MODELING OF PROCESS STATES BY USING ARTIFICIAL NEURAL NETWORKS IN A FLUIDIZED BED ENERGY PLANT

M. Liukkonen[1], M. Heikkinen[1], T. Hiltunen[1], E. Hälikkä[2], R. Kuivalainen[2], Y. Hiltunen[1]
[1]University of Kuopio, Finland; [2] Foster Wheeler Power Group, Varkaus, Finland

Corresponding Author: M. Liukkonen, University of Kuopio, Department of Environmental Science
P.O. Box 1627, 70211 Kuopio, Finland; mika.liukkonen@uku.fi

Abstract. The efforts to reduce environmental emissions are nowadays affecting increasingly the production of energy, and higher targets are set for the efficiency of combustion processes. Therefore it is important to develop such data analysis and modeling methods that can respond to these new demands. In this study, the formation of nitrogen oxides (NO_x) in a circulating fluidized bed (CFB) boiler was modeled by using a sub-model -based artificial neural network (ANN) approach. In this approach, the process data are processed first by using a self-organizing map (SOM) and k-means clustering to create subsets representing the separate process states in the boiler. These process states can include for example start-ups, shutdowns, and idle times in addition to the normal process flow. After the determination of process states, a variable selection procedure based on multilayer perceptrons (MLP) is performed to create sub-models and to determine whether different variables are affecting the NO_x formation in the defined process states. The results show that this kind of approach can be a fruitful way to get new information from combustion processes.

1 Introduction

Currently, the efficiency of energy plants is becoming a more important issue because of tightening environmental legislation and increasing fuel costs. One of the main issues in this respect is to minimize the process emissions, which include harmful emission components such as nitrogen oxides (NO_x). Despite the growing environmental issues, the production must be capable of reacting fast to changes in the boiler load and to variations in the fuel quality, which necessitates a faster and more accurate operational control of the process. Due to this, such advanced and intelligent systems for process monitoring and optimization are needed that can respond to these new dynamic demands.

Archived process data resources can be used for the optimization and improvement of productivity. At present, artificial neural networks (ANN) are considered an advantageous way to model process data in a diverse field [1–4]. Many of the past applications have also demonstrated that ANNs can provide an efficient automated method for modeling industrial data [1–4]. For example self-organizing maps (SOM) [4] have been successfully used in many research fields [4–7]. The SOM method offers an efficient means of handling complex and multidimensional data, which is a typical situation in any industrial application. Multilayer perceptrons (MLP) have likewise proved their efficacy in the modeling of industrial processes [2, 8–11]. In this study, we combine these two ANN methods to model and analyze the circulating fluidized bed (CFB) boiler process.

As we have disclosed earlier [6], different states of the fluidized bed combustion process can be extracted from process data. These states can involve for instance start-ups, shut-downs, and idle times in addition to the normal process flow. The behavior of the process may be extremely diverse between these conditions. For example, the quantities of different emission components may vary greatly between the states for the simple reason that oil is used most abundantly at start-ups, and less in normal process conditions. However, the normal process flow can also include different process states, where e.g. the bed temperature is unstable or the steam flow is lower than usual. It is important to learn to identify these states as well, because the behavior of the process can vary also in a smaller scale but still in a way that affects the model accuracy.

Creating sub-models instead of generic models forms an interesting aspect of modern-day process analysis because process states and their sub-models may contain valuable information on the behavior of the process, as our previous findings from the wave soldering and the activated sludge treatment process suggest [7, 12]. The sub-model -based approach is reasonable for instance in cases where certain variables correlate strongly with each other, whereby it seems evident that less perceivable but still important phenomena remain undetected behind the characteristic behavior of the data. Despite their lesser perceptiveness, these phenomena can be important considering e.g. the combustion process and certain events in it. Fortunately, this kind of underlying information can be revealed by identifying different process states and constructing sub-models as presented in this paper.

2 Process and data

The main parts of a typical circulating fluidized bed (CFB) boiler include a combustion chamber, a separator and a return leg for the recirculation of the bed particles. Combustion occurs in a fluidized bed, which is typically a mixture of sand, fuel ash and a matter for capturing the sulfur. The primary combustion air is brought in from the bottom of the chamber to fluidize the bed material. Because CFB boilers require the use of high fluidizing velocities, the bed particles are constantly moving with the flue gases, proceeding through the main combustion chamber into a separator, from which the larger particles are extracted and returned to the combustion chamber. Meanwhile, the finer particles are removed from the flue gases by an electrostatic precipitator or by a bag-house filter located downstream from the boiler's convection section. The large heat capacity of the bed ensures stable combustion, and supporting fuels such as oil or gas are used only during the start-up. The intense turbulence of the circulating fluidized bed facilitates the mixing and combustion of fuel. The typical combustion temperature is between 850 and 900 °C. The raw data are extracted once a month from databases of utility scale CFB boilers and the time resolution of the data set is typically 15 minutes. The size of the data matrix used as an example is 10 000 x 42 (10 000 rows, 42 variables in columns).

3 Methods

3.1 Self-organizing maps (SOM)

Kohonen's self-organizing map [4] is a well-known unsupervised learning algorithm. One of the main applications of SOM is to facilitate data analysis by mapping *n*-dimensional input vectors to the neurons for example in a two-dimensional lattice (map). On the map, the input vectors with common features arrive in the same or neighboring neurons. This way the topological order of the original input data is preserved. The lattice of arranged neurons reflects variations in the statistics of the data set and chooses common features, which approximate to the distribution of the data points. Each neuron includes an *n*-dimensional reference vector (prototype vector), which describes the common properties of the neuron. The array of neurons (the map), can be illustrated as a rectangular, hexagonal, or even irregular organization. The size of the map can be altered depending on the application; the more neurons, the more details are represented.

The SOM analysis is based on unsupervised learning. At first, random values for the preliminary reference vectors are sampled from an even distribution whose limits are defined by the input data. During learning the input vectors are categorized one by one into particular neurons (best matching unit, BMU) based on the smallest *n*-dimensional distance between the input vector and the reference vectors. The nearest neighbors of the activated neuron are also activated according to a neighborhood function (e.g. Gaussian distribution) that is dependent on the network topology. At the final stage, the reference vectors of all activated neurons are updated.

3.2 K-means

The k-means method is a widely-used non-hierarchical clustering algorithm [13]. The basic version of the algorithm is started by randomly defining *k* cluster centers, and directing each data point to the cluster whose mean value is closest in the Euclidean-distances-sense. At the next step, the mean vectors of the data points included to each cluster are calculated and used as new centers in an iterative approach. The optimal number of clusters can be determined e.g. by using the Davies-Bouldin -index [14]. Small values of DB-index correspond to clusters whose centers are far from each other, so the optimal number of clusters is the number where the index reaches its minimum. This eliminates the need for knowing the clusters *a priori*.

3.3 Multilayer perceptrons (MLP)

Multilayer perceptrons are well-known feed-forward neural networks [1–2], which consist of processing elements called neurons, and connections. The neurons are arranged in three or more layers: an input layer, one or more hidden layers, and an output layer. A MLP network is trained with data samples, which leads to a supervised learning procedure. The network input signals are processed forward through successive layers of neurons on a layer-by-layer basis. At the first phase, the input layer distributes the inputs to the first hidden layer. Next, the hidden neurons summarize the inputs based on predefined weights, which either weaken or strengthen the effect of each input. The weights are determined by learning from examples (i.e. data samples), which is called supervised learning. Eventually, the inputs are processed by a transfer function, and the result is transferred as a linear combination to the next layer, which is generally the output layer. The performance of the model is then evaluated with an independent validation data set.

MLP neural networks must be trained to the problem concerned. A popular MLP training technique is the back-propagation algorithm [15]. In back-propagation the output values are compared with the proper answer from the original data to calculate the value for a predefined error function. Eventually the iterative training procedure defines a set of weights which minimizes the error between the actual and expected outputs for all input patterns.

3.4 Determination of process states

At the first stage, all the input values were variance scaled. Next, the raw data were coded into inputs for a self-organizing network, and a SOM having 384 neurons in a 24x16 hexagonal arrangement was constructed by using experimentally determined parameters. The linear initialization and batch training algorithm were used in the training of the map, and the neighborhood function was Gaussian. The map was taught with 10 epochs, and the initial neighborhood had the value of 6. The SOM Toolbox (http://www.cis.hut.fi/projects/somtoolbox/) was used in the analysis under a Matlab (version 7.6) software (Mathworks Inc., Natick, MA, USA, 2008) platform. The k-means algorithm was used to the clustering of the trained map or, more precisely, to the clustering of the SOM reference vectors. The cluster information was then combined with process knowledge to identify the process states indicated by clusters.

3.5 Construction of sub-models

To select variables within process states, a variable selection procedure was implemented by using a MLP network with a back-propagation algorithm. The purpose of variable selection was to find the most important factors affecting the formation of NO_x in different process states. The data was divided into three subsets, namely training, training test and validation sets. The training data set, being 60 % of the total 1073 samples was used for training the network. In addition, 20 % of the data set was put to the separate test set to be used in the back-propagation error calculations. The validation data set (the remaining 20 % of the samples) was used as an independent test set for testing the model. The artificial neural network consisted of the process parameters as inputs, one hidden layer with 15 hidden neurons and the output neuron describing the predictable NO_x-concentration. The parameters of the neural network were decided experimentally. The hyperbolic tangent sigmoid (*tansig*) transfer function was used for the hidden layer, and the linear (*purelin*) transfer function for the output layer, whereas the resilient back-propagation (*trainrp*) algorithm and the mean squared error function (*mse*) were exploited in the training procedure. Matlab (version 7.6) software with the Neural Network Toolbox (version 6.0) was used in data processing.

3.6 Selection of variables

The variables were brought one by one to the input layer of the neural network, using the NO_x concentration as an output. To reduce the impact of statistical variation on the results, the training of the network was done in five rounds, calculating the model goodness every round for each added variable. Within each round, the same data set was used in the training of all the parallel models. However, the training data set was chosen differently in every separate round, which reduces the impact of statistical contingency. After this, the mean model goodness value and the standard deviation of all rounds were calculated to rank the variables. In other words, several parallel MLP models were created, from which the best was always selected to be developed in the next selection round. The influence of each selected variable was then added to the total cumulative impact to get a visualization of the evolving total effect. The presented variable selection procedure was performed first for the entire process data, and then for each process state separately to reveal the possible increase of model goodness within process states. Index of agreement [16] (IA), or Willmott's index, was used as the model goodness measure.

In addition, the selection of variables was performed by using multiple linear regression (MLR) [17–18] to validate the results. Except for the modeling method the selection procedure was essentially the same as in the MLP method: the variables were selected by creating parallel models and always picking the best of them, keeping the best model as a basis for the next selection round until all the variables were eventually selected.

4 Results and discussion

The self-organizing map was obtained by training a self-organizing network using the fluidized bed boiler data as inputs. After clustering, the process states were determined by using process knowledge. The process states are identified on the SOM in Fig. 1. The performances of the NO_x simulation sub-models based on the corresponding process states, and the first ten selected variables of those models can be seen in Table I. The goodness of these sub-models varies from 0.97 to 0.98, as measured by the index of agreement (IA). In comparison with the generic process model including all the process data (IA ca. 0.95), these values are 2–3 % greater. This suggests that dividing the overall process model into separate process state models clearly improves the model goodness.

Furthermore, there was a remarkable congruence between variables selected by the nonlinear MLP method and the linear method (MLR). First of all, the first ten variables selected by both of these methods were essentially the same within each process state; the remaining differences between the models can be explained by nonlinearity. However, the overall goodness varied between 0.8–0.9 in the linear sub-models, so the value added by the MLP method (IA ca. 0.97–0.98) is outstanding.

Generally speaking, there seems to be three kinds of process variables. Firstly, some of them seem to be dominating variables, i.e. they have a great impact on the NO_x formation in every process state. These include, for instance, the fluidized bed temperature and pressure. On the other hand, certain variables seem to affect the NO_x

formation only in certain process states. These variables seem to vary with each state of process, as can be seen in Table I. Thirdly, in addition to the affecting variables there are also such variables that have no great impact on the output variable in any of the process states.

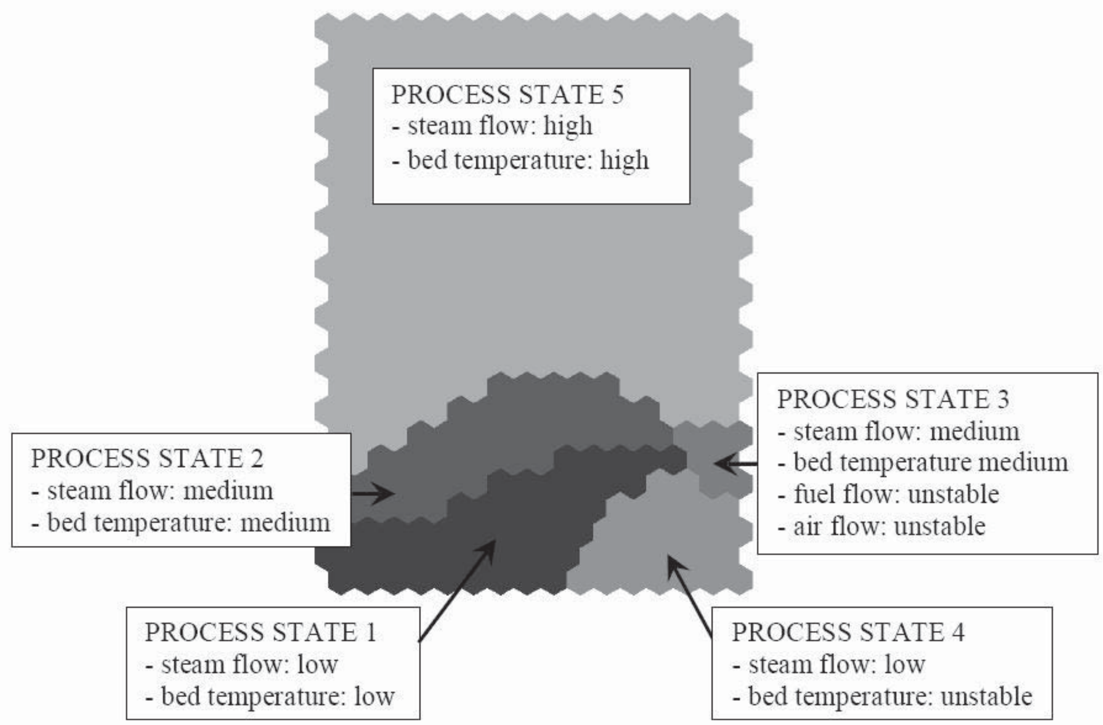

Fig. 1. SOM with identified process states.

The schematic presentation of the method used is shown in Fig. 2. At first the raw data are pre-processed. Preprocessing includes all the necessary actions, such as the rough selection and normalization of variables, to process the data to a suitable form for modeling. Next, the SOM and k-means clustering are used and combined with expert process knowledge to obtain and identify the different process states. Finally, the modeling is performed within process states to get process sub-models.

The approach presented is useful because the main model may involve interesting phenomena that are hidden behind the dominating behavior of the data. These hidden phenomena are not necessarily perceivable without creating sub-models, but can still have significant effects on the process. For example in this case the accuracy of the NO_x simulations improved by creating process states and their sub-models. In this respect, the results confirmed our previous findings [7, 12] concerning the fact that process states may involve important information on a process.

The method presented in this paper facilitates data analysis and can be used to define process states and to create sub-models within those states. Especially, the method is advantageous when dealing with a large number of process variables, because in such cases the different data processing stages can be laborious if more conventional methods are used in the analysis. The classification of data samples into different categories, which can be identified as process states by using expert knowledge, seems to provide extra accuracy to the models. In addition, the ability of the method to reveal nonlinear and multivariate interactions can bring additional value to the models. For these reasons, the presented data analysis method offers a fruitful option to identify the different process states and their corresponding sub-models.

PROCESS STATE 1		PROCESS STATE 2	
FLUIDIZED BED PRESS AVE	0,865	FLUIDIZED BED TEMP AVE	0,902
SA FLOW TO LEFTRIGHT4 AVE	0,881	SECONDARY AIR FLOW AVE	0,928
Coal conveyor speed AVE	0,891	FLUIDIZED BED PRESS AVE	0,947
FLUIDIZED BED TEMP AVE	0,946	FG O2 AFT ECO1TO4 AVE	0,961
FLUIDIZED BED TEMP STD	0,954	PRIMARY AIR FLOW AVE	0,970
FG O2 AFT ECO1TO4 AVE	0,956	FG O2 BEF STACK	0,974
FG TEMP AFT SEPTR STD	0,957	FURN LWR LVL TEMP STD	0,969
FURN LWR LVL TEMP STD	0,962	FURN MDL LVL TEMP AVE	0,975
PA FLOW TO WINDBOX AVE	0,962	FURN MDL LVL TEMP STD	0,973
FLUIDIZED BED PRESS STD	0,971	OIL RTN FLOW AVE	0,979
PROCESS STATE 3		**PROCESS STATE 4**	
FLUIDIZED BED TEMP AVE	0,935	FLUIDIZED BED PRESS AVE	0,774
PRIMARY AIR FLOW STD	0,982	FLUIDIZED BED TEMP AVE	0,896
FG O2 AFT ECO1TO4 AVE	0,983	FG O2 AFT ECO1TO4 AVE	0,950
FURN MDL LVL TEMP STD	0,983	Coal conveyor speed AVE	0,970
FG TEMP AFT SEPTR STD	0,971	PRIMARY AIR FLOW AVE	0,971
OIL FLOW BEF BRN STD	0,979	PRIMARY AIR FLOW STD	0,972
SA FLOW TO LEFTRIGHT6 AVE	0,971	OIL RTN FLOW AVE	0,975
SA FLOW TO LEFTRIGHT4 AVE	0,966	SECONDARY AIR FLOW STD	0,975
SA FLOW TO LEFTRIGHT4 STD	0,972	PA FLOW TO FUEL FEED STD	0,978
PRIMARY AIR FLOW AVE	0,980	FURN LWR LVL TEMP STD	0,976
PROCESS STATE 5		**Abbreviations:**	
FLUIDIZED BED TEMP AVE	0,921	BRN = burner	AFT = after
PA FLOW TO WINDBOX AVE	0,942	ECO = economizer	BEF = before
Coal conveyor speed AVE	0,952	FG = flue gas	LVL = level
FLUIDIZED BED TEMP STD	0,962	FURN = furnace	LWR = lower
FG O2 AFT ECO1TO4 STD	0,964	PA = primary air	MDL = middle
SA FLOW TO S/U BNR STD	0,965	PRESS = pressure	
FURN MDL LVL TEMP STD	0,968	SA = secondary air	
OIL FLOW BEF BRN AVE	0,968	SEPTR = separator	
FLUIDIZED BED PRESS AVE	0,970	TEMP = temperature	
FURN LWR LVL TEMP STD	0,971		

Table I. Process state -specific models obtained by MLP variable selection. The numbers indicate the development of the model performance with each added variable as a means of the IA. For practical reasons only the ten first of the selected variables are shown. 'AVE' refers to an averaged variable and 'STD' to standard deviation.

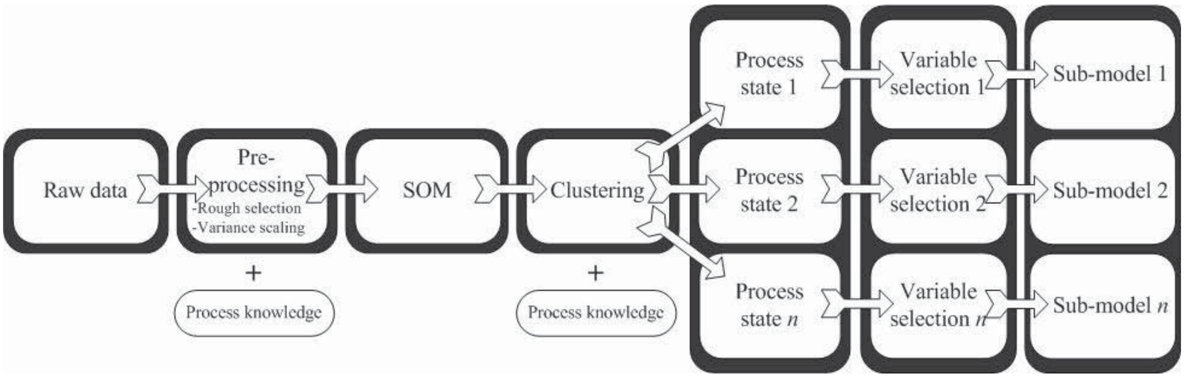

Fig. 2. Schematic presentation of the method used.

5 Conclusion

Because of the growing need for optimizing industrial processes due to, for example, increasing environmental issues, developing new methods for process analysis is important. The results presented in this paper show that the artificial neural network method used is an efficient and fruitful way to model the circulating fluidized bed process and to simulate the process emissions.

6 References

[1] S. Haykin. Neural Networks: A Comprehensive Foundation. Prentice Hall, Upper Saddle River, NJ, 1999.

[2] M. R. G. Meireles, P. E. M. Almeida, and M. G. Simões. A Comprehensive Review for Industrial Applicability of Artificial Neural Networks. IEEE Trans. Industrial Electronics, 2003, 50(3): 585–601.

[3] I. M. Mujtaba, and M. A. Hussain, Eds. Application of Neural Network and Other Learning Technologies in Process Engineering. Imperial College Press, London, UK, 2001.

[4] T. Kohonen. Self-Organizing Maps, 3rd ed. Springer-Verlag, Berlin Heidelberg New York, 2001.

[5] M. Heikkinen, M. Kolehmainen, and Y. Hiltunen. Classification of process phases using Self-Organizing Maps and Sammon's mapping for investigating activated sludge treatment plant in a pulp mill. In: Proc. Fourth European Symposium on Intelligent Technologies and their implementation on Smart Adaptive Systems, 2004, 281–297.

[6] M. Heikkinen, A. Kettunen, E. Niemitalo, R. Kuivalainen, and Y. Hiltunen. SOM-based method for process state monitoring and optimization in fluidized bed energy plant. In: Lecture Notes in Computer Science 3696 (Eds.: Duch, W., Kacprzyk, J., Oja, E., Zadrożny, S.), Springer-Verlag Berlin Heidelberg, 2005, 409–414.

[7] M. Liukkonen, E. Havia, H. Leinonen, and Y. Hiltunen. Application of Self-Organizing Maps in Analysis of Wave Soldering Process. Expert Systems with Applications, accepted for publication.

[8] R. D. Reed, and R. J. Marks II. Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks. MIT Press, Cambridge, Massachusetts, 1999.

[9] M. Heikkinen, V. Nurminen, T. Hiltunen, and Y. Hiltunen. A Modeling and Optimization Tool for the Expandable Polystyrene Batch Process. Chemical Product and Process Modeling, 2008, 3(1), article 3.

[10] M. Liukkonen, T. Hiltunen, E. Havia, H. Leinonen, and Y. Hiltunen. Artificial Neural Network based Modeling of Solder Bridging in Wave Soldering Process. Engineering Applications of Artificial Intelligence, submitted for publication.

[11] M. Liukkonen, T. Hiltunen, E. Havia, H. Leinonen, and Y. Hiltunen. Modeling of Soldering Quality by Using Artificial Neural Networks. IEEE Trans. Electronics Packaging Manufacturing, accepted for publication.

[12] M. Heikkinen, T. Heikkinen, and Y. Hiltunen. Process States and Their Submodels Using Self-Organizing Maps in an Activated Sludge Treatment Plant. In: Proc. 48th Scandinavian Conference on Simulation and Modeling (SIMS 2008) [CD-ROM], Oslo University College, 2008.

[13] J. MacQueen. Some methods for classification and analysis of multivariate observations. In: Proc. Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume I: Statistics, University of California Press, Berkeley and Los Angeles, 1967, 281–297.

[14] D.L. Davies and D.W. Bouldin: A cluster separation measure. IEEE Trans. Pattern Recognition and Machine Intelligence, 1979, 1(2):224–227.

[15] P. J. Werbos. The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting. John Wiley & Sons, New York, 1994.

[16] C. J. Willmott. On the validation of models. Physical Geography, 1981, 2, 184–194.

[17] S. Chatterjee and A. S. Hadi. Influential Observations, High Leverage Points, and Out-liers in Linear Regression. Statistical Science, 1986, 1(3):379–393.

[18] J. Cohen. Multiple Regression as a General Data-Analytic System. Psychological Bulletin, 1968, 70(6):426–443.