, 2011; Tatjewski & Lawrynczuk, 2006). Predictive model is a basic element of MPC. Most of the MPC algorithms are based on a linear model of the processes. The major drawback associated with these linear controllers is that they do not perform well over the wide range of operating conditions and with large disturbances. Since most industrial processes exhibit severe nonlinearity, the studies on the use of artificial intelligence techniques such as artificial neural network (ANN), support vector machine (SVM) and fuzzy logic have drawn increasing attention in recent years because of their ability to represent nonlinear systems and their self-learning capabilities.

tert-amyl methyl ether (TAME) is a widely studied model system to understand the complex behavior of reactive distillation column (Katariya, Kamath, Moudgalya, & Mahajani, 2008). But very limited research has been found in the area of the control, due to the complex dynamics of the system. Al-Arfaj and Luyben (2004) studied the plant-wide control of the TAME process using the proportional-integral (PI) control methodology. In the present work, we studied control of reactive distillation column using artificial intelligence-based control methodology and compare these techniques with the conventional control methodology (PI).

Synthesis of multiple reaction system TAME exhibits highly nonlinear behavior, i.e. steady-state multiplicity, strong interactions between process variables, etc. (Katariya, Moudgalya, & Mahajani, 2006; Mohl et al., 1997, 1999; Sharma & Singh, 2010). This complex dynamics makes process control of the reactive distillation column a challenging task. The need to handle such difficult control problem has led to use ANN and SVM in MPC and has recently attracted a great deal of attention. The attractive advantage of the neural network approach is that an accurate representation of the process can be obtained by training the network. Neural networks are capable of handling complex and nonlinear problems and can reduce the engineering efforts required in controller model development. In the field of chemical engineering, these have been successfully implemented in distillation control, for example. The papers by ZareNezhad and Aminian (2011), Hui, Hui, Aziz, & Ahmad (2011), Lawrynczuk (2010), Konakom, Kittisupakorn, Saengchan, and Mujtaba (2010), Arumugasamy and Ahmad (2009), Ahmad and Mat Noor (2009), Hussain (1999), Ramachandran and Russell Rhinehart (1995), Thibault and Grandjean (1991) and Åström and McAvoy (1992), Bhat and McAvoy (1990) provide in-depth reviews on neural network application in chemical process control.

Least-square support vector machine (LS-SVM) has been proposed by Syukens and Vandewalle (1999) and has been successfully applied to many applications.
Artificial Intelligence (AI) techniques are able to deal with nonlinear problems, and once trained can perform prediction and generalization at high speed (Kalogirou, 2003). They are widely used in system modeling. In this paper, we have used ANN and LS-SVM as a predictive model in MPC, namely neural network predictive control (NNPC) and support vector machine predictive control (SVMPC). MATLAB® was used to implement NNPC, SVMPC and proportional-integral-derivative (PID) in the TAME reactive distillation column. The paper is organized as follows: in Section 2, we briefly describe the synthesis of the process TAME, Section 3 introduces the ANN and LS-SVM as a model and describes the steps for model development and its statistical analysis. Section 4 describes the design methodology for artificial intelligence-based model predictive controller, namely NNPC and SVMPC. Section 5 gives optimum values of parameters of NNPC and SVMPC. Section 6 compares both control methodologies for disturbance rejection and set-point tracking which is justified by several performance criteria such as integral of time absolute error (ITAE), integral of squared error (ISE), integrated absolute error (IAE) and integral of time squared error (ITSE).

2. Process description

TAME is one of most possible antiknock additives to gasoline. It is added both to enhance octane number to replace banned tetraethyl lead and to raise the oxygen content in gasoline. The largest volume component in the past was MTBE but it is being phased out because of groundwater contamination problems. The TAME reactive distillation is an etherification process that is similar to ETBE. Therefore, TAME is becoming more important. TAME is formed by reaction of isomylenes (IAs) (2M1B and 2M2B), which is coming from C5-stream of the refinery, with methanol (MeOH) in the presence of inert components (i.e. isopentane). Three reactions take place simultaneously in TAME synthesis: etherification of the two methylbutenes and their isomerization. The TAME reactions have been shown to be reversible and fairly exothermic. Besides TAME formation, several side reactions such as isomerization of reactive amylene and hydration of IAs to tert-amyl alcohol. also take place, among which the isomerization reaction between the two IAs is the most important.

\[
\begin{align*}
\text{MeOH} + \text{2M1B} & \leftrightarrow \text{TAME} \quad R_1 \\
= M_{\text{cat}}(k_{F1}x_{2M1B}\text{MeOH} - k_{B1}x_{\text{TAME}}). \\
\text{MeOH} + \text{2M2B} & \leftrightarrow \text{TAME} \quad R_2 \\
= M_{\text{cat}}(k_{F2}x_{2M2B}\text{MeOH} - k_{B2}x_{\text{TAME}}). \\
\text{2M1B} & \leftrightarrow \text{2M2B} \quad R_3 \\
= M_{\text{cat}}(k_{F3}x_{2M1B} - k_{B3}x_{2M2B}).
\end{align*}
\]

The chemical reaction kinetic model is adopted from Luyben and Yu (2008): 

\[
\begin{align*}
k_{F1} & = 1.19367 \times 10^{14} \exp \left( \frac{-76103.737}{RT} \right), \\
k_{B1} & = 2.118 \times 10^{17} \exp \left( \frac{-110540.899}{RT} \right), \\
k_{F2} & = 1.23462 \times 10^{17} \exp \left( \frac{-98230.2176}{RT} \right), \\
k_{B2} & = 1.38726 \times 10^{20} \exp \left( \frac{-124993.965}{RT} \right), \\
k_{F3} & = 2044683 \times 10^{16} \exp \left( \frac{-965122.6384}{RT} \right), \\
k_{B3} & = 3.86397 \times 10^{16} \exp \left( \frac{-104196.053}{RT} \right).
\end{align*}
\]

where the kinetic constants $k$’s are in mol/L-s, $R$ in J/mol-K and $T$ in K. The TAME synthesis process flow sheet using reactive distillation in MATLAB® is presented in Figure 1. An equilibrium stage dynamic model for the synthesis of TAME of the RDC has been developed and described in our previous paper (Sharma & Singh, 2012). The column has a total condenser and reboiler. The theoretical stages are numbered from top to bottom. For this purpose, continuous mode of operation has been assumed. The equations for 15 stages for the system and five-component system were solved in MATLAB® by ODE15S solver. TAME is the heaviest component, which leaves the column in the

Figure 1. Synthesis of TAME in reactive distillation column.
Table 1. Column specification.

| Parameter                | Value                  |
|--------------------------|------------------------|
| Feed flow rates:         |                        |
| Fresh methanol           | 65.3 mol/s             |
| Reactor effluent         | 341.2 mol/s            |
| Pressure                 | 4 bar                  |
| Reflux ratio             | 4                      |
| Number of stages (N)     | 15                     |
| Reactive zone            | Tray number 5–10       |
| Feed stage location:     |                        |
| Fresh methanol           | Tray number 5          |
| Reactor effluent         | Tray number 10         |
| Volume of each tray      | 1220 l                 |
| Initial volume of reboiler | 12201                 |
| Reboiler heat duty       | 15.727 MW              |

3. Methodologies used for model development

3.1. Artificial neural network

Steps for the neural network model are as follows:

(i) Data generation: At first, the fundamental model was used as the real process; it was simulated open-loop in order to obtain two sets of random number data, namely training and test sets. Training data sets contained 5000 samples and test data set contained 2001 samples with 100 s sampling interval. These data sets were used to generate an input matrix consisting of an output variable and manipulated variable.

(ii) Configuration of neural network: Feed-forward back propagation network was created, which consists of number of layers using the weight function, net input function and the specified transfer functions. The first layer has weights coming from the input. Each subsequent layer has a weight coming from the previous layer. The last layer is the network output. Network contains two bias vectors (values given in Table 2), one input weight matrix and one layer weight matrix (values of weight matrices are

Table 2. Values of bias vectors.

| Bias vectors | Hidden layer | Output layer |
|--------------|--------------|--------------|
| −0.1781      |              | −0.3801      |
| 0.2255       |              |              |
| −0.2014      |              |              |
| 0.2655       |              |              |
| 0.4394       |              |              |
| −0.4617      |              |              |
| −0.4475      |              |              |
| 0.1295       |              |              |
| 0.0435       |              |              |
| 0.4256       |              |              |
| −0.6434      |              |              |
| 1.5630       |              |              |
| −0.9224      |              |              |
| 0.0108       |              |              |
| −0.4806      |              |              |
| 0.0519       |              |              |
| 0.1085       |              |              |
| 0.6328       |              |              |
| 0.0869       |              |              |
| −0.2992      |              |              |
| −0.2281      |              |              |
| 1.5911       |              |              |
| 0.0070       |              |              |
| −0.8065      |              |              |
| −0.0724      |              |              |
| −0.3683      |              |              |
| −0.3659      |              |              |
| 0.0021       |              |              |
| −0.2727      |              |              |
| 0.2461       |              |              |

Figure 2. Effect of reflux ratio on TAME purity in bottoms.
Table 3. Weight matrices of neural network.

| Neuron in hidden layer | $u_{k-1}$ | $u_{k-2}$ | $u_{k-3}$ | $u_{k-4}$ | $u_{k-5}$ | $u_{k-6}$ | $u_{k-7}$ | $u_{k-8}$ | $u_{k-9}$ | $u_{k-10}$ |
|------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Input-hidden layer weight matrix |           |           |           |           |           |           |           |           |           |           |
| 1 Neuron               | −1.456    | 1.032     | 0.407     | −0.045    | 0.033     | 0.438     | −0.429    | −0.435    | −0.22     | −0.023    |
| 2 Neuron               | 0.626     | −0.477    | 0.022     | −0.427    | −0.429    | −0.445    | 0.125     | 0.104     | 0.465     | 0.459     |
| 3 Neuron               | 0.066     | 0.253     | 0.002     | 0.164     | 0.479     | 0.492     | 0.222     | 0.148     | −0.02     | −0.199    |
| 4 Neuron               | −0.698    | −0.177    | 0.127     | 0.217     | 0.007     | 0.038     | −0.277    | −0.768    | −0.334    | −0.577    |
| 5 Neuron               | −0.466    | 0.05      | −0.006    | 0.083     | −0.092    | −0.381    | 0.456     | 0.176     | 0.278     | −0.221    |
| 6 Neuron               | −0.688    | 0.544     | −0.234    | −0.231    | −0.213    | −0.032    | −0.621    | 0.025     | 0.016     | −0.226    |
| 7 Neuron               | 0.517     | −0.269    | −0.086    | −0.311    | 0.79      | 0.493     | −0.16     | 0.201     | −0.098    | 0.044     |
| 8 Neuron               | 0.241     | −0.065    | 0.185     | −0.492    | −0.327    | −0.28     | −0.37     | −0.272    | −0.559    | −0.357    |
| 9 Neuron               | 0.689     | −0.338    | 0.307     | −0.352    | 0.282     | 0.407     | 0.15      | −0.183    | 0.429     | −0.059    |
| 10 Neuron              | 0.155     | −0.141    | 0.076     | 0.693     | −0.195    | 0.479     | 0.291     | 0.105     | 0.076     | −0.179    |
| 11 Neuron              | −1.163    | 0.401     | 0.452     | 0.28      | 0.361     | 0.11      | −0.317    | −0.014    | −0.517    | −0.562    |
| 12 Neuron              | 1.247     | −0.492    | −0.184    | −0.245    | 0.164     | −0.977    | 0.345     | 0.01      | 0.372     | 0.231     |
| 13 Neuron              | −0.77     | 0.605     | −0.117    | −0.598    | −0.407    | −0.456    | 0.048     | −0.157    | −0.143    | −0.347    |
| 14 Neuron              | −0.306    | −0.767    | 0.259     | 0.023     | −0.011    | 0.244     | −0.02     | −0.17     | −0.184    | −0.141    |
| 15 Neuron              | −0.154    | 0.219     | −0.528    | −0.493    | −0.182    | 0.164     | −0.271    | 0.038     | −0.331    | −0.201    |
| 16 Neuron              | −0.233    | 0.485     | 0.179     | 0.779     | −0.896    | 0.498     | 0.352     | −0.418    | 0.412     | 0.02      |
| 17 Neuron              | 0.455     | 0.097     | 0.25      | −0.068    | 0.366     | −0.176    | 0.297     | 0.218     | 0.174     | −0.081    |
| 18 Neuron              | −0.691    | −0.2      | 0.114     | 0.351     | −0.236    | −0.044    | 0.288     | 0.136     | 0.015     | −0.53     |
| 19 Neuron              | −0.501    | 0.032     | 0.066     | 0.35      | 0.062     | 0.258     | 0.47      | 0.107     | −0.378    | 0.366     |
| 20 Neuron              | −1.347    | 0.988     | −0.177    | 0.019     | 0.263     | 0.711     | 0.206     | 0.053     | −0.42     | 0.112     |
| 21 Neuron              | −0.07     | −0.262    | 0.136     | 0.34      | 0.128     | 0.283     | −0.181    | −0.284    | −0.443    | −0.292    |
| 22 Neuron              | 1.621     | 0.029     | −0.327    | −0.384    | −0.033    | −0.434    | 0.228     | 0.238     | 0.357     | 0.254     |
| 23 Neuron              | 0.325     | −0.013    | −0.234    | −0.151    | 0.261     | 0.011     | −0.548    | −0.2      | −0.108    | 0.208     |
| 24 Neuron              | −0.836    | 0.491     | −0.3      | 0.04      | 0.319     | 0.577     | −0.137    | 0.174     | 0.005     | 0.099     |
| 25 Neuron              | −0.155    | 0.35      | −0.161    | 0.441     | 0.644     | 0.864     | −0.591    | −0.287    | −0.348    | −0.459    |
| 26 Neuron              | −0.628    | −0.031    | 0.358     | 0.263     | 0.195     | 0.177     | −0.244    | −0.364    | −0.407    | −0.279    |
| 27 Neuron              | −1.376    | 0.795     | 0.255     | −0.026    | −0.175    | 0.044     | −0.013    | −0.528    | −0.514    | 0.288     |
| 28 Neuron              | 0.213     | 0.394     | −0.264    | 0.057     | 0.141     | 0.145     | 0.279     | 0.708     | 0.562     | 0.434     |
| 29 Neuron              | −1.159    | 0.839     | −0.391    | −0.327    | −0.12     | −0.085    | 0.024     | 0.052     | −0.339    | −0.088    |
| 30 Neuron              | 0.457     | −0.152    | 0.276     | 0.268     | 0.13      | 0.833     | −0.627    | −0.126    | −0.064    | −0.387    |

Hidden layer–output weight matrix

−1.27  1.45  0.73  0.76  1.18  1.17  −0.92  0.85  0.11  1.14  0.67

(Continued).
| Neuron in hidden layer | \(y_{k-1}\) | \(y_{k-2}\) | \(y_{k-3}\) | \(y_{k-4}\) | \(y_{k-5}\) | \(y_{k-6}\) | \(y_{k-7}\) | \(y_{k-8}\) | \(y_{k-9}\) | \(y_{k-10}\) |
|------------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| **Input-hidden layer weight matrix** |
| 1 Neuron | 0.331 | -0.587 | -0.137 | -0.244 | -0.885 | 0.944 | 0.571 | -0.087 | -0.29 | -0.275 |
| 2 Neuron | -0.015 | 0.32 | 0.05 | 0.084 | 0.011 | 0.368 | 0.14 | -0.458 | -0.55 | 0.809 |
| 3 Neuron | 0.329 | 0.649 | -0.329 | 0.063 | -0.384 | -0.142 | 0.07 | 0.013 | -0.33 | 0.599 |
| 4 Neuron | 0.665 | -0.039 | -0.289 | 0.103 | -0.695 | 0.292 | 0.301 | 0.384 | 0.299 | -0.41 |
| 5 Neuron | -0.339 | -0.345 | 0.139 | -0.088 | -0.215 | -0.029 | 0.114 | -0.08 | 0.062 | 0.012 |
| 6 Neuron | 0.02 | 0.58 | 0.294 | -0.418 | 0.59 | -0.254 | 0.068 | -0.765 | -0.092 | -0.162 |
| 7 Neuron | 0.678 | 0.457 | 0.173 | -0.11 | -0.231 | -0.334 | 0.207 | 0.087 | -0.336 | 0.387 |
| 8 Neuron | -0.362 | -0.298 | 0.446 | 0.2 | 0.221 | -0.023 | 0.083 | 0.359 | 0.068 | -0.213 |
| 9 Neuron | 0.438 | -0.586 | 0.335 | -0.12 | 0.18 | -0.257 | 0.315 | 0.022 | 0.313 | 0.121 |
| 10 Neuron | -0.35 | -0.039 | 0.492 | 0.34 | -0.29 | -0.091 | 0.248 | 0.375 | -0.066 | 0.616 |
| 11 Neuron | 0.483 | -0.424 | 0.14 | -0.202 | 0.489 | -0.148 | -0.236 | 0.081 | 0.5 | -0.366 |
| 12 Neuron | -0.217 | 0.205 | -0.25 | -0.073 | 0.274 | -0.129 | 0.137 | 0.028 | -0.269 | -0.688 |
| 13 Neuron | 0.365 | 0.463 | 0.044 | 0.102 | 0.104 | -0.069 | 0.147 | -0.295 | 0.317 | -0.336 |
| 14 Neuron | 0.014 | -0.549 | -0.353 | 0.042 | -0.362 | -0.146 | 0.243 | 0.205 | 0.302 | 0.028 |
| 15 Neuron | 0.235 | 0.773 | 0.217 | 0.337 | 0.416 | -0.505 | 0.22 | -0.414 | 0.139 | -0.313 |
| 16 Neuron | -0.546 | 0.474 | 0.095 | 0.163 | 0.216 | -0.157 | 0.32 | 0.062 | -0.073 | 0.016 |
| 17 Neuron | 0.18 | -0.079 | -0.196 | 0.153 | 0.102 | 0.491 | -0.102 | -0.318 | -0.326 | 0.439 |
| 18 Neuron | 0.021 | -0.221 | -0.202 | -0.121 | -0.452 | 0.02 | -0.103 | 0.011 | 0.601 | 0.058 |
| 19 Neuron | 0.326 | 0.414 | -0.636 | -0.004 | -0.348 | 0.338 | -0.046 | 0.657 | -0.128 | 0.048 |
| 20 Neuron | -0.218 | 0.056 | -0.216 | 0.116 | 0.05 | -0.21 | -0.056 | 0.069 | 0.038 | 0.15 |
| 21 Neuron | -0.734 | 0.194 | -0.201 | -0.69 | 0.798 | -0.167 | 0.022 | -0.385 | -0.611 | 0.738 |
| 22 Neuron | -0.267 | 0.257 | 0.534 | -0.019 | 0.202 | -0.003 | -0.195 | -0.227 | -0.401 | -0.175 |
| 23 Neuron | 0.201 | -0.186 | -1.002 | -0.412 | 0.267 | -0.03 | -0.377 | -0.283 | 0.14 | -0.66 |
| 24 Neuron | 0.054 | 0.129 | 0.062 | -0.015 | 0.211 | 0.16 | -0.144 | -0.171 | -0.125 | 0.319 |
| 25 Neuron | -0.028 | -0.543 | -0.15 | -0.342 | -0.467 | -0.014 | 0.201 | 0.51 | 0.5 | -0.936 |
| 26 Neuron | -0.364 | 0.01 | -0.127 | -0.769 | 0.865 | -0.07 | 0.259 | -0.408 | -0.573 | 0.635 |
| 27 Neuron | 0.052 | -0.154 | -0.099 | 0.058 | -0.711 | 0.669 | 0.652 | 0.165 | -0.599 | -0.384 |
| 28 Neuron | -0.444 | 0.206 | 0.029 | -0.171 | 0.291 | -0.102 | -0.254 | -0.704 | -0.257 | 0.331 |
| 29 Neuron | 0.203 | 0.009 | -0.07 | 0.153 | -0.318 | -0.396 | -0.285 | 0.4 | 0.476 | -0.293 |
| 30 Neuron | 0.256 | -0.368 | -0.299 | -0.087 | 0.002 | -0.213 | -0.186 | 0.066 | 0.532 | 0.226 |

| **Hidden layer-output weight matrix** |
|-----------------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
given in Table 3). For creating a feed-forward back propagation network, first we define the number of hidden layers and number of neurons in the hidden layer. In this study, Levenberg–Marquardt training algorithm with Bayesian regularization in single hidden layer with 30 neurons is used. Tansigmoid transfer function is used for the input layer and pure linear transfer function is used for the output layer. Figure 3 shows the resulting network.

(iii) Training: The neural network was trained using the Levenberg–Marquardt backpropagation (trainlm) method. The regression shows that the training is perfect as shown in Figure 4 having $R^2$ value as 99.94%.

(iv) Validation: Test data set were validated with the trained neural network. Figure 5 shows the validation with $R^2 = 0.9979$ and mean square error is $8.2745 \times 10^{-4}$. Minimum mean squared error (MSE) is used as the criterion for the network selection and also for the stopping of weights and bias adjustment. Table 4 shows the performance of ANN model on both training and testing data sets. These results indicate that the neural network was trained perfectly.

Table 4. Prediction error by ANN-based model.

| Prediction performance by ANN model | Training data sets | Testing data sets |
|-------------------------------------|--------------------|-------------------|
| RMSE                                | $7.8258 \times 10^{-4}$ | $802.745 \times 10^{-4}$ |
| $R^2$                               | 0.9981             | 0.9979            |
| Optimum number of nodes             | 7                  |                   |
| Input activation function           | Tansigmoid         |                   |
| Output activation function          | Purelin            |                   |
| Best training algorithm             | Levenberg–Marquardt|                   |

3.2. Support vector regression

Steps for SVM model

(i) Data generation: Data generation in SVM is similar to the neural network model as described earlier by using fundamental model as the real process. Training data sets were chosen as 5001 samples and test data set contained 2001 samples with 100 s sampling interval. These data sets were used to generate an input matrix consisting of output variable and manipulated variable.

(ii) Selection of kernel: SVM handles nonlinear systems by using “kernel function.” Different kernel types such as linear, polynomial and radial basis function (RBF) have been trained, validated and tested for best prediction using these data sets. There are two parameters to be evaluated to design a successful regression model. These parameters are kernel type and kernel parameter.
N. Sharma and K. Singh

Figure 6. Performance of LS-SVM for different kernel type.

Figure 7. Process output prediction by trained SVM for testing data.

(iii) Tuning: In the formulation of LS-SVM, the model includes two hyper-parameters, i.e. regularization parameter ($\gamma$) and kernel parameter ($\sigma$). The objective of tuning is to find the optimum values of the tuning parameters ($\gamma$ and $\sigma$). Finding the hyperparameters with a good generalization performance is crucial for the successful application of LS-SVM. A popular way to estimate the generalization performance of a model is cross-validation (CV). In this study, we are using 10-fold ($k$-fold) CV, which means the following:

(a) It breaks data into 10 sets of size $n/10$,
(b) Train on 9 data sets and test on 1 data set and
(c) Repeat 10 times and take a mean accuracy.

SVM algorithm with CV cost function demands a high computational load. It finds the value of tuning parameters in approximately 40 min on a computer with Intel core i-5 processor and 4 GB RAM; the execution time was found to be much higher than that in the neural network (1 minute only) to train network. Low value of regularizing parameter increases the complexity of the model and high value of $\gamma$ fitted good on training data points, whereas large value of kernel parameter ($\sigma^2$) indicates a stronger smoothing. Optimal value of regularization parameter ($\gamma$) is $1.3092 \times 10^5$ and kernel parameter ($\sigma^2$) is $90.5660$ with RBF kernel (as given in Table 5).

3.3. Statistical analysis of the model

The statistical analysis of the NN and SVR prediction is based on the following performance criteria:

(1) RMSE on test data set should be the minimum

$$RMSE = \sqrt{MSE(Y_{\text{experimental}} - Y_{\text{predicted}})}.$$ 

(2) The cross-correlation co-efficient ($R^2$) between the input and the output should be around unity

$$R^2 = \frac{\sum((Y_{\text{experimental}} - \text{mean}(Y_{\text{experimental}}))(Y_{\text{predicted}} - \text{mean}(Y_{\text{predicted}})))^2}{\sum((Y_{\text{experimental}} - \text{mean}(Y_{\text{experimental}}))^2 \sum((Y_{\text{predicted}} - \text{mean}(Y_{\text{predicted}}))^2 \sum((Y_{\text{predicted}} - \text{mean}(Y_{\text{predicted}}))^2)}.$$ 

Prediction performances of both the controllers are almost similar as given in Table 6 in terms of mean absolute percent error (MAPE), RMSE and $R^2$.

Table 5. Performance of SVM for different kernel types.

| Type of Kernel | RMSE Training | RMSE Testing | $R^2$ Training | $R^2$ Testing | Cost CV | Regularization parameter ($\sigma^2$) | Kernel parameter ($\sigma^2$) |
|---------------|---------------|--------------|---------------|---------------|---------|--------------------------------------|-----------------------------|
| Linear        | 0.0024        | 0.0024       | 0.9811        | 0.9881        | 6.0731 $\times 10^{-6}$ | 6.2948 | --                                   |
| Polynomial    | $6.1157 \times 10^{-4}$ | $7.8614 \times 10^{-4}$ | 0.9988 | 0.9981 | 5.1405 $\times 10^{-7}$ | 0.0340 | 18.6222 | |
| RBF           | $5.6875 \times 10^{-4}$ | $7.5817 \times 10^{-4}$ | 0.999 | 0.9982 | $4.8789 \times 10^{-7}$ | $1.3092 \times 10^5$ | 90.5660 | |
Table 6. Comparison of performance of SVM and ANN models.

|                  | Prediction performance by ANN | Prediction performance by SVM |
|------------------|------------------------------|------------------------------|
| RMSE             | $8.2745 \times 10^{-4}$      | $7.5019 \times 10^{-4}$      |
| MAPE             | 0.0528                       | 0.0445                       |
| $R^2$            | 0.9978                       | 0.9982                       |

4. Design of artificial intelligence-based model predictive controller

NNPC and SVMPC are basically a model-based predictive control where the model for predictions is the neural network and SVM model. It offers an alternative approach to modeling process behavior as they do not necessarily require a previous knowledge of the process phenomena. They are taught to emulate a process by “training” them, in which they are exposed to sets of input–output data and a least-squares optimization is performed. During this optimization, the neural network and SVM forms its own model of the process which can be used to predict output(s) for a given set of inputs. This model is then used for prediction in MPC algorithm. Figure 8 represents the block schemes of the control configuration. NNPC and SVMPC uses a neural network and SVM model of the process, a history of past control moves and an optimization cost function over the receding prediction horizon to calculate the optimal control moves (Sharma & Singh, 2012). The control evaluation consists of minimization of cost function given by

$$ f = Q \sum_{i=1}^{N_p} (y_{c,i} - y_{sp})^2 + \sum_{i=1}^{N_c} S(\Delta u_i)^2, $$

where $y_{sp}$ is the set-point value of controlled variable, $Q$ is the weighting co-efficient for the relative importance of error between set-point and actual values and $S$ is the weighting co-efficient penalizing relative big changes in the manipulated variable.

The objective function is subject to the following constraints:

$$ u_{\min} \leq u_i \leq u_{\max} (i = 1, \ldots, N_c - 1) $$

(Constraint on manipulated variable),

$$ \Delta u_{\min} \leq \Delta u_i \leq \Delta u_{\max} (i = 1, \ldots, N_c - 1) $$

(Constraint on control move),

$$ y_{\min} \leq y_i \leq y_{\max} (i = 1, \ldots, N_p - 1) $$

(Constraint on process variable),

where $y_{\max}$ and $y_{\min}$, $u_{\max}$, and $u_{\min}$, $\Delta u_{\max}$ and $\Delta u_{\min}$ are upper and lower bounds for the vectors $y$, $u$ and $\Delta u$, respectively. $N_c$ and $N_p$ are the control horizon and prediction horizon. The optimization algorithm calculates the values of $\Delta u$ to minimize the cost objective function. The predictions of the neural networks and support vectors are corrected by a process/model error:

$$ y_d = y_k - y'' $$

$$ y_c = y_m + y_d, $$

where $y_k$ is the vector of measured outputs at the present sampling interval, $y''$ is the vector of the network predictions (calculated at the previous sampling interval), $y_c$ is the vector of the corrected model predictions and $y_m$ is the value predicted by the neural network and SVM.

As the neural networks and SVM are trained to predict the controlled variables only one step ahead, in the control algorithm the networks are iterated to obtain a total of $N_p$ future predictions by using the outputs of the neural networks and SVM as their own inputs in the next iteration.

Although the optimization is based on a control horizon, only the first control action (for each manipulated variable) is implemented in the process and the optimization problem is solved again at the next sampling interval. Matlab® function fmincon was used to solve the optimization problem. The constraints on variables were taken as follows:

$$ 14.727 \leq u \leq 16.72, $$

$$ -0.1 \leq \Delta u \leq +0.1, $$

$$ 0.88 \leq y \leq 0.95. $$

Figure 8. Algorithm of the neural network and SVMPC.
5. Parameter optimization of NNPC and SVMPC

5.1. Weighting co-efficient

Figure 9 shows the effect of weighting co-efficients $Q$ and $S$ on the NNPC methodology. Usually, the value of $Q$ is selected to be much higher than the $S$ factor. If we increase the value of $S$, then system becomes sluggish and takes more time to obtain steady-state condition (Figure 9(a)), whereas on increasing $Q$, system responds faster to reach to set point (Figure 9(b)). When we further increase the value of $Q$ (beyond 1000), there is not much change in the response of both controllers. Table 7 shows the effect of $Q$ and $S$ in terms of performance indexes for set-point change from 0.92506 to 0.93506. For all these performance criteria, the error is minimum with $Q$ value as 1000 and $S$ as 0.03. Therefore, for further study of the NNPC for different set-point and load changes, the values of $Q$ and $S$ were specified as 1000 and 0.03, respectively.

5.2. Integral action

To improve the performance of the controller, we add integral action in the algorithm of the NNPC methodology. We observed that for large load changes in methanol feed flow rate such as from 65.3 to 50 mol/s at 1000 s, controller shows offset. As shown in Figure 10, dotted line shows the response of controller without integral action, whereas dashed and solid lines show the response of controller with integral action with different values of integral constant ($0.001$ and $0.1$). When methanol feed flow rate is given a step decrease, then initially TAME composition increases and then after some time comes back to the set point. The result shows that controller gives better performance with integral constant of 0.001.

5.3. Effect of prediction horizon

Prediction horizon ($N_p$) represents the number of samples into the future over which NNPC computes the predicted process variable profile and minimizes the predicted error. Control horizon ($N_c$) is the number of manipulated variable moves that controller computes at a given sampling interval to eliminate the current predicted error. Figure 11 shows a comparison between the responses for different values of the prediction horizon. In both the cases (Figure 11), with higher values of the prediction horizon, controller shows a sluggish behavior. Lower value of $N_p$ is suggested. The error is found to be minimum for $N_p = 3$ in the NNPC.

| Parameter | ITAE | IAE | ISE | ITSE |
|-----------|------|-----|-----|------|
| $Q = 10$  | $1.6913 \times 10^3$ | 3.3351 | 0.01365 | 4.08469 |
| $Q = 100$ | $4.1913 \times 10^2$ | 1.4981 | 0.00588 | 1.15512 |
| $Q = 1000$| $3.3472 \times 10^2$ | 1.2868 | 0.005113 | 0.91443 |
| $S = 0.03$| $3.3471 \times 10^2$ | 1.2868 | 0.005113 | 4.08469 |
| $S = 1$   | $7.3009 \times 10^1$ | 2.1011 | 0.00858 | 1.15509 |
| $S = 10$  | $7.5642 \times 10^1$ | 2.24745 | 0.028628 | 1.15509 |

Figure 9. Effect of weighing co-efficients (a) $S$ and (b) $Q$ on performance of the NNPC methodology.

Figure 10. Effect of integral action on performance of the NNPC methodology for load change in flow rate of methanol from 65.3 to 50 mol/s.

Figure 11. Effect of prediction horizon on performance of the NNPC methodology.
Due to similar prediction performance (as shown in Table 6), same weighting co-efficients \((Q, S)\) and integral constant were taken with the SVMPC methodology. Response of prediction horizon implies a larger impact on the performance of SVMPC. As shown in Figure 12 as we increase the value of prediction horizon, by taking other parameters same as the NNPC, controller gives an offset. Value of prediction horizon has been taken as 1 with SVMPC.

6. Results and discussion

Responses of both control methodologies were compared with the PID control methodology. The values of \(K_c\) and \(\tau_I\) were specified as 1 and 0.1, respectively, as discussed in our previous paper (Sharma & Singh, 2012).

6.1. Set-point change

As a base case, reboiler heat duty was fixed at 15.727 MW with a TAME composition of 0.92506 and reflux ratio fixed at 1. To observe the effect of NNPC and SVMPC, different simulation runs were performed at different step changes in set point as shown in Figure 13. Upper part of this figure shows NNPC, SVMPC and PID responses of reactive distillation column for step changes of \(+0.01\) and \(-0.01\) in the set point of the TAME mole fraction in the bottom product. Lower part of the figure shows the

Figure 12. Effect of prediction horizon on performance of the SVMPC.

Figure 13. Responses of the SVMPC (green dashed lines), NNPC (red solid) and PID controller (dotted lines) for set-point change.
corresponding reboiler heat duty. It is depicted from the figure that both NNPC and SVMPC give exactly similar response, due to similar prediction performance, which can be seen in the expanded part of top-right corner of the figure. The result shows that both NNPC and SVMPC obtained steady state at around 1000 s, whereas the PID controller shows a sluggish response (obtained steady state at around 4000 s).

6.2. Load change in methanol feed flow rate
A load change was given in methanol feed flow rate from 65.3 to 71.83 (+10%) mol/s at 1000 s and another step change was given in the feed flow rate from 65.3 to 58.77 (−10%) mol/s at 1000 s. The corresponding responses of both controllers were compared in Figures 14 and 15. On giving a step increase in methanol feed flow rate, TAME composition decreases initially and after some time is brought back to the set point (Figure 14). Similarly, when methanol feed flow rate is given a step decrease, then initially TAME composition increases and then after some time is brought back to the set point (Figure 15). It is clear from these figures that for load change in methanol feed flow rate, NNPC and SVMPC work well in comparison to the PID controller.

6.3. Load change in molar ratio
Molar ratio of methanol to IA in reactor effluent feed implies a larger impact on TAME purity and IA conversion. The methanol feed to the reactive column is very important. Less methanol in feed will result in low conversion and purity. This occurs because methanol forms an azeotrope with isopentane, which remains in significant amount in distillate. When the methanol is reduced, the amount of methanol needed in the azeotrope does not change and less methanol is available for reaction. Further increase in methanol will result in higher conversion and purity (molar ratio 0.5–1.5). Further increasing methanol will result in (i) reducing TAME purity because most of the excess methanol will leave the column in the bottoms and (ii) reducing the IA conversion because some amount of product leave in the distillate. Therefore, both reactants must be used in correct proportions to maintain the high conversion and purity of the TAME. Initially, molar ratio of 1.5 was considered for the study.

A load change was given in molar ratio of methanol to IA in reactor effluent feed from 1.5 to 1.65 (+10%) at 1000 s and another load change was given in the molar ratio from 1.5 to 1.35 (−10%) at 1000 s. The corresponding responses of both controllers were compared in Figures 16 and 17. On giving a step increase in molar ratio, TAME composition decreases initially and after some time is brought back to the set point (Figure 16). Similarly, when molar ratio is given a step decrease, then initially TAME composition increases and then after some time is brought back to the set point (Figure 17). It is depicted from the figure that both NNPC and SVMPC give exactly similar response, which can be seen in the expanded part of top-right corner of the figure. It is clear from these figures that for load change in molar ratio of methanol to IA in reactor, effluent feed machine learning-based control methodology works well in comparison to the conventional PID control methodology.

6.4. Performance of the control system
Model is said to be optimal with respect to cost criteria if the cost is the lowest when using that model. In general, there

Figure 14. Responses of the SVMPC, NNPC and PID controllers for load change in flow rate of methanol from 65.3 to 71.83 (+10%) mol/s.
Figure 15. Responses of the SVMPC, NNPC and PID controllers for load change in flow rate of methanol from 65.3 58.77 (−10%) mol/s.

Figure 16. Responses of the SVMPC, NNPC and PID controllers for load change in molar ratio of methanol to IA in reactor effluent feed from 1.5 to 1.65 (+10%).

are different types of performance criteria of the controllers such as the IAE, ISE, ITAE and ITSE. The IAE and ISE cost criteria weight the initial values of the error more than the later value, while the time weighted criteria such as the ITAE and ITSE criteria weight the later values of the error more. Minimizing the integral constraint tries to keep the
error small in general sense.

\[
\text{ITAE} = \int_0^t (y - y_{sp}) \, dt,
\]
\[
\text{IAE} = \int_0^t |y - y_{sp}| \, dt,
\]
\[
\text{ISE} = \int_0^t (y - y_{sp})^2 \, dt,
\]
\[
\text{ITSE} = \int_0^t (y - y_{sp})^2 \, dt.
\]

In this paper, all these indexes were used to estimate the performance of the NNPC, SVMPC and PID. Tables 8–10 show the comparison between the responses of all the controllers for set-point change, load change in methanol feed rate and molar ratio of methanol to IA in reactor effluent feed. For all these performance criteria, the error is minimum with NNPC and SVMPC. Therefore, the implementation of machine learning-based controller to the present system has proved to be effective in maintaining the desired purity in the presence of disturbances and set-point changes.

Table 8. Performance criteria for positive (+0.01) and negative (−0.01) set-point change using SVMPC, NNPC and PID controllers.

| Performance criteria | Positive set-point change (+0.01) | Negative set-point change (−0.01) |
|----------------------|-----------------------------------|----------------------------------|
| ITAE                 | SVMPC 4.178 × 10^3                | NNPC 3.418 × 10^3                |
|                      | PID 7.187 × 10^3                  | NNPC 3.036 × 10^3                |
| IAE                  | SVMPC 2.959                       | NNPC 1.811                       |
|                      | PID 8.525                         | NNPC 6.004                       |
| ISE                  | SVMPC 0.012                       | NNPC 0.010                       |
|                      | PID 0.040                         | NNPC 0.033                       |
| ITSE                 | SVMPC 1.0656                      | NNPC 0.7395                      |
|                      | PID 17.291                        | NNPC 9.5635                      |

Table 9. Performance criteria for positive (+10%) and negative (−10%) load change in methanol feed flow rate using SVMPC, NNPC and PID controllers.

| Performance criteria | Positive set-point change (+10%) | Negative set-point change (−10%) |
|----------------------|-----------------------------------|----------------------------------|
| ITAE                 | SVMPC 1.8679 × 10^3               | NNPC 1.6495 × 10^3               |
|                      | PID 12.762 × 10^3                 | 12.997 × 10^3                   |
| IAE                  | SVMPC 1.2893                      | NNPC 1.4083                      |
|                      | PID 6.9745                        | 6.7881                          |
| ISE                  | SVMPC 0.0051                      | NNPC 0.0048                      |
|                      | PID 0.0331                        | 0.0272                          |
| ITSE                 | SVMPC 6.2961                      | NNPC 6.0036                      |
|                      | PID 51.1814                       | 43.8539                         |
Table 10. Performance criteria for positive (+10%) and negative (−10%) load change in feed ratio (methanol/IA) using SVMPC, NNPC and PID controllers.

| Performance criteria and controller | Positive load change (+10%) | Negative load change (−10%) |
|------------------------------------|-----------------------------|-----------------------------|
| ITAE                               | SVMPC 0.7590 × 10^3 0.7248 × 10^3 | PID 2.8246 × 10^3 2.6048 × 10^3 |
| IAE                                | SVMPC 0.6023 0.5928 | NNPC 0.6018 0.5784 | PID 1.9548 1.8256 |
| ISE                                | SVMPC 7.4753 × 10^-4 7.7341 × 10^-4 | NNPC 7.4742 × 10^-4 7.3745 × 10^-4 | PID 0.0042 0.0037 |
| ITSE                               | SVMPC 0.8356 0.8725 | NNPC 0.8276 0.8422 | PID 5.4707 4.9085 |

7. Conclusions

NNPC and SVMPC methodologies were applied and compared to control the product purity in bottoms of reactive distillation column. Both controllers were compared for set-point changes as well as load changes in feed flow rate of methanol and molar ratio of methanol to IA in reactor effluent feed. Responses of both the controllers were then compared with the PID control methodology. SVM algorithm demands a high computational load due to the form of its optimization problem. Neural network provides good generalization results, if the structure is suitably chosen. In this study, it was found that for large disturbances, both NNPC and SVMPC shows offset. Addition of an integral action in these controllers improves the performance of the controllers, thus it is suggested that the machine learning-based control methodology with integral action could be advantageous for nonlinear processes. In comparison to PID, machine learning-based control methodologies, i.e. SVMPC and NNPC in the present study, show better response for both disturbance rejection and set-point tracking as justified by several performance criteria, namely ITAE, ISE, IAE and ITSE.

References

Ahmad, Z., & Mat Noor, R. (2009). Multiple neural networks modeling techniques in process control: A review. Asia-Pacific Journal of Chemical Engineering, 4, 403–419.

Al-Arfaj, M. A., & Luyben, W. L. (2004). Plantwide control for TAME production using reactive distillation. AIChE Journal, 50, 1462–1473.

Arumugasamy, S. K., & Ahmad, Z. (2009). Elevating model predictive control using feedforward artificial neural networks: A review. Chemical Product and Process Modeling, 4, 1–40.

Áström, K. J., & McAvoy, T. J. (1992). Intelligent control. Journal of Process Control, 2, 115–127.

Bao, Z., Pi, D., & Sun, Y. (2007). Nonlinear model predictive control based on support vector machine with multi-kernel. Chinese Journal of Chemical Engineering, 15, 691–697.

Basak, D., Pal, S., & Patranabis, D. C. (2007). Support vector regression. Neural Information Processing – Letters and Reviews, 11, 203–224.

Bhat, N., & McAvoy, T. J. (1990). Use of neural nets for dynamic modeling and control of chemical process systems. Computers & Chemical Engineering, 14, 573–582.

Deng, Z., Cao, H., Li, X., Jiang, J., Yang, J., & Qin, Y. (2010). Generalized predictive control for fractional order dynamic model of solid oxide fuel cell output power. The Journal of Power Sources, 193, 8097–8103.

Garcia, C. E., & Morshedl, A. M. (1986). Quadratic programming solution of dynamic matrix control (QDMC). Chemical Engineering Communications, 46, 73–87.

Hui, L. Q., Hui, L. W., Aziz, N., & Ahmad, Z. (2011). Nonlinear process modeling of “shell” heavy oil fractionator using neural network. Journal of Applied Sciences, 11, 2114–2124.

Hussain, A. (1999). Review of the applications of neural networks in chemical process control – simulation and online implementation. Artificial Intelligence in Engineering, 13, 55–68.

Ipliki, S. (2010). A support vector machine based control application to the experimental three-tank system. ISA Transactions, 49, 376–386.

Kalogirou, S. A. (2003). Artificial intelligence for the modeling and control of combustion processes: A review. Progress in Energy and Combustion Science, 29, 515–566.

Kataria, A. M., Kanath, R. S., Moudgalya, K. M., & Mahajan, S. M. (2008). Non-equilibrium stage modeling and non-linear dynamic effects in the synthesis of TAME by reactive distillation. Computers & Chemical Engineering, 32, 2243–2255.

Kataria, A. M., Moudgalya, K. M., & Mahajan, S. M. (2006). Nonlinear dynamic effects in reactive distillation for synthesis of TAME. Journal of Industrial and Engineering Chemistry, 43, 4233–4242.

Konakom, K., Kittisupakorn, P., Saengchan, A., & Mujtaba, I. M. (2010). Optimal policy tracking of a batch reactive distillation by neural network-based model predictive control (NMPC) strategy. World congress on engineering and computer science (WCECS 2010), San Francisco.

Lawrynczuk, M. (2010). Training of neural models for predictive control. Neurocomputing, 73, 1332–1343.

Li, L. J., Su, H. Y., & Chu, J. (2007). Generalized predictive control with online least squares support vector machines. Acta Automatica Sinica, 33, 1182–1188.

Liu, Y., Wang, H., & Li, P. (2008). Kernel learning adaptive one-step-ahead predictive control for nonlinear processes. Asia-Pacific Journal of Chemical Engineering, 3, 673–679.

Luyben, W. L., & Yu, C.-C. (2008). Reactive distillation design and control. Hoboken, NJ: John Wiley & Sons.

Mohl, K. D., Kienle, A., Gilles, E. D., Rapmund, P., Sundmacher, K., & Hoffmann, U. (1997). Nonlinear dynamics of reactive distillation processes for the production of fuel ethers. Computers & Chemical Engineering, 21, S989–S994.

Mohl, K. D., Kienle, A., Gilles, E. D., Rapmund, P., Sundmacher, K., & Hoffmann, U. (1999). Steady-state multiplicities in reactive distillation columns for the production of fuel ethers MTBE and TAME: Theoretical analysis and experimental verification. Chemical Engineering Science, 54, 1029–1043.

Qin, S. J., & Badgwell, T. A. (1997). An overview of industrial model predictive control technology. AIChE Journal, 43, 1462–1473.
Ramchandran, S., & Russell Rhinehart, R. (1995). A very simple structure for neural network control of distillation. *Journal of Process Control, 5*, 115–128.

Rewagad, R. R., & Kiss, A. A. (2011). Dynamic optimization of a dividing-wall column using model predictive control. *Chemical Engineering Science, 66*, 132–142.

Sharma, N., & Singh, K. (2010). Control of reactive distillation column: A review. *International Journal of Chemical Reactor Engineering, 8*, R5, 1–57.

Sharma, N., & Singh, K. (2012). Model predictive control and neural network predictive control of TAME reactive distillation column. *Chemical Engineering and Processing, 59*, 9–21.

Suykens, J. A. K., Van Gestel, T., De Brabanter, J., De Moor, B., & Vandewalle, J. (2002). *Least squares support vector machines*. Singapore: World Scientific.

Suykens, J. A. K., and Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters, 9*, 293–300.

Tatjewski, P., & Lawrynczuk, M. (2006). Soft computing in model-based predictive control. *International Journal of Applied Mathematics and Computer Science, 16*, 7–26.

Thibault, J., & Grandjean, B. P. (1991). *Neural networks in process control – a survey*. Advanced Control of Chemical Processes – ADEHEM’91, IFAC Symp., Toulouse, France, pp. 251–260.

Xi, X. C., Poo, A. N., & Chou, S. K. (2007). Support vector regression model predictive control on a HVAC plant. *Control Engineering Practice, 15*, 897–908.

ZareNezhad, B., & Aminian, A. (2011). Application of the neural network-based model predictive controllers in nonlinear industrial systems: Case study. *Journal of the University of Chemical Technology and Metallurgy, 46*, 67–74.

Zhang, R., & Wang, S. (2008). Support vector machine based predictive functional control design for output temperature of coking furnace. *Journal of Process Control, 18*, 439–448.

Zhang, R., Xue, A., & Wang, S. (2011). Modeling and nonlinear predictive functional control of liquid level in a coke fractionation tower. *Chemical Engineering Science, 66*, 6002–6013.