

A TOOLBOX FOR ON-LINE PROCESS MONITORING WITH SOME INDUSTRIAL APPLICATIONS

S-L. Jämsä-Jounela, M. Vermasvuori, S. Haavisto, P. Endén

*Laboratory of Process Control and Automation, Helsinki University of Technology,
P.O.Box 6100, FIN-02015 HUT, Finland*

Abstract: On-line process monitoring with fault detection can provide stability and efficiency for a wide range of processes. A toolbox for on-line monitoring using Kohonen self-organizing maps (SOM), in conjunction with heuristic rules is described in this paper. Four different industrial applications using the toolbox are presented and discussed at the end of the paper. *Copyright © 2002 IFAC*

Keywords: Neural networks, rule based systems, fault detection computer software, computer application.

1. INTRODUCTION

Advanced methods of fault detection and diagnosis have been developed. A number of approaches using mathematical models for fault detection have been developed during the last 20 years (Isermann, 1997; Himmelblau, 1978). The idea is to generate signals that reflect inconsistencies between nominal and faulty system operation. The conventional method has been to use static and dynamic models of the process, the faults appearing as parameter or state changes caused by malfunctions of the components. Parameter or state changes are determined using estimation techniques (Isermann, 1993). Rule-based expert systems for fault detection and diagnosis problems have been investigated very intensively. Fault diagnosis based on rule-based expert systems has an extensive database of rules (Kramer, 1987). Some researchers have successfully combined parameter estimation or observers with fuzzy logic (Frank and Kuipel, 1993; Isermann, 1994) in order to be able to combine symbolic knowledge with the quantitative information, thereby minimizing the false alarm rate (Patton et al., 2000).

Process monitoring and analysis through statistical modelling techniques and neural networks have received considerable attention in recent years. The general objectives of process monitoring are to detect any abnormal event, reduce off-specification production and provide early warnings and identify important process disturbances, malfunctions or faults. Process analysis by statistical or neural network methods promotes understanding of the process without the task of physical modelling, and ultimately improves plant performance. A neural network algorithm, the Self-Organizing Map (SOM), developed by Teuvo Kohonen (Kohonen, 1990), has

been successfully used in improving the geometric quality parameters of hot rolled strips as described in (Cser et al. 1998). Deventer et al. (1996) used SOM in visualising flotation process disturbances, and Ahola et al. (1999) in monitoring a continuous pulp digester.

The software tool presented in this paper utilizes the SOMs and states of the measurements, in conjunction with heuristic rules of the inference engine, in order to monitor the process online. The purpose of the fault diagnosis system is to detect abnormal process states and to inform the plant operator accordingly. A process deviation from the normal operating range is often caused by equipment malfunctions or an incorrect control strategy. Early detection of undesirable process states enables the execution of correcting actions, thus minimizing the damage caused by process malfunctions.

This paper first outlines the theory behind the techniques employed by the fault diagnosis system, the structure of the toolbox is then described, and finally applications and results from industrial implementations and testing periods are presented and discussed.

2. THE KOHONEN SELF-ORGANIZING MAPS

The Self-Organizing Map (SOM) is a neural network algorithm developed by Professor Teuvo Kohonen that forms a two-dimensional presentation from multi-dimensional data. The topology of the data is kept in the presentation such that data vectors that closely resemble one another are located next to each other on the map. Another important characteristic of the SOM is generalization of the information, thus

enabling classification of the data vectors not used in training the SOM. The SOMs are applied to classify large amounts of data. They can be used to form a neural network model of an unknown system based only on the data received from the system. In contrast to traditional methods, such as principal component analysis, the Kohonen model can also be created from highly deviating, non-linear data.

Before the SOMs are trained, values of each variable in the data should be normalized to have a zero mean and a variance of one. This procedure ensures that every variable has equal importance in training the SOM. The following formula is used for normalization:

$$\hat{x}_k(t) = \frac{x_k(t) - \bar{x}_k}{\frac{1}{n-1} \sum_{i=0}^{n-1} (\bar{x}_k - x_k(i))^2} \quad (1)$$

where

$x_k(t)$ is the k :th component of the measurement vector x , and \bar{x}_k is the mean of all the k :th components

Training a map is an iterative process in which a best matching unit (BMU) must first be found for each data vector. Each data vector must therefore be compared with each neuron on the map in order to find the BMU. A neuron on the map that most closely resembles the current vector is selected as its BMU, and the weight factor of the neuron and its neighboring neurons are adjusted according to the following formula:

$$w_{i_k}(t+1) = w_{i_k}(t) + h_{ci}(t) [x_k(t) - w_{i_k}(t)] \quad (2)$$

where

$w_{i_k}(t+1)$ is the new value of the k :th component of the weight factor vector of neuron i ,

$w_{i_k}(t)$ is the old value of the k :th component of the weight factor vector of neuron i , and

h_{ci} is a scalar gaussian kernel function:

$$h_{ci} = \alpha(t) \cdot \exp\left(-\frac{\|r_i - r_c\|^2}{2\sigma(t)^2}\right) \quad (3)$$

where

$\alpha(t)$ is the training rate factor,

$\sigma(t)$ is a factor that implies the size of the effective neighborhood, and r :s are the coordinates of the neurons.

The SOM can be interpreted by naming its neurons according to the classified measurement vectors. Neurons are named after the most probable process state, which can be calculated using the following formula:

$$Prob_{ij} = \frac{100 * (state)_{ij}}{(total)_j} \quad (4)$$

where

$Prob_{ij}$ is the probability of process state i in neuron j ,
 $(state)_{ij}$ is the number of vectors describing state i in neuron j ,
and $(total)_j$ is the total number of vectors in neuron j .

This gives the map a clear physical interpretation. However, the weakness of this method is that a neuron normally receives hits from measurement vectors representing both normal and undesired states. Therefore a neuron cannot be explicitly named to represent either a normal or undesired state.

If the SOM is used to detect only one disturbance, neurons can be named as disturbance neurons even if the probability for disturbance is less than 50 %. The sensitivity of the SOM can be adjusted by using different probability levels when naming the neurons. If neurons are labelled as disturbance neurons even with a low probability of the disturbance, the SOM will detect disturbances occurring with a high probability, but the number of false alarms will also be high. The kind of probability level that should be used depends on the application.

The state of the process can be monitored by drawing a pointer that displays the neuron corresponding to the latest measurement vector. It is often also advantageous to monitor how the state of the process has evolved during the last measurements, and therefore a suitable length trajectory of the latest neurons representing the latest operational states can be drawn on the map.

3. THE FAULT DIAGNOSIS SYSTEM TOOLBOX

The online toolbox presented in this paper monitors process states using Kohonen SOMs in conjunction with heuristic rules. The system also enables states of measurements and mathematical methods, based on limit value checking, to be used in heuristic rules.

The system consists of the following modules: process interface, application, databases and user interfaces. The structure of the toolbox is shown in Fig. 1.

Process interface. The data from process measurements and laboratory and maintenance databases are transferred through the process interface to the toolbox. The toolbox reads data directly through the network from an SQL-based relational database or from a text file.

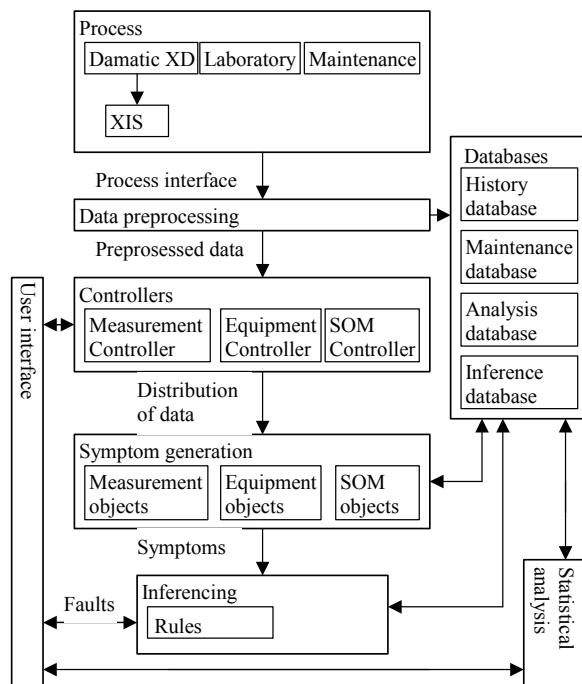


Fig. 1. Structure of the fault diagnosis system.

Application. After pre-processing, the data are transferred to controllers that distribute the data to measurement, formula and SOM objects. Based on the new measurement data, the objects determine their own state. If the state changes, the inference machine is informed accordingly. At the moment there are five state levels available: LOLO, LO, NORMAL, HI and HIHI. Based on the state changes of measurements and SOM objects, the inference machine updates all those rules where a state has been changed. If a rule's conditions are evaluated as true, the operator is informed accordingly.

The toolbox uses equipment-based storing of rules. The software model of the equipment consists of input variables, output states and rules. Measurements, formulae, SOMs and other equipment can be defined as input variables. Possible output variables are the state of the equipment or state of the process defined for the equipment. The conditions of the rules added to equipment model are based solely on the states of input variables. States of output variables are set as output of the rules. Rules are added to the system in IF-THEN form for example:

```
IF Gas_blower_SOM_vibration = HI AND
Waste_heat_boiler_SOM_sooting = NOT HI THEN
Gas_blower_dust_aggregation = HI
```

This makes it easier to control the rules, and also possible to chain the rules. This means that a disturbance detected in the last process equipment can be back traced to the real cause of the problem, which might be the first piece of equipment in the process.

Databases and analysis. All the measurement data and the rules that have evaluated as true are stored,

together with their symptoms and timestamps, in SQL databases for future analysis.

The toolbox has several different analysis methods. The system can calculate the most typical rules that have been evaluated as true within a given time interval. For more sophisticated methods the data can be easily imported, e.g. to MS Excel through an ODBC interface. The data stored in the database can also be run through the system offline in a simulation mode. This is useful when new rules or self-organizing maps are tested.

Graphical user interfaces. The toolbox consists of two different user interfaces. One is developed for the administrator and the other for the operator. The operational user interface offers all the information required for on-line monitoring of processes. Through this window it is possible to monitor the current state of a process and even to predict the future trend. The process monitoring user interface consists of four main blocks: maps, trends, active faults and history.

Self-organizing maps. The Kohonen self-organizing maps are presented to the user as a collection of red and blue neurons, each neuron being connected to its six neighbors. A totally red neuron represents a 100% probability fault, a totally blue neuron a totally normal condition of the process, and shades in between different probabilities of faults. A trajectory is also presented at the points where the last locations of the selected neuron can be seen as a darkening line. One can also take an individual look at the variables used in training the maps and see how the values of each variable are distributed on different neurons of the map. It is also possible to see the distribution of the hits of the training in order to determine whether or not the data were distributed evenly on the map. These are helpful features when analyzing the quality of the training. As SOMs may contain several different types of fault, only the component space of the most probable fault is presented to the operator. This keeps the operator view simple and clear. It is of course possible to examine all the other component spaces of different faults individually or all at once.

Trends. The interface can show two different types of trend: measurement trends and fault probability trends. The measurement trends show the development of individual measurements on a selected time-scale. The fault monitoring trends show, on the other hand, the development of the probability of each selected fault of different maps. By examining the trends it is possible to see which faults have occurred in the near past, and which faults are very probably going to occur in the near future.

Active faults. The faults currently occurring in the system are presented to the operator as a list. It contains information about the time instant when

each fault was detected, the equipment that the fault is associated with, the actual fault, and the symptoms related to the fault. The faults appear in the active faults window when they are detected, and are removed automatically when they disappear from the system being monitored. A more detailed description and recovery information are also shown for the selected fault. By clicking on the list with a mouse the selected fault can be changed.

History. The history window can analyze past faults by calculating some statistical averages like mean duration of a fault and the mean time lag between faults. It collects the information from the database in which all the faults are logged immediately after they are detected. Later on the information in the database is updated to mark down the ending times when the faults disappear. Exact time intervals, with an accuracy of one second, can be defined for the history view. An equipment-based view of faults can also be used to analyze the most usual problems of a specified piece of equipment. For the user, the name, beginning and ending times and symptoms are shown. As the rules are chained, it is also useful to inspect only those symptoms for which the same statistical information as for the faults is available.

The operational window is presented in Figure 2. Numbers 1 to 4 represent maps in an on-line state, and 5 to 8 are the corresponding trend displays. Number 9 is the history window with the four, pull-down menus for selecting the desired view. Number 10 is the active faults window, and number 11 shows the different pull-down menus and buttons for selecting maps, trends and the other windows described earlier.

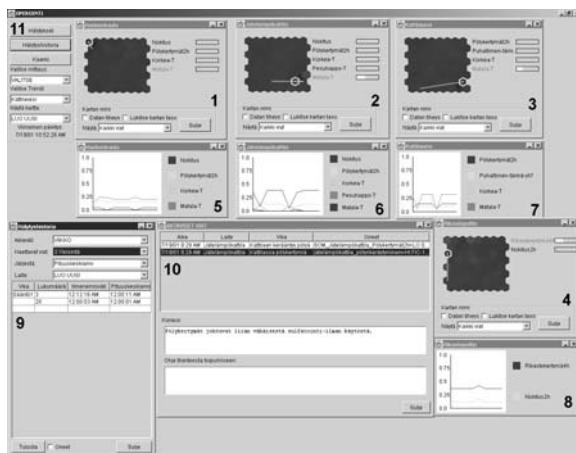


Fig 2. Fault diagnosis system in on-line use.

Administrator user interface. The administrator side, which has all the components required for training the maps, making the rules and other general configuration tools, can be made transparent for the operators. As long as it is configured properly, the operator only has to monitor the process without being concerned about data transfer or other modifications. The administrator side has five

important blocks: measurements, formulae, map training, equipment and trends.

Measurements and formulae. All the measurements that are used have to be defined, and they can be given four alarm limits to be utilized in the rules. Each measurement has two low and two high limits. Different formulae can be formed from the measurements using all the basic calculations. The block also supports selection of the maximum or minimum of two or more measurements that can later on be used as training variables for the maps.

SOM training. The most important tool for the engineer is the map training block. Maps can be trained in different sizes and with different parameters. The required training parameters and faults are selected via a database connection, and the map is trained with the selected amount of data. The map size, alpha parameter, number of epochs and the sizes of the neighborhood in the beginning and the end of the run can be defined. This block also contains two very important features that can be used to visualize the distribution of the data on the map in training and in simulation. These hit-counter files can be used together to evaluate the efficiency of the prediction of the map. The first hit-counter file is formed from the training data, but the data for the latter hit-counter-file can be freely selected in order to test the map.

Equipment and trends. Equipment block defines the connections between equipment and the measurements and maps related to each piece of equipment. It also has the interface for creating rules. Rules can only be formed from measurements and states defined for the equipment. In the trends block it is possible to define which trends on the maps are followed on the operator side. Different fault probabilities can be selected from different maps on one trend display.

In addition, the engineering side also contains a useful database organizer that can be used to prepare data for training the maps. The system has an open architecture, which enables easy integration with other applications. It has been developed using the platform-independent, object-oriented programming language Java. As all process information such as measurements, SOMs and rules are independent of the system, the toolbox application does not need re-coding if applied to a different process.

4. APPLICATIONS

4.1 Feed type identification using SOM

The heterogeneous characteristics of ore deposits pose a problem in achieving profitable production, because different ores require diverse optimal treatments at the concentrator. The expert system

developed by Jämsä-Jounela (1998) uses SOM for the on-line identification of the feed ore type, and a knowledge database that contains information about how to handle a specific ore type. The system was first tested at the Hitura Concentrator. changes in the mineralogy of the concentrator feed at the Hitura mine caused problems in process control. After each change in the feed type a new process control method had to be found. This was done through experimentation because the new type was often not known. These experiments took time and the resulting treatment method was not always the optimal one. The on-line SOM was trained using the five variables describing the feed type. A bubble algorithm was used to train a hexagonal SOM with 8 rows and 12 columns. The locations of the ore types were studied in the on-line SOM.

The system has been in use at the Hitura concentrator since December 1996. During this period, the Hitura personnel have maintained the system. According to the users, the system is capable of approximately indicating the feed type. The economical benefits of the expert system have primarily been due to faster and more accurate adaptation to ore type changes.

4.2 Estimating the Physical Quality of Cathode Copper using SOM

In the refining process, impure (appr. 99.0 w-%) copper anodes are electrically purified to high purity copper cathodes (over 99.99 w-%). As the electrolyte is continuously circulated through the electrolysis cells, its temperature and additive agent composition must be controlled. The pure electrolyte is an aqueous solution of copper sulfate and sulfuric acid. Removal of the soluble impurities (arsenic, nickel, etc.) is conducted in a separate process. Insoluble impurities, such as gold, silver and other noble metals, are handled in another process.

The production rate is controlled by the electric current applied to the cells. The current density typically varies between 200 – 330 A/m². The most important variables include electrolyte composition, temperature, current density, additive agent concentrations and anode quality. The efficiency of the process is expressed as the overall quality (physical and chemical) of the cathodes produced and the current efficiency of the refinery. The object of this study was to investigate the impact of anode impurities on the quality of the cathode copper produced. Electro-refining is a good example of a process in which variables are strongly correlated and the time constants exceptionally large, both of which make the process very problematic to monitor and control. The chemical and physical quality of the anodes are recognized as the major disturbance source for the process, and affect the essential variables of the refinery. The physical influence mechanisms are very complex and mostly not

completely understood. Investigating the problem through a physical model is thus laborious and uncertain. The chemical quality history of the anodes supplied to Pori refinery consisted of 1052 chemical analyses during the past 1.5 years of operation. There were nine variables representing the contents of the impurities occurring in the anode: antimony, arsenic, bismuth, lead, nickel, oxygen, selenium, silver and tellurium. The PCA modeling was carried out. The modeling resulted in three principal components with an explained variance of 72%. The most interesting result was that the observations clearly clustered in separate areas in the principal component space when there was a fall in the cathode quality. This implies that the anode impurities have an effect on the physical quality of the cathodes.

An 8 x 8 Self-Organizing Map was trained to the original anode analysis. The SOM was not as successful in classification as the PCA models since all the drops in cathode quality could not be traced to specific neurons. As PCA was successful in anode quality classification but not as visually efficient as SOM, it was decided to use these methods together. The procedure consisted of training a SOM with the principal component vectors of the PCA model. An 8 x 8 Map was used. This hybrid method further improved the classification and the results could be investigated from a one two-dimensional map. The quantization error of the SOM training was reduced to one third of the value achieved in direct SOM analysis. The drops in the physical quality of the cathode could be traced to specific neurons in the constructed map. The PCA-SOM model can be used to indicate whenever the anodes that have proved to be problematic arrive at the refinery.

4.3 Harjavalta Copper flash smelter

Outokumpu flash smelting is a pyro-metallurgical process for smelting metal sulfide concentrates. The Outokumpu flash smelting process consists of a flash furnace, waste heat boiler and electrostatic precipitator. A flash smelter usually also includes the following auxiliary units: feed mixture preparation and drying, converters, slag treatment system, SO₂ fixation system, anode furnace and anode casting. The implemented fault diagnosis system has been in test use at Outokumpu's Harjavalta copper smelter during summer 2001. The data used in training the SOMs were collected between November 2000 and August 2001. In August the SOMs were updated to include the latest available data. In contrast to our previous study, we concentrated on detecting only two process disturbances: formation of concentrate aggregations in the concentrate burner and formation of dust aggregations in the waste heat boiler. Both of these phenomena are difficult to detect merely by looking at plain process measurements. They are also major causes of process downtime and production losses.

Two SOMs were trained. In the training of the concentrate burner SOM, four temperature differences across the reaction shaft and amount of zinc in the concentrate were used as training variables. The waste heat boiler SOM was trained with the following variables: mean temperature of the boiler divided by the total gas flow through the boiler, pressure difference between the furnace and the boiler, pressure difference over the boiler, pressure of the boiler divided by the rotation speed of the gas blower, amount of flue dust in the feed, and amounts of zinc and lead in the concentrate. The SOMs are depicted in Figure 3.

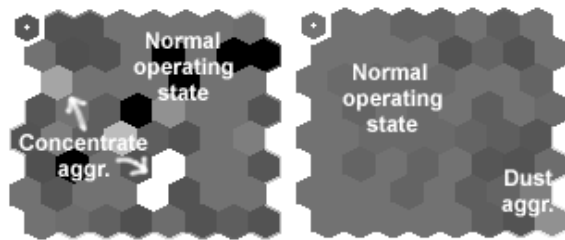


Fig. 3. The concentrate burner SOM and the waste heat boiler SOM.

Offline analysis of the concentrate burner SOM showed that it is able to detect 88,0 % of cases where the operation state is normal and 48,9 % of cases where concentrate aggregations are formed. The overall accuracy of the SOM is 81,3 %. Similarly the waste heat boiler SOM detects 92,6 % of the normal operation cases and 58,3 % of the cases where dust aggregations are formed. The overall accuracy of the SOM is 91,0 %.

4.4 Monitoring an industrial dearomatization process

The dearomatization process is a continuous chemical process the purpose of which is to remove aromatic compounds in the feedstock by hydrogenation. The feed to the unit is a form of petroleum oil cut. The process consists of two continuous reactors filled with a packed bed of catalyst, a distillation column, and several other unit operations. In the case study, the ability of the SOMs to detect faults was compared to statistical multivariate methods, such as partial least squares.

A dynamic process simulator model was first adjusted to match a dearomatization process in Fortum's petroleum oil refinery, and then used in generating data related to both normal operating and several fault conditions. The results indicated that both SOM and PLS are capable of detecting faults, while identification of the faults is considerably easier with the SOMs. Another result was that principal component pre-processing clearly improved the SOMs' performance. The results are presented in more detail by Bergman *et al.* (2002).

5. CONCLUSIONS

The structure of the process monitoring and fault diagnosis system and some of its applications have been described in this paper. Our recent studies show that emphasis on the utilization of trends in process variables can yield very promising results, especially in fast dynamic processes. Future effort will also be put on further development of the system in general, and some measurement malfunction detection algorithms will be implemented. The overall view of the operator window will be enhanced in accordance with operator interviews. The implemented system will be further developed.

6. REFERENCES

- Ahola, J., Alhoniemi, E. and Simula, O. (1999) Monitoring Industrial Processes Using the Self-Organizing Map, Proceedings of SMCia 99, pp. 22-27
- Bergman, S., Sourander, M. and Jämsä-Jounela, S-L. (2002). Monitoring of an Industrial Dearomatization Process, , submitted.
- Cser, L., Korhonen, A. S., Mäntylä, P. and Simula, O. (1998) Data Mining in Improving the Geometric Quality Parameters of Hot Rolled Strips, *Proceedings of the ICME 98*, pp. 265-270.
- Deventer, J., Moolman, D. and Aldrich, C. (1996) Visualisation of Plant Disturbances Using Self-Organising Maps, *Computers chem. Engng*, vol 20, Suppl., pp. S1095-S1100
- Frank, P.M., Kuipel, N. (1997), A fuzzy FDI decision-making System for the Support of the Human Operator, In Patton & Chen (eds), Proc. IFAC Symposium *SAFEPROCESS '97*, Hull, UK, Pergamon Press 1998, pp. 721-726.
- Himmelblau, (1978), Fault Detection in Chemical & Petrochemical Processes, *Elsevier*, Amsterdam, pp. 414.
- Isermann, R. (1984), Process Fault Detection Based on Modelling and Estimation Methods - A Survey, *Automatica*, **20**, pp.387.
- Isermann, R. (1997), Supervision, Fault-detection and Fault-diagnosis Methods-Advanced Methods and Applications, *Control eng. Practise* **5**, 639-652.
- Jämsä-Jounela, S-L. (1998), Ore Type Based Expert Systems in Mineral Processing Plants, *Part. Part sys. Charact.* **15** (1998), pp 200-207.
- Kohonen, T. (1990) The Self-Organizing Map, *Proceedings of the IEEE*, vol 78, pp 1464-1480.
- Kramer, M.A.. (1987), Malfunction Diagnosis Using Quantitative Models with Non-boolean Reasoning in Expert Systems, *AICHE J.*, **33**, pp.130-143.
- Patton, R.J., Uppal, F.J., Lopez-Toribo, C.J. (2000), Soft Computing Approaches to Faulty Diagnosis for Dynamic Systems: A Survey, Proc. IFAC Symposium *SAFEPROCESS '00*, Budapest, Hungary, pp. 298-311