

Inferential Sensor for Estimation of the Concentration of Benzene in the Distillation Column Using TSK Fuzzy System Based on Modified Clustering Approach

Babak Ghanaati, Mehdi Shahbazian

Department of Instrumentation and Automation Engineering, Petroleum University of Technology, Ahwaz, Iran

Email address:

shahbazian@put.ac.ir (M. Shahbazian)

To cite this article:

Babak Ghanaati, Mehdi Shahbazian. Inferential Sensor for Estimation of the Concentration of Benzene in the Distillation Column Using TSK Fuzzy System Based on Modified Clustering Approach. *American Journal of Chemical Engineering*. Vol. 5, No. 6, 2017, pp. 122-129. doi: 10.11648/j.ajche.20170506.11

Received: July 23, 2017; **Accepted:** August 21, 2017; **Published:** November 5, 2017

Abstract: An inferential sensor is a computer program used for inferring the process variables, which are very hard to measure from the available measurement data. Measurement noises can affect the quality of the data which can be improved by wavelet denoising method. The objective of this paper is to design an inferential sensor for estimation of Benzene concentration in a typical distillation column. Selection of the most relevant input variables for estimation can improve the performance of inferential sensor which is done by Principal Component Analysis (PCA) technique. In this paper an inferential sensor is proposed based on a novel modification of the nearest neighbor distance-based clustering for developing a Takagi-Sugeno-Kang (TSK) fuzzy model optimized by the Particle Swarm Optimization (PSO) algorithm. The proposed technique was compared against the conventional nearest neighbor distance-based clustering approach optimized by PSO. The simulation results confirm that the designed inferential sensor based on the proposed method is more accurate even for a noisy data set.

Keywords: Inferential Sensor, Distillation Column, Takagi-Sugeno-Kang Fuzzy System, Nearest Neighborhood Clustering, Particle Swarm Optimization, Wavelet, Principal Component Analysis

1. Introduction

Inferential sensors [1-3] are the estimation models, which employ measurable variables (like temperature and flow) to predict the variables that can be hardly measured, because of the technological constraints, long measurement periods, and considerable expenses. Moreover, they can detect fault and validate the measurements of physical sensors as a backup sensor [4]. Generally, inferential sensors can be divided to two main categories which are the model-driven and data-driven inferential sensors. The model-driven ones are commonly based on physical and chemical equations governing to the plant. The model-driven method is much more expensive, incredibly complex, and sometimes impractical, because of the high complexity of industrial processes [5]. Subsequently, the data-driven inferential sensors got much more popularity, because they only deal with measurement data and does not participate in principles of processes. As a consequence, data-driven inferential

sensors have been broadly employed in the process industry, considering the fact that typical models are based on data measured in process plants and make the model of real processes [6]. There are various techniques available for modelling the process by data-based inferential sensors, each with advantages and disadvantages [5]. An overview of different inferential sensor modelling approaches is given in [7]. The most common techniques used in designing the inferential data-driven sensor are the Artificial Neural Network (ANN) [8], Fuzzy systems including clustering method [9], Partial Least Square (PLS) [10], Neuro-Fuzzy systems (NFS) [11], Principle Component Analysis (PCA) [12] and Support Vector Regression (SVR) [13, 14]. Additionally, the data-driven inferential sensor can be designed using hybrid techniques [15].

Fuzzy models are knowledge-based models and in complex applications it is difficult to tune them. This difficulty can be overcome by optimizing the fuzzy model using evolutionary algorithms. One of the most famous types

of fuzzy systems are Takagi-Sugeno-Kang (TSK) fuzzy systems [16]. In [17], the TSK-fuzzy model is tuned using a GA-based approach. In [18], the work of [17] was applied to learn the TSK-fuzzy with the selection of together input variables and delays together.

Fuzzy rule-based models of the so-called Takagi-Sugeno-Kang (TSK) have been broadly utilized as a main part of modelling and process control of [19] and [20]. The major reason for employing TSK models is their dual nature: they are not linear generally but are locally linear in their sub-models. Indeed, TSK models have multi-model architecture with fuzzily defined zones of validity of local models. The best feature of the TSK models is that because the local regions are fuzzily described as simple as possible, the global model can be nonlinear and suitable for complex chemical systems. In [21] a systematic approach is presented to design first order TSK fuzzy systems. Commonly linear (first-order) or constant (zero-order) sub-models are employed, though architecture would be less complex in zero-order type [22]. The offline clustering approach can be applied to the construction of TSK models, which have several stages to be identified [23, 24]. It is respected that as a general rule, in any case, there is no guarantee that a model will perform satisfactory regardless of successful validation. Design of inferential sensor in zero-type Fuzzy TSK based on modified nearest neighbor distance-based clustering (MNND) improved by PSO for a typical distillation column is the aim of presented work. Employment of zero-type fuzzy TSK can decrease the complexity of fuzzy system and it would be faster in response.

This paper is organized as follows. The methodology is thoroughly explained in Section 2. The case study and results and comparison between the proposed and existing methods [25] is introduced in Section 3. Finally, the paper is concluded in Section 4.

2. Methodology

The Fuzzy System (FS) has been introduced by Zadeh [26]. A typical fuzzy system includes four basic parts as shown in Figure 1: 1) fuzzifier 2) rule base; 3) inference engine; 4) defuzzifier.

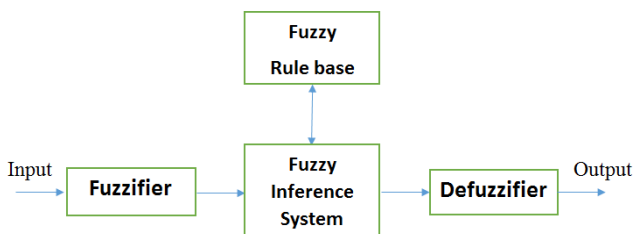


Figure 1. A general fuzzy system scheme.

In a fuzzy system, the input is converted from crisp to fuzzy according to input MFs. The result is employed by inference engine to generate fuzzy output, and finally, a crisp value y is produced by defuzzifying the output value [25]. It is based on a set of rules written as in the following form:

IF (antecedent) THEN (consequent)

Two basic groups of FSs had been used during the last decades. These two groups differ by the type of the consequent part.

a. Mamdani models

A fuzzy rule in Mamdani FS is written as follows:

IF (X_1 is A_1) AND (X_2 is A_2)... Then (y is B)

Where X_1, X_2, \dots are the input variables, A_1, A_2, \dots are the input fuzzy sets, and B is the output fuzzy set.

b. Takagi-Sugeno-Kang models

The fuzzy rules in Takagi-Sugeno-Kang (TSK) models are defined as follows:

IF (X_1 is A_1) AND (X_2 is A_2) ... Then $y = b_0 + b_1X_1 + \dots + b_nX_n$

Where b_0, b_1, \dots are the constants in the linear consequent part which represents the main difference between Mamdani and TSK models. Takagi-Sugeno FIS is constructed by a set of fuzzy rules in which the antecedent part represents TSK a fuzzy partitioning or clustering of the input space, and the output is calculated using a linear function, generally. The proposed TSK model is zero-type model that can be considered as a kind of distance-based clustering, while some modifications have been applied to the conventional nearest neighbor distance-based clustering (CNND). The i th rule in fuzzy rule-base based on zero-type TSK fuzzy model has the following form:

R i : If ($x_1=A_{i1}$) and ... and ($x_j=A_{ij}$) and ... and ($x_n=A_{in}$) Then ($y=b_i$)

Where A_{i1}, \dots, A_{in} are the input fuzzy sets and b_i is a constant in the consequent part [25]. Gaussian membership functions are assumed for A_{ij} fuzzy sets.

$$\mu_{A_{ij}} = \exp \left\{ -\frac{1}{2} \left(\frac{x_j - c_{\mu_{ij}}}{\sigma_{\mu_{ij}}} \right)^2 \right\} \quad (1)$$

Where $c_{\mu_{ij}}$ and $\sigma_{\mu_{ij}}$ are the center and spread of j th Gaussian membership function in i th rule.

For decreasing the run time during the train and test, it has been verified that the input reduction of fuzzy system can be effective [27]. One of the common promising way to reduce the input dimensions is the Principal Component Analysis (PCA), which extract the important features. Indeed, the PCA as the main linear technique for dimensionality reduction convert the data to a lower-dimensional in such a linear way that the variance of the data in the low-dimensional is maximized. Actually, at first, PCA find the eigenvectors and eigenvalues of the data covariance matrix and then sort the columns of the eigenvector matrix and eigenvalue matrix in order to decreasing eigenvalues. The eigen vectors that are correspond to the largest eigenvalues (the principal components) can now be used to reconstruct a large fraction of the variance of the original data. Moreover, the first few eigen vectors can often be employed to describe the behavior of the system [28]. The original space has been reduced (with data loss, but hopefully remaining the most important variance) to the space constructed by a few eigenvectors.

2.1. Modified Nearest Neighbor Distance-Based Clustering (MNND) Based on Zero-Type TSK

The proposed clustering technique (MNNDC) is developed by some modifications applied to the simple distance-based clustering method for more accuracy and decreasing the complexity. The base function of MNNDC is how incoming data find its cluster through the modified way. Generally, the proposed algorithm consists of the following stages:

- a. Normalize the data to [-1, 1].
- b. Select randomly the 70% of the data for training and the remaining as test data.
- c. Start with an empty rule base (set the center matrix to zero), select the radius (r) of the clusters from a suitable initial guess from past experience (in the range [0.2, 0.3] $\sqrt{(P + 1)/2}$) where P is the number of inputs.

Start training:

- d. Take the incoming first train data sample (x_1) and consider that as center vector of the first cluster (c_1) and suppose variances vector of input Gaussian fuzzy sets are ε , so build up the first cluster (rule) i.e.:

$$c_1 = x_1, \sigma_1^2 = \varepsilon \quad (2)$$

Also

$K=1$ (K is number of clusters or rules)

$n_1=1$ (n is the number of data belonged to the first cluster)

- e. Take the next incoming data (x_j)
- f. Calculate the distance of the selected data sample to all existent centers and select the nearest center as the winner cluster (c_{win})
- g. If $\|x_j - c_{win}\| \geq r$

New cluster is constructed and consider the x_j as center of the new cluster, hence $c_{k+1} = x_j$, where x_j includes all dimensions of the current data.

$\sigma_{k+1}^2 = \varepsilon$ (σ_{k+1}^2 is the variance of the Gaussian membership functions of inputs)

$K=K+1$ (number of clusters would be incremented)

$n_{K+1}=1$ (first data sample belonged to the $(k + 1)th$ cluster)

Else

- (a) $n_{win} = n_{win} + 1$, where n_{win} is the number of data samples belonging to the winner cluster
- (b) Update the P+1 components of the center of the winner cluster while training by:

$$c_{win}^{new} = c_{win}^{old} + \alpha_{win}(x_j - c_{win}^{old}) \quad (3)$$

Where $\alpha_{win} = \frac{gain}{n_{win}}$ and *gain* is the correction constant.

- (c) Calculate the P+1 components of the variance of the winner cluster (variance vector of Gaussian membership functions in inputs) by:

$$\sigma_{win}^2 = n_{win}(c_{win}^{new} - c_{win}^{old})^2 + (1 - \beta)(x_j - c_{win}^{old})^2 \quad (4)$$

Where β is a constant that is performing like forgetting factor.

- h. If data samples are coming in for training go to step VI, otherwise stop and offline training has been terminated. The estimated output \hat{y} of the zero-type TSK fuzzy method

by using weighted average defuzzification method is:

$$\hat{y} = \frac{\sum_{i=1}^K \tau_i b_i}{\sum_{i=1}^K \tau_i} \quad (5)$$

Where τ_i is the strength, K is the number of clusters and b_i are the consequent parameters of i th cluster (rule) i.e.

$$\tau_i = \mu_{i1}(x_1) \times \dots \times \mu_{iP}(x_P) \quad (6)$$

It is assumed that b_i is mean of the outputs of data samples limited by the zone of i th cluster. Also, $\mu_{ij}(X)$ is the final fuzzy sets described by Gaussian membership functions after training, i.e.,

$$\mu_{ij} = \exp\left\{-\frac{1}{2}\left(\frac{x_j - c_{\mu_{ij}}}{\sigma_{\mu_{ij}}}\right)^2\right\} \quad (7)$$

Where $c_{\mu_{ij}}$ and $\sigma_{\mu_{ij}}$ are the center and width of the j th Gaussian membership function in the i th rule (cluster) respectively and $j = 1, 2, \dots, P$ and $i = 1, 2, \dots, K$

In the proposed training algorithm, there are two free parameters *gain* and β in Eq. (3) and Eq. (4) respectively, which affect the minimization of prediction error to achieve an accurate model. There are different types of optimizations algorithms that can be utilized for fine tuning the noted parameters. In this paper, the PSO tunes the parameters in the training part, automatically. Particle Swarm optimization shares many similarities with Evolutionary Computation (EC) techniques in general. Compared to most of other ECs, the advantages of PSO are that PSO is easy to implement and there are few parameters to adjust. It should be noted that root mean square error (RMSE) is selected as the cost function for the optimization performance.

2.2. Particle Swarm Optimization (PSO)

The PSO optimization is a parallel computational technique that is based on population, iteratively. Firstly, PSO is invented by Kennedy, Eberhart and Shi [29]. Basically, it has been derived from the behavior of bird flocking.

The PSO starts the solution with a random particles and then searches for optimum solution by updating generations. Let i show a particle's index in the swarm. The feature of each particle is expressed as position vector X , velocity V . All of m particles fly through the d -dimensional search R^d with a velocity V_i . The best position among the particles is selected as the best solution. Iteratively, each particle's position and velocity are update according to the following relations:

$$X_i(k + 1) = X_i(k) + v_i(k + 1) \quad (8)$$

$$v_i(k + 1) = w(k).v_i(k) + c_1.r_1(k).(L_{best_i}(k) - X_i(k)) + c_2.r_2(k).(G_{best_i}(k) - X_i(k)) \quad (9)$$

Where $X_i(k + 1)$ and $X_i(k)$ are the positions of $(k + 1)th$ and kth iteration of i^{th} particle, as $v_i(k + 1)$ and $v_i(k)$ are the velocities of $(k + 1)th$ and kth iteration of i^{th} particle, respectively, and $w(k)$ is the inertial weight coefficient. c_1

and c_2 are the nonnegative fixed parameters, Here they are assumed 2. $r_1(k)$ and $r_2(k)$ are the random numbers distributed uniformly in the range of $[0,1]$. $L_{besti}(k)$ is the best position answer of the i^{th} particle and $G_{besti}(k)$ is the best position answer between all particles in the swarm. In order to restraint the particles to fall into local optimal solution, v_i and X_i are limited within a range $[v_{min}, v_{max}]$ and $[X_{min}, X_{max}]$ in respect. It should be noted that searching is a repeat performance and the stop criteria are the maximum iteration or the minimum error during the search [29]. To make the relation between the proposed algorithm and PSO for optimizing the parameters ($gain, \beta$) in the algorithm, PSO produces sets of particles, which show all of the new approach (modified nearest neighbor distance-based clustering algorithm). Here, the range limitation of the parameters are set as $gain \in [0.3, 0.7]$ and $\beta \in [0, 0.2]$. So the tuned parameters by PSO, which result in minimum RMSE as cost function, are selected as the most suitable parameters.

3. Case Study

3.1. Process Description

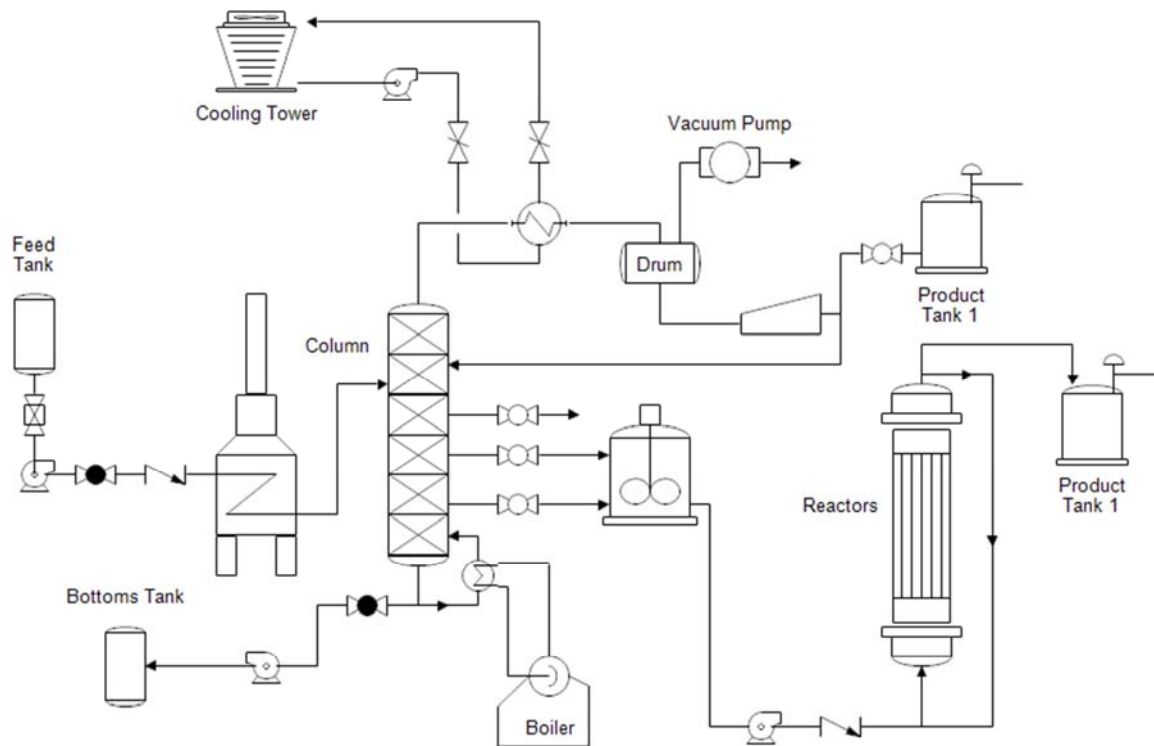


Figure 2. Schematic dynamic of considered distillation column.

3.2. Implementation and Results

Actually, 28 input variables and their time lags can be utilized for modeling of Bottom Benzene concentration in a distillation column [30]. The most relevant inputs for predicting output are chosen using the Principle Component Algorithm (PCA) technique. Due to the process dynamic behavior, 3 regressors for lags of input variables have been considered that affect the output. The most correlated time lag of inputs with output and the range of input variables are

One of the most important and common unit operations in the chemical and petroleum industry is distillation column. After all, its complex behavior and highly un-predictive nature, has made it as a unit operation which is complicated and difficult to handle by engineers. Actually, the control system must manipulate the bottom and the top composition to achieve the demanded purity of products. Measuring the compositions by using sampling in laboratory is really time-consuming and may it faces open loop instability while measuring, so it is needful to predict it with high precision, concurrently with quick response. Benzene concentration in the bottom product of distillation column is the desired variable to be predicted. The considered distillation column in this study, contains 20 valve trays, where the feed enters at tray 4 and the bottom product are extracted at tray 1. The feed stream to the distillation column is compositions of the light hydrocarbons including n-Butane, i-Pentane, n-Pentane, 2,2-Dimethyl Butane, 2-Methyl Pentane, n-Hexane, 2,4-Dimethyl Pentane, Benzene, 2-Methyl Hexane, n-Heptane, Toluene, 2-methyl heptane, Ethylbenzene, o-xylene, 1,3,5-Trimethylbenzene [30]. The Process Flow Diagram of the considered distillation column is shown in Figure 2:

provided in Table 1.

Table 1. Steady state operating condition of distillation column.

Optimum inputs	unit	Variable range
First tray temp(t-1)	°C	[150.09, 160.28]
Second tray temp(t-2)	°C	[144.60, 150.82]
Reboiler temp(t-1)	°C	[156.50, 166.35]
Reflux drum temp(t-4)	°C	[55.50, 65.89]
Feed flow rate(t-4)	kg/hr	[38152, 183283]
Bottom product flow rate(t-1)	kg/hr	[19681, 73288]

After gathering 4320 data samples of the multi component distillation column, with 10s sampling time [6], 3000 data samples (about 70 percent of all data samples) have been randomly considered as training data for making the model and rest of them (unseen data) have been used for validation. Actually, 30 percent of all data samples have been picked up randomly for testing the model. Before constructing the model, all the data is contaminated by a white Gaussian noise of magnitude within ± 15 percent of steady state value of the model inputs. Consequently, Wavelet toolbox in Matlab is employed to reduce the noise.

In order to implement each inferential sensor (based on Conventional and Modified algorithms improved by PSO), two tests are performed. Firstly, dataset is noise free and the second test is performed when the dataset is contaminated by Gaussian white noise with SNR=15.

The parameters *gain* and β of modified clustering approach are optimized by PSO. The specification of the

implemented PSO algorithm are listed in Table 2.

Table 2. Modified Nearest Neighbor Distance-based clustering (MNNDC)-PSO parameters to estimate the Benzene concentration.

Parameters	Values
Maximum number of iteration	50
Population size	20
Personal learning coefficient	1
Global learning coefficient (c_1 and c_2)	2
Inertia weight damping ratio	0.99
gain	0.44
β	0.08

Figures 3 and 5 show the predictions of Bottom Benzene Concentration using CNNDC and proposed MNNDC approaches. The proposed MNNDC algorithm has predicted the concentration much more accurate, obviously. Also, the correlation coefficients shown in Figures 4 and 6 have proved the superiority of the proposed MNNDC over the CNNDC algorithm.

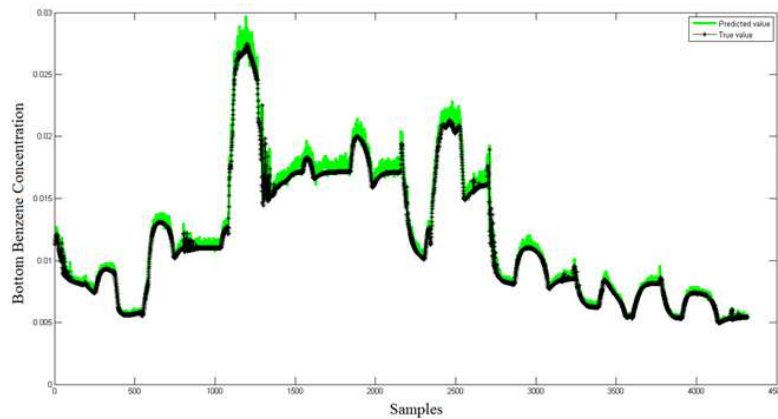


Figure 3. Prediction and true values for Bottom product Benzene concentration using conventional distance-based clustering (CNNDC) improved by PSO after noise elimination using Wavelet De-noising.

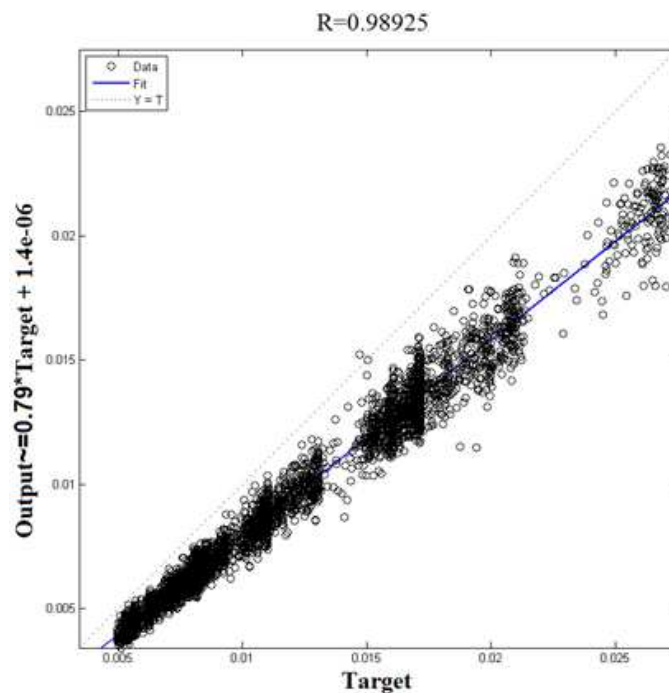


Figure 4. Correlation coefficient value for Bottom Benzene concentration using CNNDC improved by PSO after noise elimination using Wavelet De-noising.

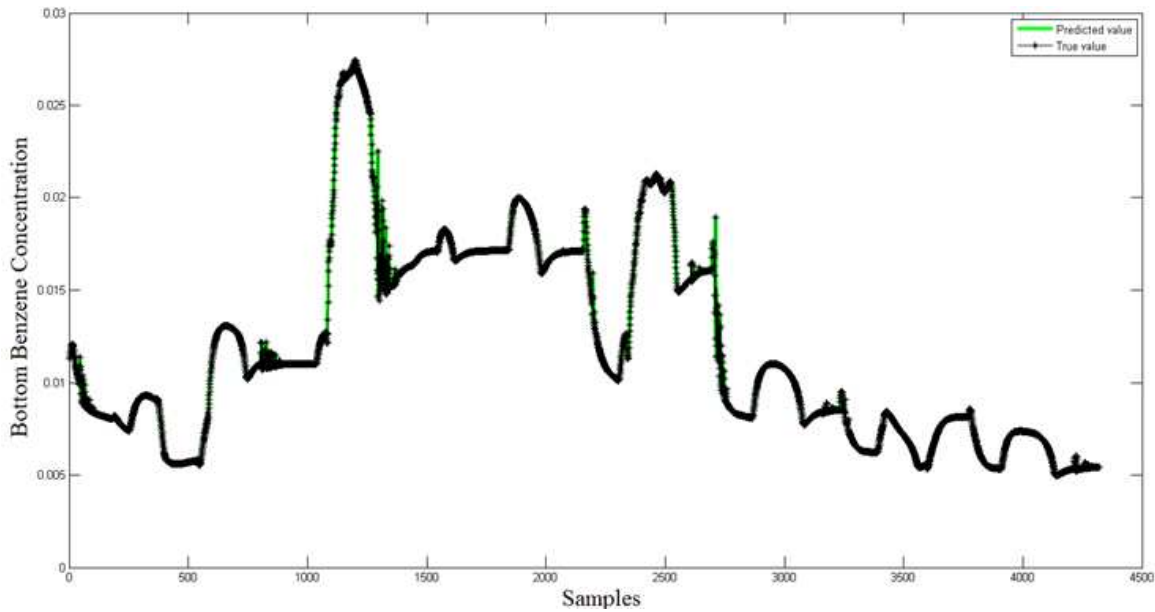


Figure 5. Prediction and true values for Bottom product Benzene concentration using modified distance-based clustering (MNNDC) improved by PSO after noise elimination using Wavelet De-noising.

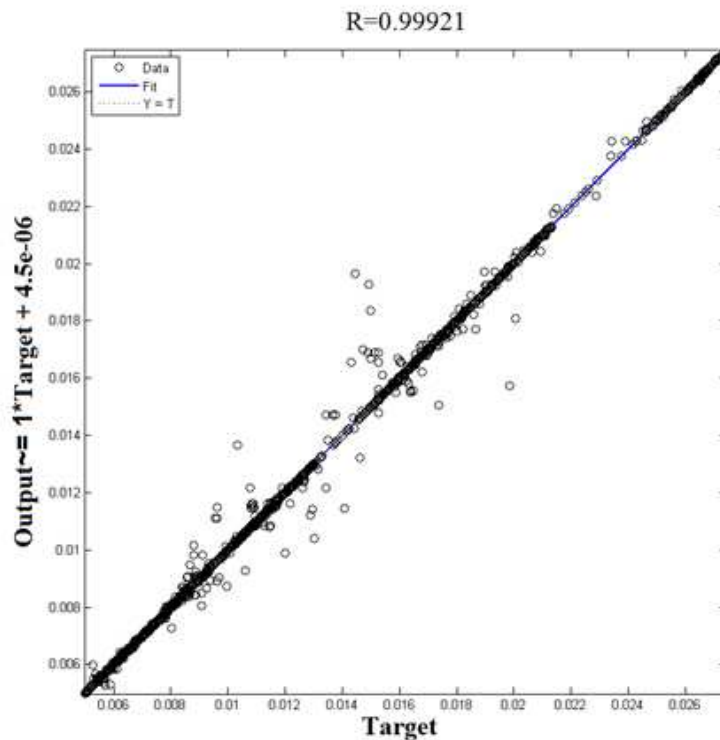


Figure 6. Correlation coefficient value for Bottom Benzene concentration using MNNDC improved by PSO after noise elimination using Wavelet De-noising.

To validate the proposed method Root Mean Square Error (RMSE) and Correlation Coefficient (CC) are used. The RMSE is defined as

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y(est)_i - Y_i)^2} \tag{10}$$

Where $Y(est)_i$ and Y_i are the predicted and real values, respectively. The value of CC is given by

$$CC = \frac{\sum_{i=1}^n (Y(est)_i - \overline{Y(est)}) (Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^n (Y(est)_i - \overline{Y(est)})^2 (Y_i - \overline{Y})^2}} \tag{11}$$

Where $\overline{Y(est)}$ and \overline{Y} are the mean values of $Y(est)$ and Y . The RMSE and CC values of each of inferential sensor is calculated by applying to the test data, calculated from left out of train data (about 30 percent).

As it can be seen in Table 3, the modified nearest neighbor distance-based clustering (MNNDC) algorithm has more

prediction accuracy in comparison with conventional nearest neighbor distance-based clustering (CNNDC) algorithm. In

addition, the number of clusters in MNNDNC approach is smaller.

Table 3. Performance and comparison of conventional and modified distance-based clustering improved by PSO using noisy and de-noised data-set by Wavelet.

Inferential sensor	Noisy Data set			De-noised Data set		
	RMSE	CC	Number of clusters	RMSE	CC	Number of clusters
CNNDC-PSO	0.042	0.9632	75	0.017	0.98925	55
MNNDNC-PSO	0.0182	0.9793	38	0.00933	0.99921	31

4. Conclusion

In this paper, a new approach has been proposed to design a data-driven inferential sensor for estimation of concentration of Benzene in a distillation column. In this approach, a zero-type TSK fuzzy inference system based on modified distance-based clustering optimized by PSO is used. The process data was contaminated by measurement noises and the removed using Wavelet denoising method. This method have several advantages to conventional distance-based clustering and employs PSO as optimization algorithm. In this algorithm the clusters centers are not fixed during training of the model that results in less generation of clusters or rules and faster training in comparison with the conventional type. The simulation proved that proposed algorithm have less rule numbers, whereas prediction accuracy has been increased. Finally, compared to conventional nearest neighbor clustering (CNNDC), it has better generalization abilities, smaller RMSE and larger CC for testing data and acceptable performance when the data-set is contaminated. It can be concluded that the superior performance of proposed algorithm makes it powerful to predict the Benzene concentration quickly and precisely follow the process variations.

References

- [1] Fortuna, L., Graziani, S., Rizzo, A. and Xibilia, M. G., 2007. *Soft sensors for monitoring and control of industrial processes*. Springer Science & Business Media.
- [2] Lee, P. and Dexter, A., 2005. "A fuzzy sensor for measuring the mixed air temperature in air-handling units". *Measurement*, 37 (1), pp. 83-93.
- [3] Ibarguengoytia, P. H., Delgadillo, M. A., García, U. A. and Reyes, A., 2013. "Viscosity virtual sensor to control combustion in fossil fuel power plants". *Engineering Applications of Artificial Intelligence*, 26 (9), pp. 2153-2163.
- [4] Sun, K., Liu, J., Kang, J.-L., Jang, S.-S., Wong, D. S.-H. and Chen, D.-S., 2014. "Development of a variable selection method for soft sensor using artificial neural network and nonnegative garrote". *Journal of Process Control*, 24 (7), pp. 1068-1075.
- [5] Kadlec, P., Gabrys, B. and Strandt, S., 2009. "Data-driven soft sensors in the process industry". *Computers & Chemical Engineering*, 33 (4), pp. 795-814.
- [6] Gholami, A. R. and Shahbazian, M., 2015. "Soft sensor design based on fuzzy C-Means and RFN_SVR for a stripper column". *Journal of Natural Gas Science and Engineering*, 25, pp. 23-29.
- [7] Souza, F. A., Araujo, R. and Mendes, J., 2016. "Review of soft sensor methods for regression applications". *Chemometrics and Intelligent Laboratory Systems*, 152, pp. 69-79.
- [8] Rani, A., Singh, V. and Gupta, J., 2013. "Development of soft sensor for neural network based control of distillation column". *ISA transactions*, 52 (3), pp. 438-449.
- [9] Pan, T.-H., Wong, D. S.-H. and Jang, S.-S., 2010. "Development of a novel soft sensor using a local model network with an adaptive subtractive clustering approach". *Industrial & Engineering Chemistry Research*, 49 (10), pp. 4738-4747.
- [10] Kaneko, H., Arakawa, M. and Funatsu, K., 2009. "Development of a new soft sensor method using independent component analysis and partial least squares". *AIChE Journal*, 55 (1), pp. 87-98.
- [11] Jassar, S., Liao, Z. and Zhao, L., 2009. "Adaptive neuro-fuzzy based inferential sensor model for estimating the average air temperature in space heating systems". *Building and environment*, 44 (8), pp. 1609-1616.
- [12] Zamprognà, E., Barolo, M. and Seborg, D. E., 2005. "Optimal selection of soft sensor inputs for batch distillation columns using principal component analysis". *Journal of process control*, 15 (1), pp. 39-52.
- [13] Jain, P., Rahman, I. and Kulkarni, B., 2007. "Development of a soft sensor for a batch distillation column using support vector regression techniques". *Chemical Engineering Research and Design*, 85 (2), pp. 283-287.
- [14] Yu, J., 2012. "A Bayesian inference based two-stage support vector regression framework for soft sensor development in batch bioprocesses". *Computers & Chemical Engineering*, 41, pp. 134-144.
- [15] Kalos, A., Kordon, A., Smits, G. and Werkmeister, S., 2003. "Hybrid model development methodology for industrial soft sensors". American Control Conference, 2003. Proceedings of the 2003, IEEE, pp. 5417-5422.
- [16] Takagi, T. and Sugeno, M., 1985. "Fuzzy identification of systems and its applications to modeling and control". *IEEE transactions on systems, man, and cybernetics*, (1), pp. 116-132.
- [17] Delgado, M. R., Nagai, E. Y. and de Arruda, L. V. R., 2009. "A neuro-coevolutionary genetic fuzzy system to design soft sensors". *Soft computing*, 13 (5), pp. 481-495.
- [18] Mendes, J., Souza, F., Araújo, R. and Gonçalves, N., 2012. "Genetic fuzzy system for data-driven soft sensors design". *Applied Soft Computing*, 12 (10), pp. 3237-3245.

- [19] Zheng, Y., Fang, H. and Wang, H. O., 2006. "Takagi-Sugeno fuzzy-model-based fault detection for networked control systems with Markov delays". *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 36 (4), pp. 924-929.
- [20] Ma, X.-J. and Sun, Z.-Q., 2000. "Output tracking and regulation of nonlinear system based on Takagi-Sugeno fuzzy model". *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 30 (1), pp. 47-59.
- [21] Rezaee, B. and Zarandi, M. F., 2010. "Data-driven fuzzy modeling for Takagi-Sugeno-Kang fuzzy system". *Information Sciences*, 180 (2), pp. 241-255.
- [22] Juang, C.-F. and Lo, C., 2008. "Zero-order TSK-type fuzzy system learning using a two-phase swarm intelligence algorithm". *Fuzzy Sets and Systems*, 159 (21), pp. 2910-2926.
- [23] Ren, Q., Balazinski, M., Baron, L. and Jemielniak, K., 2008. "Tool condition monitoring using the TSK fuzzy approach based on subtractive clustering method". *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, Springer, pp. 52-61.
- [24] Wang, N. and Yang, Y., 2009. "A fuzzy modeling method via Enhanced Objective Cluster Analysis for designing TSK model". *Expert Systems with Applications*, 36 (10), pp. 12375-12382.
- [25] Wang, L.-X., 1999. *A course in fuzzy systems*. Prentice-Hall press, USA.
- [26] Zadeh, L. A., 1965. "Fuzzy sets". *Information and control*, 8 (3), pp. 338-353.
- [27] Jolliffe, I., 2002. *Principal component analysis*. Wiley Online Library.
- [28] Smith, L. I., 2002. "A tutorial on principal components analysis". *Cornell University, USA*, 51 (52), p. 65.
- [29] Shi, Y. and Eberhart, R. C., 1999. "Empirical study of particle swarm optimization". *Evolutionary Computation*, 1999. CEC 99. Proceedings of the 1999 Congress on, IEEE, pp. 1945-1950.
- [30] Shahbazian, M., Jazayerirad, H. and Ebnali, M., 2014. "ANFIS based identification and control of distillation process". *Journal of Automation and Control*, 2 (2), pp. 49-56.