Fuzzy Inference Maps for Condition Monitoring with Self-Organizing Maps

A. Alberto Cuadrado Ignacio Díaz Alberto B. Díez
Área de Ingeniería de Sistemas y Automática.
Universidad de Oviedo.
Campus de Viesques. Ed. Dep. 2.
33204 Gijón, Spain
cuadrado@isa.uniovi.es, idiaz@isa.uniovi.es,
alberto@isa.uniovi.es

Faustino Obeso Juan A. González
Centro de Desarrollo Tecnológico.
Aceralia Corporación Siderúrgica.
33480 Avilés, Spain

Abstract

In the framework of SOM methodology for process and system analysis, we suggest a method for identifying regions in the SOM visualization space corresponding to different conditions of a monitored process by means of a fuzzy rule system, as a way of incorporating expert knowledge to this methodology.

Keywords: Self-organizing map, fuzzy inference system, condition monitoring.

1 Introduction

The importance of condition monitoring and predictive maintenance in industry is well known. An effective system for condition monitoring can save large amounts of money, avoid accidents and even improve the quality of manufactured products.

There are many widely used methodologies for condition monitoring. The simplest one is the method of thresholds: a set of selected variables of the process are compared with threshold values previously configured for each variable. Methods of this kind use little prior knowledge, often based on experience. They are typically applied in vibration monitoring of electrical machines [10].

Another approach is that of the model-based methods, which use explicit equations, or prior knowledge about the process in general, to detect deviations from the operation considered normal [3]. Their main drawback is the difficulty of obtaining suitable models for many complex processes.

On the other hand, data-based methods extract information from data samples of the process. In these methods usually a model of the process is also involved, but these models are generated from data. We can include in this group methods based on neural networks [9]. Some of them have had a wide diffusion in the last decade. They consist in the visualization of a state trajectory in a bidimensional "state space" [6] [1] [2] [13].

2 SOM Methodology for Process and System Analysis

Several years ago, a method was proposed for condition monitoring of electrical machinery and industrial processes in general by means of the SOM (Self-Organizing Map) [6] [1]. A high-dimensional space, made from measured variables or computed features of the process, is non-linearly projected onto a low-dimensional space that can be displayed. That projection is performed by the SOM, which is previously trained with data obtained from the process, after a feature extraction stage oriented to the elimination of information that is not significant to the process condition.

The projection of the feature vector at consecutive times describes a trajectory in a visualization space. This can be considered as a non-linear projection of the trajectory that the process describes in the state space. The main advantage of this approach is that process state can be represented in a graphical way. This visualization space can be divided into regions corresponding to different conditions of the process, which allows to graphically determine the process condition.

The dimension reduction carried out by the SOM is expected to preserve the information about process condition contained in the original process data. This is possible through the property of topology preservation of SOM.

In the methodology for condition monitoring of Kohonen et al. it was proposed a displaying mode of the state trajectory having as background a colour map called *component plane*. There are always n component planes available, each one corresponding to a component or dimension of the high-dimensional input space. The colour under the projected point represents the value of that component predicted by the SOM model for the corresponding input point. There are many other types of information which can be visualized in the same way, as can be seen next.

2.1 Identification of regions in visualization space

Identifying the condition of the process corresponding to each region in the visualization space is not a direct task. Projected trajectories in the SOM visualization space produced by data taken from the process for known conditions can be compared with trajectories produced by new data to determine the condition corresponding to these new data. In the same way, histograms of activation of the SOM neurons can also be used for comparison. These histograms show in the visualization space how many times each neuron has been activated for a data set [1].

The *response surfaces* [11] are representations in the visualization space where the values

$$g_i = \frac{1}{1 + ||\mathbf{x} - \mathbf{m}_i||} \qquad (1)$$

are associated to each neuron i. The value $\mathbf{x}$ is a data sample from the process, and $\mathbf{m}_i$ is the codebook vector of the neuron i. The response surfaces are useful for novelty detection, i.e., detection of data samples corresponding to conditions that were not present in SOM training data. They also indicate the region of the visualization space which is activated by the sample. Accumulated (summed) response surfaces of labelled data sets (data for known process conditions) can be used as references to identify the condition of one unlabelled sample.

All these methods are completely data-based and they are only possible when labelled data are available. On the other hand we can reveal in the visualization space the clusters of data in the input space, each one corresponding to an operation condition of the process, by means of *distance matrices*. Under the name *distance matrices* we include several ways of displaying, in the SOM visualization space, information about inter-neuronal distances in the input space [11]. This kind of representations also show up in what extent one condition is similar to another[1]. Other techniques based on finding the cluster structure of the codebook vectors by means of standard unsupervised techniques have also been proposed recently [12]. However, all these techniques just display information on the cluster structure of the data and they require a subsequent labelling process to match regions in the visualization space to their corresponding process states.

Component planes can also be used to identify regions. Since they show in each point of the visualizations space the values of the process variables, if we can relate these values to the process operation condition then we will be able to label each neuron. However this is a difficult task when the process has a lot of variables.

3 Identifying Regions with Fuzzy Inference Systems

Although we do not have an explicit mathematical model of the process, almost intuitive knowledge about the process can be used by technical staff to determine its condition. An expert in the process can roughly infer its condition from the values of certain significant variables by just applying few rules. This task could be carried out automatically if he were able to express his knowledge in terms of fuzzy rules [4][8].

These fuzzy rules will have relations between the significant variables of the process as antecedents, and they will yield some desired information about the process as consequents. Codebook vectors of SOM can be used as inputs of a fuzzy inference system (FIS) built with these rules to yield truth values of the consequents for each node of the SOM. As a result, colour maps showing up these values can be displayed in the same way as component planes. Thus, information that the expert was able to deduce by looking

---

[1] Only for neighbour regions in the SOM visualization space

at the trajectory of the process over the component planes is now obtained in a straightforward manner and it is made available to anybody lacking of expert knowledge.

If $\mathbf{x}$ is a vector in the input space and ϕ is a fuzzy inference system then

$$c = \phi(\mathbf{x}) \tag{2}$$

the value c can be a fulfillment degree of the antecedent, when there is only one rule, or a fuzzy-variable value obtained after defuzzification. Particularly, if the vectors of the input space are the SOM codebook vectors, for each SOM unit i:

$$c_i = \phi(\mathbf{m}_i) \tag{3}$$

and we obtain a fuzzy inference map.

4 Experimental Results

This approach was applied to a large (6000 Kw) DC motor of a hot-rolling mill in Aceralia Steel Company under different working conditions. The measured variables were the mean values of armature voltage V_a, armature current I_a and field current I_f, so

$$\mathbf{x} = (V_a, I_a, I_f) \tag{4}$$

Data were taken during half an hour, which generated 960 samples. These samples were used to train a SOM.

A simple Mamdani-type fuzzy inference system with three inputs and three outputs was created. The inputs were the above mentioned motor variables and the outputs were the following:

- `State`: this fuzzy variable points out the working condition within the rolling process (rolling, not rolling, rolling end / motor regenerating, motor stopped).
- `Control`: this variable shows how the control is working (correctly or incorrectly).
- `Speed`: this variable is a fuzzy estimation of the motor speed (very low, low, medium, high, very high).

Figure 1 shows the component planes and the fuzzy maps for the previously trained SOM, and a projected trajectory for the following working conditions sequence: rolling, rolling end, not rolling, regeneration and stop (figure 2).

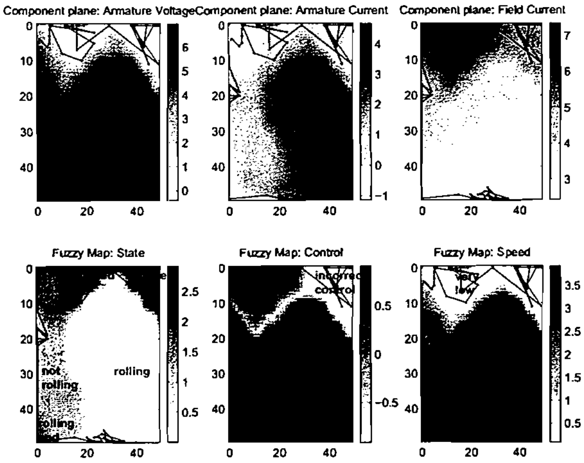

Figure 1: Component planes and fuzzy maps for a DC motor

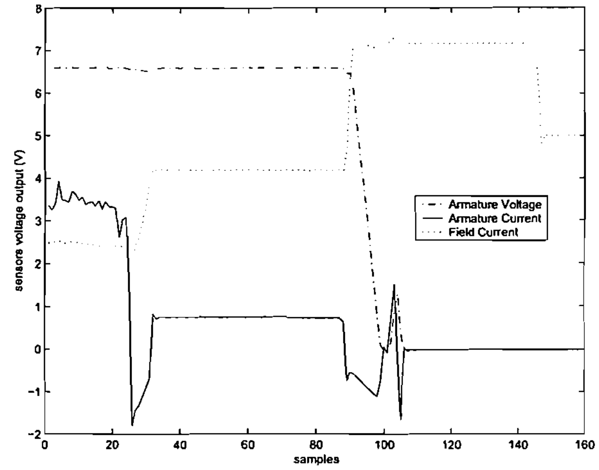

Figure 2: Data samples for the example sequence

The rules used to build the fuzzy inference maps were the following:

```
1. If (Va is not low) and (Ia is negative) then (state is regenerating)
2. If (Va is not low) and (Ia is medium) then (state is not-rolling)
3. If (Va is not low) and (Ia is high) then (state is rolling)
4. If (Va is low) and (Ia is zero) then (state is stopped)
5. If (Va is not high) and (If is not high) then (control is incorrect)
6. If (Va is high) or (If is high) (control is correct)
7. If (Va is high) and (If is low) then (speed is very-high)
8. If (Va is high) and (If is medium) then (speed is high)
9. If (Va is high) and (If is high) then (speed is medium)
10. If (Va is medium) and (If is high) then (speed is low)
11. If (Va is low) and (If is high) then (speed is very-low)
```

The fuzzy map for `state` could be substituted for a fulfillment degree for each state (the fulfillment degree of the antecedent of the rule corresponding to that state), which would produce four fuzzy maps. However, the used method, though not strictly exact, has the advantage of being displayable in only one graph.

The rules for the output `control` are given by the DC-motor control system. The motor is operated such that it can deliver maximum torque below its base speed and maximum power above its base speed. To control the speed below its base speed, the voltage applied to the armature of motor is varied with the field voltage held at its nominal value. To control the speed above its base speed, the armature is supplied with its rated voltage and the field is weakened. Therefore, if both armature voltage and field current are not high then the control system is not working properly.

The rules for the output `speed` are given by the fact that the higher is the armature voltage and the lower is the field current, the higher is the speed.

5 Conclusion

In this paper, a method is presented which allows to integrate prior knowledge given in terms of fuzzy rules within the framework of the SOM methodology for process condition monitoring. Subtle regions in the visualization space corresponding to specific process states verifying a specified set of conditions can be revealed in a straightforward manner through the proposed approach. This is specially useful in processes with a great number of variables or with many possible stable states, where other region identification approaches are not efficient.

Acknowledgements

This work is supported in part by the 1998 ECSC Steel Research Programme, project CECA-98-7210PE088-C.

References

[1] Esa Alhoniemi, Jaakko Hollmén, Olli Simula, and Juha Vesanto. Process monitoring and modeling using the self-organizing map. *Integrated Computer Aided Engineering*, 6(1):3-14,1999

[2] Ignacio Díaz Blanco, Alberto B. Díez González, Abel A. Cuadrado Vega, and José M. Enguita González. RBF approach for trajectory interpolation in self-organizing map based condition monitoring. In *J.M. Fuertes, editor, Proceedings of the 7th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'99)*, pages 1003-1010, UPC, Barcelona, Catalonia, Spain, 1999.

[3] Janos J. Gertler. Survey of model-based failure detection and isolation in complex plants. *IEEE Control Systems Magazine*, pp. 3-11, December 1988

[4] Rolf Isermann. On Fuzzy Logic Applications for Automatic Control, Supervision, and Fault Diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 28, no. 2, pp. 221-235, March 1998

[5] J.-S.R. Jang, C.-T. Sun and E. Mizutani. Neuro-Fuzzy and Soft Computing. *Prentice-Hall*, 1997

[6] Teuvo Kohonen, Erkki Oja, Olli Simula, Ari Visa, and Jari Kangas. Engineering applications of the self-organizing map. *Proceedings of the IEEE*, 84(10):1358-1384, october 1996.

[7] Teuvo Kohonen. Self-Organizing Maps. *Springer-Verlag*, 1995.

[8] W. Pedrycz and H.C. Card. Linguistic interpretation of self-organizing maps. In *Proc. of International Conference on Fuzzy Systems '92*, 1992.

[9] Timo Sorsa, Hiekki N. Koivo and Hannu Kovisto. Neural networks in process fault diagnosis. *IEEE Transactions on Systems, Man and Cybernetics*, vol. 21, no. 4, pp. 815-849, July/August 1991

[10] Peter J. Tavner and James Penman. Condition Monitoring of Electrical Machines. *Research Studies Press Ltd., John Wiley and Sons Inc.*, 1987

[11] Juha Vesanto. SOM-Based Data Visualization Methods. In *Intelligent Data Analysis*, vol. 3, no. 2, pp. 111-126, 1999

[12] Juha Vesanto and Esa Alhoniemi. Clustering of the Self-Organizing Map. *IEEE Transactions on Neural Networks*, vol. 11, no. 3, May 2000

[13] David J.H. Wilson, George W. Irwin and Gordon Lightbody.RBF Principal Manifolds for Process Monitoring. *IEEE Transactions on Neural Networks*, vol. 10, no. 6, November 1999