INFOTEH-JAHORINA Vol. 13, March 2014.

Implementation of a Neural Network Based Soft Sensor

Nikola S. Popov, Željko Tepić, Darko Stanišić, Slađana Lazarević
Department for Systems, Signals and Control,
Faculty of technical sciences, University of Novi Sad
Novi Sad, Serbia
npopov@uns.ac.rs, tepic@uns.ac.rs, darkos@uns.ac.rs, sladjana.lazarevic@uns.ac.rs

Abstract — **In this paper it is presented systematic approach for implementation of neural network soft sensor. Implemented soft sensors are used in cement industry for monitoring free CaO in clinker and cement fineness but application presented in this paper can be used for modeling any industrial process. Feed forward neural networks are used for prediction. Soft sensors are implemented using .NET technology. Server developed in this paper periodically obtains data from OPC servers and uses it for estimation. Real time estimation and user interface are divided in two applications which increases robustness and allows multiple users to configure server.**

Key words - Soft sensor; OPC technology; Neural network; .NET;

I. Introduction

In this paper it is presented systematic approach for real time soft sensor implementation. Real time soft sensor is used to estimate immeasurable variables in cement plant. In this paper two soft sensors are implemented: estimator for free CaO prediction in clinker and estimator for cement fineness. Free CaO and cement fineness are main indicators for cement quality and process stability [1]. They are measured in laboratory every few hours and information about those values don't exist between two measurements. Because of that its values can't be used in closed loop systems. Also, laboratory analysis last 20 minutes which produces additional delay in system.

Soft sensor [2]-[5] is implemented using .NET technology and contains three main components: Server application, Client application and database. Client application is used to configure server applications. Server application is core of the system and it is used to communicate with OPC Server, store data, estimate variables using neural network, retrain neural network etc. Server contains software components: module for communication with OPC Server, module for communication with database, module for signal, module for neural networks [6], module for neural network retraining and control module.

II. Method

A. Server Application

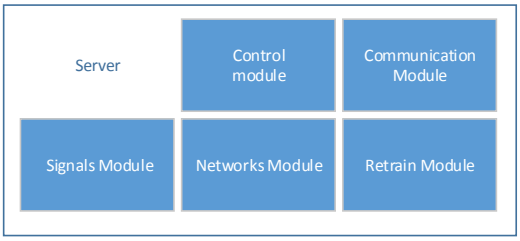

Figure 1 – Server components

Server is implemented as Windows service and contains 5 main modules shown in Fig. 1. Control module is main module and triggers other modules.

Algorithm sample time is one minute, but signals are sampled every 10 seconds and average value is used as network input. Every minute 6 samples are obtained and every sample has its value and time when signals is updated on server. Time when signal is updated on server varies during time. Because of that, average value is calculated as integral of signal collected in that minute divided with time between first and last collected signal. This is first filter which eliminates measurement's noise but it doesn't eliminate outliers, which is done later. Sever can communicate with multiple OPC Servers in same time and networks can have inputs from varies OPC Servers. Also it is possible to configure new signals which are combination of two existing signals.

On every minute new thread is started which uses copy of averaged data collected in that minute and sends them to Signals module which inserts them in database and save them in local Dictionary for faster processing. Every OPC server has watchdog tag which is toggled. After that copy of all networks is obtained and processed. Every network collects necessary data from Signals module. If every collected signal is between defined limits, network output is calculated and sent to OPC server. Also every network has its watchdog signal which is toggled. This is indication for later algorithms that network output is calculated correctly and can be used for advanced control.

Every second in minute new thread is started that controls network retrain. Every network can be automatically

This research was supported by the Ministry of Education and Science of the Republic of Serbia (Project no. III-41007)

retrained if network error in defined period is larger than defined limit or periodically on defined period. If retrain is necessary and network is not already in retraining process new thread for retrain is started. New thread is started because of the possibility that networks can have large configuration and in that case retrain process can last for hours or small configuration with fast retrain process that is carried out more often. In every retrain, network is retrained 10 times and network with smallest error is used as new network. User can define whether new or old weights are used for initialization.

Every ten minutes new thread for cleaning unused data is started. This thread will delete data older than 6 months from database, data older than 3 hours from local memory and unused networks from database.

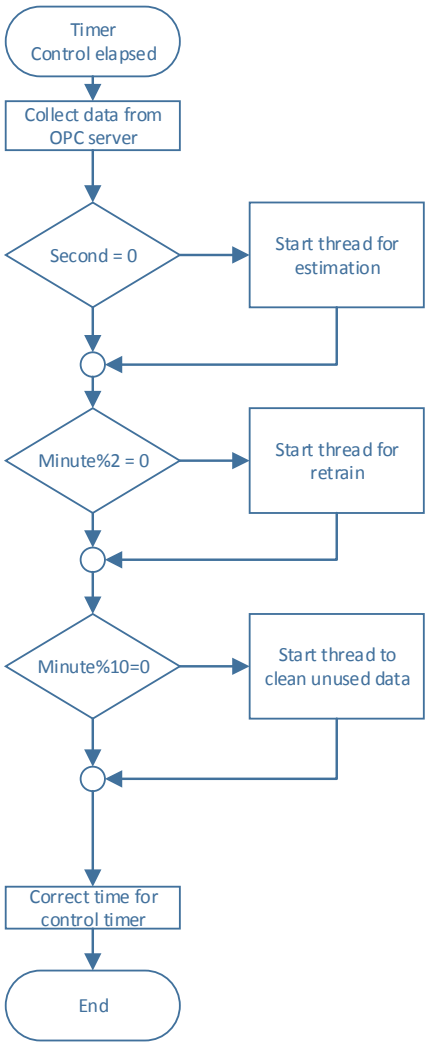

Figure 2 – Main control loop

B. *Client Application*

Client application is implemented using .NET and Windows Presentations Foundation (WPF). Application communicates with server using Windows Communication Foundation (WCF). Application can be divided in three main components: Signal configuration, Network configuration and Retrain configuration.

In Signal configuration (Fig. 3) user can define which signals will be collected from OPC servers. User can also analyze signals by viewing historical data and monitor real time signal changes.

Network configuration is used to define neural networks. User can define new network, copy network, delete network, analyze new network history, monitor network output, define OPC server tag where network output is written, and view network structure. When defining new network user can define network inputs, network output, interpolation between two laboratory analyses and duration of laboratory analysis (Fig. 4, Fig. 5). For every signal network can have one value as network input, sequence of values as network input, or average value of sequence of values as network input. User must define signals limits, delay, windows size, and filter type.

In Train section user defines period of time to collect data for train (Fig. 6). Data which are not within limits range will be dropped. User can define amount of data used for training and for validation, network structure, and monitor train process. Networks are initialized using genetic algorithm and trained using Levenberg -Marquardt method. User can monitor network output after every iteration and stop training (Fig. 7).

In Retrain section user can define criteria for network retrain and amount of data used for retrain. Network can be retrained periodically or after network error is larger than defined limit. After retrain, network will have weights same as in iteration when error on test data was smallest. Also only if old network has larger error on test data it will be replaced with new network. User can monitor when network began retraining, how long it lasted, number of iteration, and error after retrain.

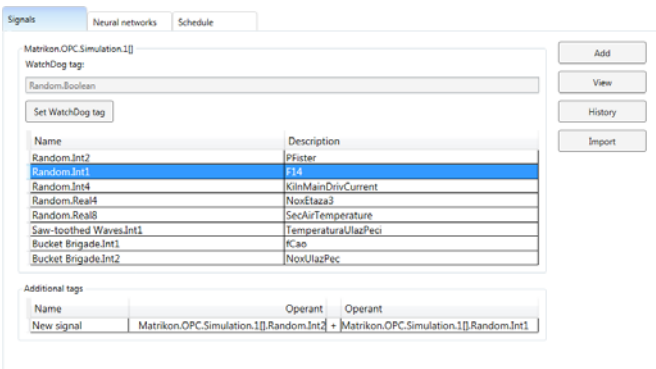

Figure 3 – Signals

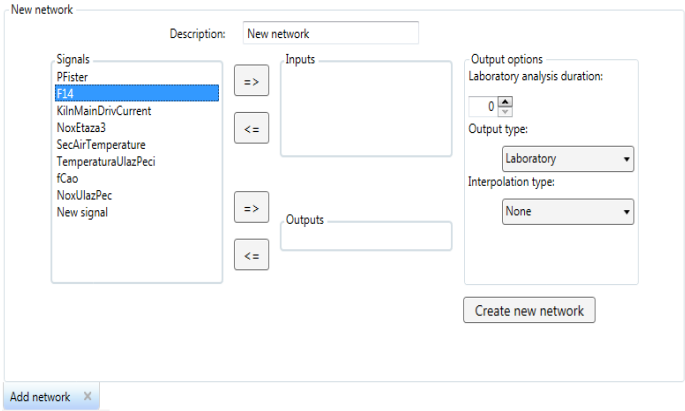

Figure 4 – Definition of input and output signals for neural network

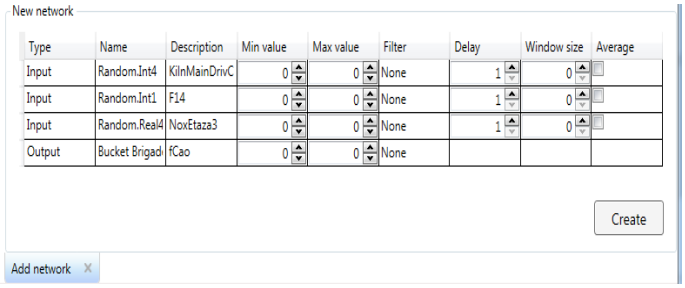

Figure 5 - Network parameters

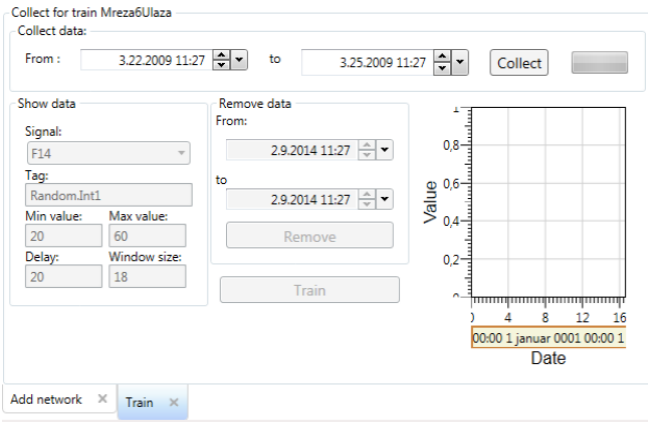

Figure 6 - Collect data for train

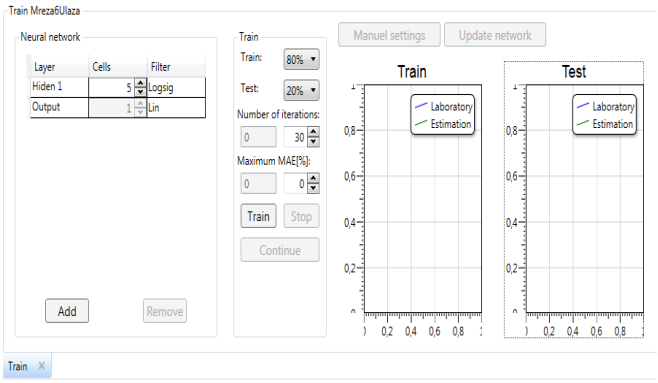

Figure 7 - Neural network training

C. *Estimation of Free CaO in Clinker*

Clinker production plant can be divided in three parts: preheating, cooking and fast cooling. Preheating is used to remove moist from raw meal and for decomposition of Magnesium carbonate (MgCO3). In large rotary kiln raw meal is turned into hot lava. In coolers lava falls on moving grid which turn lava into small pieces. The final product was obtained by cooling these pieces. Stability of process is determent by number of free CaO in clinker [7]. This value is measured in laboratory hourly and because of it cannot be used in control loops. Because input signals are collected every minute it is necessary to interpolate free CaO value between two measurements using B-spline interpolation. This interpolation will save process dynamics and provide necessary data for neural network training. Also over 1000 values are measured and acquired during this process. Soft sensor for free CaO estimation in this paper is based on moving average neural network because this model represents nonlinear moving average model configured with concern on signal delays and with assumption that appropriate results can be obtained with small number of past values for every input signals. Because process changes during time it is necessary to retrain neural network after every new laboratory analysis.

D. *Cement Fineness*

The cement grinding circuit is the final stage in the cement production process. In this stage clinker from kiln is milled to certain fineness. Cement fineness is direct indicator of the cement quality. Because laboratory cement fineness is measured in laboratory there is no direct measurement operator who control system must large amount of energy to ensure cement quality. Laboratory analysis is done hourly and for every analysis cement is collected 6 times every 6 minutes. Measured value represents average fineness in last 36 minutes. Because of it model developed in this paper must have inputs have average values of signals in that period. Model developed and implemented in plant is based on feed forward neural network. Because process changes during time it is necessary to retrain neural network after every new laboratory analysis.

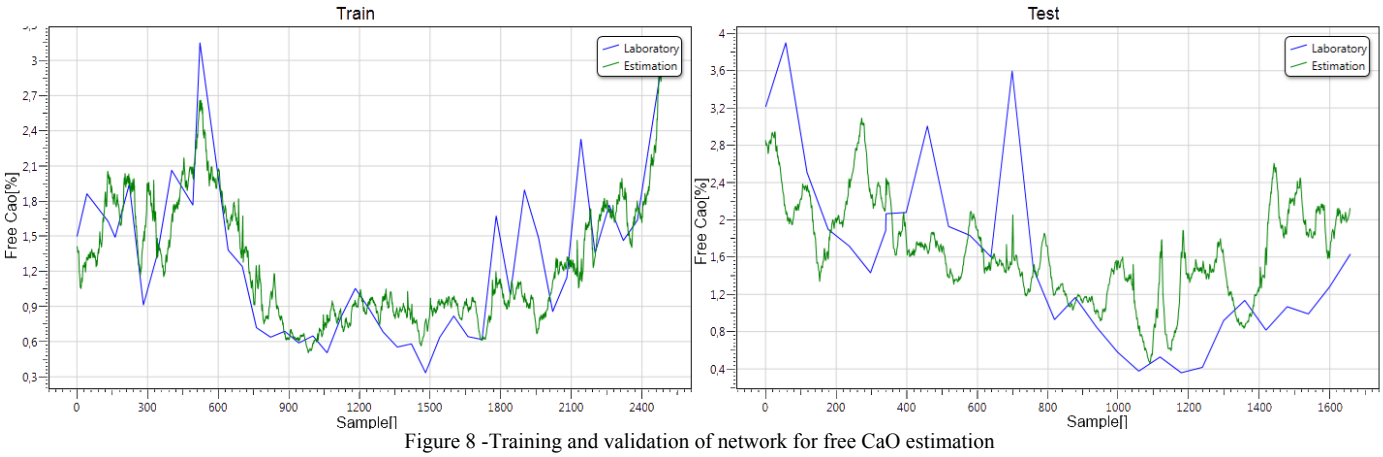

Figure 8 -Training and validation of network for free CaO estimation

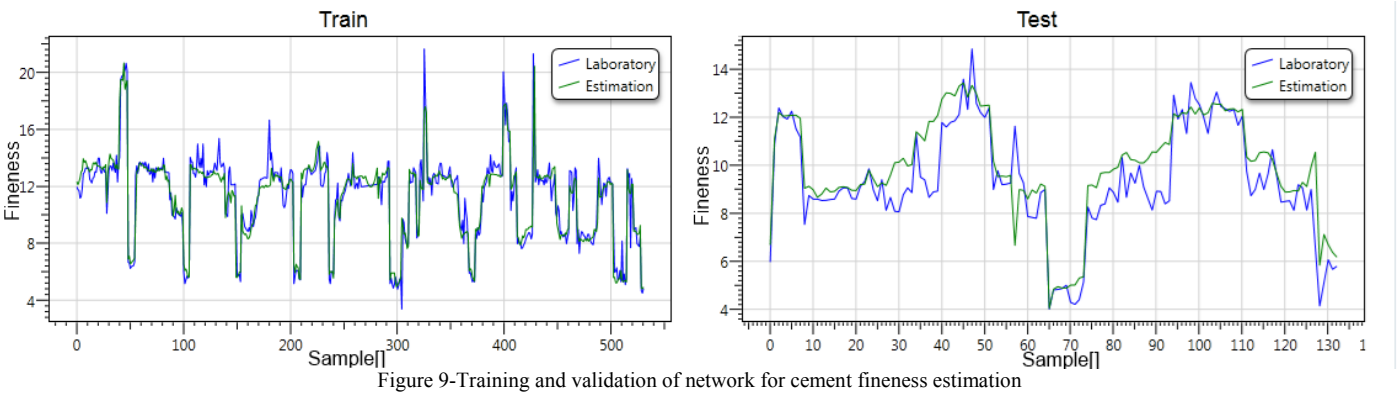

Figure 9-Training and validation of network for cement fineness estimation

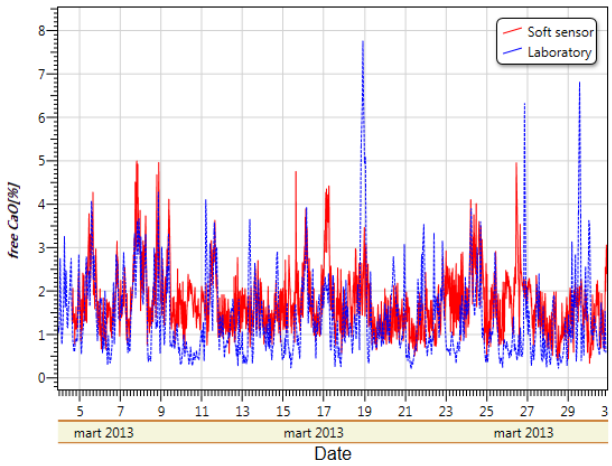

Figure 10-Network prediction for free CaO

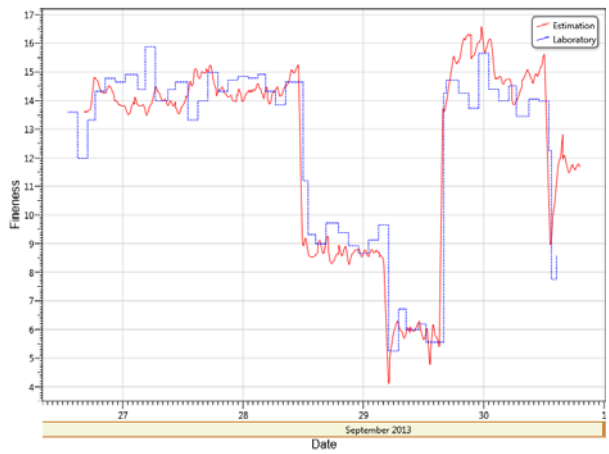

Figure 11- Network prediction for cement fineness

III. RESULTS AND DISCUSSION

A. Free CaO in Clinker

In order to implement free CaO soft sensor one must select proper inputs. Signals chosen as model inputs were: returned air temperature, kiln motor driven current, new material at kiln input, amount of NO and NO_2 at kiln input, temperature at kiln input and clinker cooler fan current. Every signal is collected during different parts of process and its delay varies. Because of it model forms input data with 5 minute radius around delay on every input signal. For initial network training one week data were used. We used 60% of data for training and 40% for validation. Developed neural network has one hidden layer with 2 neurons in it. Results of initial training are shown in Fig. 8. Retraining is done periodically every two hours. Network prediction is shown in Fig.10.

B. *Cement Fineness*

Signals chosen as model inputs were: Amount of clinker in mill input, level of material in mill, temperature of cement at mill output, difference between pressure on input and output of separator, material at feed return to mill, mill current and mill speed. Laboratory analysis provides mean value of fineness for the previous hour. Also every signal is collected during different parts of process and because of it every signal have delay. For initial network training 500 values of cement fineness were used. We used 70% of data for training and 30% for validation. Developed neural network has one hidden layer with 6 neurons in it. Results of training are shown in Fig. 9. Retraining is done periodically every two hours. Network prediction is shown in Fig. 11.

IV. CONCLUSION

This paper shows the possibilities of applying different types of models based on neural networks for prediction laboratory values and their implementation. Developed system for soft sensor estimation has robust implementation and provides simple user interface. User can easily configure models and implement them in plant. To test this system two models were implemented. Both models have satisfactory results and its estimation can be used for control in real time systems, and can be used for early fault detection. Using soft sensors plant consumes less energy for clinker and cement production. Early detection is very useful because operator can detect problem in system when it occurs, not after laboratory analysis and his reaction is faster and less radical. Also operator can see system reaction on changed control, and doesn't have to wait for next analysis.

ACKNOWLEDGMENT

Authors gratefully acknowledge support from Lafarge BFC, Beočin.

REFERENCES

[1] J.I. Bhatty, F.M. Miller, and S.H. Kosmatka, “Innovations in portland cement manufacturing,” Skokie, IL, United States of America: Portland Cement Association, 2004.

[2] D. Sbarbaroa, P. Ascencioa, P. Espinozab, F. Mujicab, and G. Cortes, “Adaptive soft-sensors for on-line particle size estimation in wet grinding circuits,” Control Engineering Practice, vol. 16, pp. 171-178, 2008.

[3] A. Casali,G. Gonzalezb, F. Torresc, G. Vallebuonaa, L. Castellic, P. Gimenez, “Particle size distribution soft-sensor for a grinding circuit,” Powder Technology, vol. 99, pp. 15-21, 1998

[4] A. Rani, V. Singh, and J.R.P. Gupta, “Development of soft sensor for neural network based control of distillation column," ISA Transactions, vol. 52, pp. 438-449, 2013.

[5] E. Etien, “Modeling and simulation of soft sensor design for real-time speed estimation, measurement and control of induction motor,” ISA Transactions, vol. 52, pp. 358-364, 2013.

[6] V. Kecman, “Learning and soft computing,” The MIT Press, 2004.

[7] M. Sadeghian, F. Alireza, “Identification, prediction and detection of the process fault in a cement rotary kiln by locally linear neuro-fuzzy technique,” World Academy of Science, Engineering and Technology, vol. 34, pp. 1109-1115, 2009.