

Visual Data Mining and Monitoring in Steel Processes

Abel A. Cuadrado*, Ignacio Díaz*, Alberto B. Diez*, Faustino Obeso† and Juan A. González†

*Department of Electrical Engineering, University of Oviedo
Campus de Viesques s/n, 33204 Gijón, Asturias, Spain

†Innovation and Research Process Department, Aceralia Steel Company
33480 Avilés, Asturias, Spain

Abstract—Steel processes are often of a complex nature and difficult to model. All information that we have at hand usually consists of more or less precise models of different parts of the process, some rules obtained on the basis of experience, and typically a great amount of high-dimensional data coming from numerous sensors and variables of process computers which convey a lot of information about the process state. We suggest in this paper the use of a continuous version of the Self-Organizing Map (SOM) to project a high dimensional vector of process data on a 2D visualization space in which different process conditions are represented by different regions. Later, all sorts of information resulting from the fusion of knowledge obtained from data, mathematical models and fuzzy rules can be described in a graphical way in this visualization space.

I. INTRODUCTION

Process monitoring and control is a very wide field of study and of a great importance in industry, nowadays and ever. The requirements of accuracy and quality in manufacturing are raising rapidly due to an increasing competition in global markets, triggering each year higher costs in control and maintenance systems. Because of this, the development of new tools for process monitoring and control has become essential.

Industrial processes are commonly very complex. The adjective “complex” comes mainly from the fact that they have many variables with nonlinear relationships which also can change in time. These characteristics make them difficult to model and this becomes an important drawback in tasks like monitoring and control of such processes. Several approaches have been considered to accomplish these tasks and, specially, those oriented to fault detection and diagnosis. Model-based techniques use mathematical models of the process to generate residuals which indicate faults in that process (analytical redundancy) [1]. More recently, methods relying on artificial intelligence (AI) approaches are appearing, concerning data-based models founded on neural networks [2] and qualitative models founded on fuzzy inference systems [3], [4], [5], and even combinations of them [6].

Lately, AI techniques have been oriented to process monitoring by means of visualization methods [7], [8], [9], [10], [11] and nonlinear dimensionality reduction techniques in general, which have received considerable attention for high-dimensional data processing in many fields of sciences [12], [13], [14]. These approaches, in some way, try to take advantage of the visual-perception human skills, which are superior to those of any machine.

This paper is structured as follows. In section II, the basic problems in process monitoring are formulated, with special attention on steel processes. We also introduce the visualization approach, and the example of a hot-rolling mill used in subsequent sections is described. In section III, the idea of dimensionality reduction for process monitoring is presented. In section IV, several possibilities of process visualization are outlined, and finally, in section V we give some concluding remarks and propose future research lines.

II. MONITORING OF STEEL PROCESSES

Steel processes are difficult to model. Their behavior usually depends on several tuning parameters, control algorithms, preset values, lookup tables, nonlinear blocks, etc. which typically suffer plenty of modifications along the process history. Most of the times, we lack a global model which describes the whole process behavior and the relationships among all the variables. All information that we have at hand usually consists of more or less precise models of different parts of the process, some rules obtained on the basis of experience, and typically a great amount of high dimensional data coming from numerous sensors and variables of process computers which convey a lot of information about the process state.

Thus, we need methods which allow to deal with all these –heterogeneous– sources of knowledge, i.e. methods which:

- a) allow to learn from data
- b) allow to use the available prior knowledge about the process (rules, partial models, etc.).

Some “learning algorithms” such as multilayer perceptrons (MLP), or certain parametric statistical models allow to model the process from the information conveyed by the data by learning functional relationships among groups of process variables. However, they do not provide insight on the process and, hence, they do not allow us to integrate the prior knowledge about the process that we might have at hand in terms of models or rules. In other words, they can not be used to reason and they do not allow us to discover new features of the process behavior.

This paper is based on an idea initially proposed by Kohonen for process condition monitoring [7], [15]. We suggest here the use of a continuous extension of the Self-Organizing Map [16] to project a high-dimensional vector of process data, or features obtained from a previous feature extraction process, on a 2D visualization space in which different process conditions are represented by different regions. This visualization space provides a unified framework of process representation in which

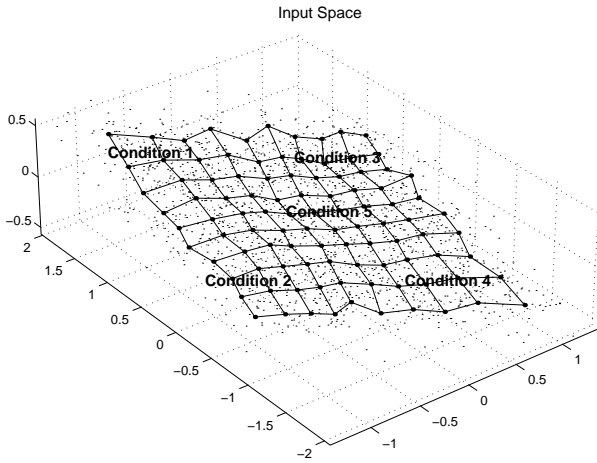


Fig. 2. Codebook vectors m_i in the input (feature) space.

not only process variables, but also other forms of prior knowledge about the process, such as validity of rules (fuzzy or crisp) as well as validity of known models of the process, can be described in a graphical way, in terms of regions, in the same way as cartographic maps allow to represent population densities, mountain elevations or rain-fall indices.

We illustrate this idea with data from a large DC motor of a hot-rolling mill (see table I). Several variables were analyzed: field current i_f , armature current i_a , speed ω , and armature voltage, V_a . Also, different segments of data, corresponding to different parts in the rolling of six coils, are labeled (see table II). Each segment is labeled with a letter identifying the coil (a,b,..., f) and two characters indicating start of rolling (sr), the several segments during rolling (r1, r2,...) or end of rolling (er). The nr means no-rolling.

TABLE I
DC MOTOR FEATURES.

Power:	6000 kW
Rated armature voltage:	700 V
Rated armature current:	9000 A
Rated field current:	150 A
Rated speed:	180 rpm
Max speed:	360 rpm
Location:	Hot finishing mill
Stand:	F2

TABLE II
LABELS AND FEATURES OF THE SIX DIFFERENT COILS.

COIL	TIME	THICKNESS	WIDTH	STEEL	YIELD STR.
A	15:44:48	1.73	1010	B085G99	32
B	15:46:41	1.73	1010	B085G99	32
C	15:48:53	4.02	1010	B012F53	22
D	15:50:47	4.02	1010	B012F53	22
E	15:53:04	4.02	1010	B012F55	22
F	15:54:44	1.55	1010	B011B99	24

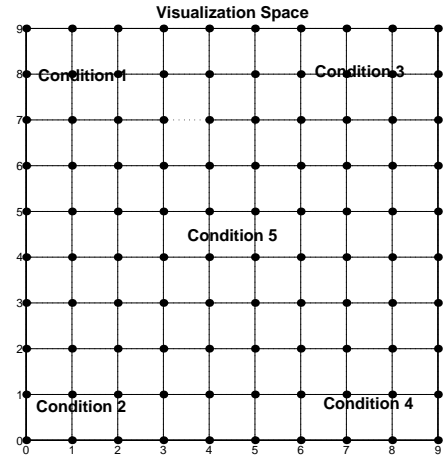


Fig. 3. Regular SOM grid in the output (visualization) space formed by the nodes g_i .

III. VISUAL ANALYSIS OF PROCESSES

It is not unusual that a process generate data even from hundreds of variables. The value of these variables form a vector $v(t)$ at each time t and, therefore, $v(t)$ is a trajectory of process measures in \mathbb{R}^p , being p the number of process variables. This measures are function of the state variables of the process and, though they are not likely to hold the whole state information, if they are properly chosen, the measures contain useful information about the process condition. Among all the measured variables, a number of them are highly correlated and they contain redundant information. Several of them may have a very complex variation pattern or they may be mixed with noise. To overcome all these problems, usually, the measured variables are previously processed to obtain a set of features with a more packed information content. After this feature extraction, which transforms the measure space into a feature space, the information about steady states of the process reflects itself as clusters of data in the trajectory of features $x(t) \in \mathbb{R}^n$, and the transition between them as data paths between clusters. This trajectory in feature space can be thought of as a state trajectory.

Maybe, the most straightforward way of exploiting the information provided by the state trajectory is its visualization. However, when $n > 3$, being n the number of process features, this trajectory can not be graphically represented. A method to project data with nonlinear relationships from a high-dimensional space onto a visualization space is needed. In addition, this method must preserve the information about the process state, i.e., it must project different states in different regions of visualization space. Several methods have been proposed to accomplish this task. Maybe, the best synergy between simplicity and high performance is provided by the Self-Organizing Map [17], [7], which can be combined with a General Regression Neural Network (GRNN) [18] in a technique called Kernel Regression SOM (KR-SOM) [16] to overcome quantization errors derived from the discrete character of the SOM. The dimension-reduction scheme for process monitoring is shown in fig. 1.

The Self-Organizing Map can be defined as a nonlinear mapping from a high-dimensional input space (here, the space of

The Dimension Reduction Approach

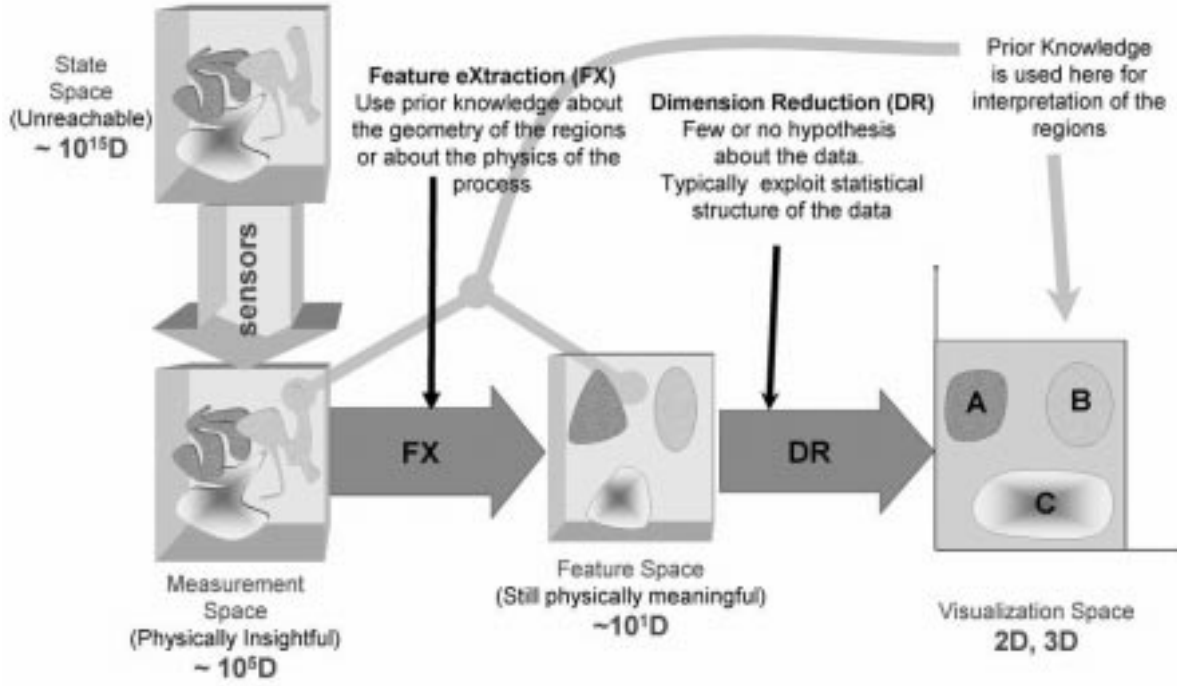


Fig. 1. The idea of dimensionality reduction for process monitoring.

feature vectors) \mathbb{R}^n , onto a low dimensional (typically 2D) rectangular grid (also called *visualization space*). The map is defined by a set of points (*codebook vectors*) \mathbf{m}_i in the input space and a corresponding set of nodes of a rectangular grid \mathbf{g}_i . A training algorithm arranges the codebook vectors \mathbf{m}_i so that they acquire the same geometry of the input data in a smooth and ordered fashion –see fig. 2 and 3; note the matching of condition-related regions between both spaces–.

For each feature vector \mathbf{x} , the SOM projection is defined so that the projection of \mathbf{x} is the 2D node position \mathbf{g}_c corresponding to the nearest codebook vector \mathbf{m}_c to the actual feature vector \mathbf{x} in the input space. Because of this definition, every \mathbf{x} in the Voronoi's region of \mathbf{m}_c projects onto \mathbf{g}_c , resulting in a quantization error $q_e = \|\mathbf{x} - \mathbf{m}_c\|$.

IV. CONTINUOUS SOM PROJECTION

In the KR-SOM, the projection from input space onto visualization space (forward projection) is carried on using a GRNN, which performs an approximation of continuous function $\mathcal{S}_{\mathbf{m}_i \rightarrow \mathbf{g}_i} : \mathbb{R}^n \rightarrow \mathbb{R}^2$ by interpolation from a set of centers $\mathbf{m}_i \in \mathbb{R}^n$ and their correspondent centers $\mathbf{g}_i \in \mathbb{R}^2$, in this case, the SOM codebook vectors and the grid units respectively:

$$\mathbf{y} = \mathcal{S}_{\mathbf{m}_i \rightarrow \mathbf{g}_i}(\mathbf{x}) = \frac{\sum_i \Phi(\|\mathbf{x} - \mathbf{m}_i\|) \mathbf{g}_i}{\sum_j \Phi(\|\mathbf{x} - \mathbf{m}_j\|)} \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^2$. In the same manner, there also exists the possibility of performing the projection from visual-

ization space onto input space (backward projection) :

$$\mathbf{x} = \mathcal{S}_{\mathbf{g}_i \rightarrow \mathbf{m}_i}(\mathbf{y}) = \frac{\sum_i \Phi(\|\mathbf{y} - \mathbf{g}_i\|) \mathbf{m}_i}{\sum_j \Phi(\|\mathbf{y} - \mathbf{g}_j\|)} \quad (2)$$

where the kernel function Φ usually has the form:

$$\Phi(\|\mathbf{z}\|) = e^{-\frac{\|\mathbf{z}\|^2}{2\sigma^2}}. \quad (3)$$

Once a feature vector is projected onto a visualizable space, the original knowledge available in feature space is not lost. The visualization space can be used to provide further information about the regions that the state trajectory goes through along time in different process working conditions. This information can be represented using colors or shades of gray, which correspond to different values of any measurable property evaluated at each point of the input (feature) space. To achieve it, each visualization space point is projected by means of the backward projection and then it is colored in the visualization space with a color or shade of gray proportional to the value of the required property in the input space of the projected point.

A. Trajectory projection

The projection of the feature vector trajectory can be obtained from direct application of the forward projection given in (1):

$$\mathbf{y}(t) = \mathcal{S}_{\mathbf{m}_i \rightarrow \mathbf{g}_i}(\mathbf{x}(t)). \quad (4)$$

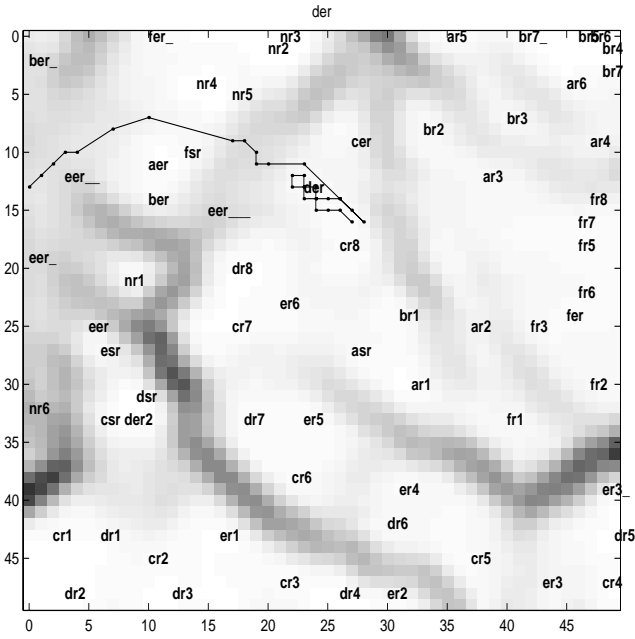


Fig. 4. Trajectory in visualization space of an end of rolling process (der). It goes from the rolling region (right) to the regeneration region (left).

In fig. 4, an example state trajectory corresponding to the end of rolling of a coil (D) is projected onto the visualization space. The methods to identify the regions that it spans will be explained in subsequent sections.

B. Component planes

The values of the original coordinates in input space of a point projected in visualization space are supplied by the component planes. In fig. 5 the SOM component planes are shown. The SOM component planes display in visualization space the values of the process features at each point as a gray shades (or colored pixels).

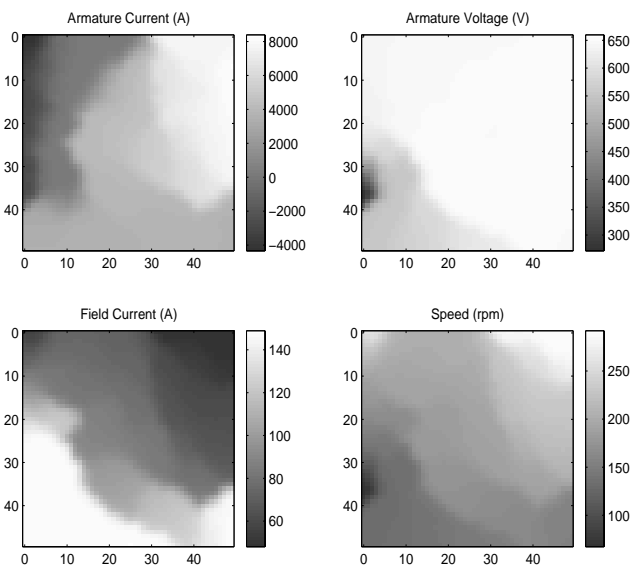


Fig. 5. SOM component planes.

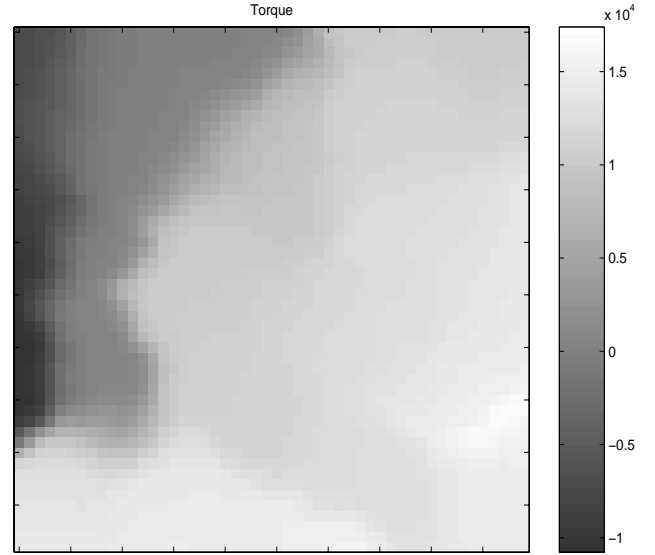


Fig. 6. SOM plane of torque T_e .

Component planes are not restricted to be of variables used in SOM training. Variables dependent of those can also be displayed in visualization space. In fig. 6, the SOM plane of torque is displayed. It is not a variable included in SOM training, but a variable derived from the expression $T_e = K \cdot i_a \cdot i_f$, where K is a constant.

The component planes themselves are an important source of information about the process. However, to extract this information, the component planes must often be combined with the knowledge about the process that the technician which uses them possesses, sometimes in the form of mathematical models or simply fuzzy rules. If the trajectory of fig. 4 were plotted on the component plane of armature current, looking at it someone who knows the rolling process of the example could infer that the trajectory goes from a rolling region (relatively high armature current) to a motor braking regeneration region (negative armature current). This can be made easy as will be explained in the following sections.

C. Distance map

Another possibility, when representing information in visualization space, is that some property D_i is known only for the SOM units. Then, an equation similar to those of (1) and (2) can be used:

$$\mathcal{D}(\mathbf{y}) = \frac{\sum_i \Phi(\|\mathbf{y} - \mathbf{g}_i\|) D_i}{\sum_j \Phi(\|\mathbf{y} - \mathbf{g}_j\|)} \quad (5)$$

where $\mathcal{D}(\mathbf{y})$ is the value to color the pixel \mathbf{y} in visualization space. If D_i is the mean distance in input space of SOM unit i to its neighbor units, then $\mathcal{D}(\mathbf{y})$ represents the interpolated distance matrix.

The distance matrix shows in visualization space the actual distances between points in input space. Dark zones point out a region of the input space which is very compressed in the visualization space. Points that appear close in visualization space are actually close only if there are light shades between them

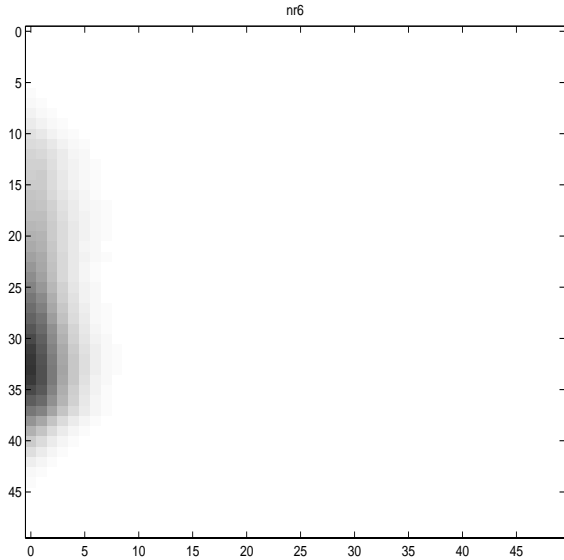


Fig. 7. Activation map for nr6 data segment. The dark zone is the region activated by this data set.

in distance matrix. Therefore, the distance matrix reveals data clusters in input space, which are expected to correspond to different process conditions. In fig. 4, an example state trajectory is projected onto the visualization space with the distance matrix.

D. Activation maps

The labels displayed in the visualization space in fig. 4 show up regions corresponding to data segments for each rolling of table II. Analogously, regions which have been identified as corresponding to certain process conditions can be labelled for fast visual identification. This labeling can also be done in an automated way. When the process condition corresponding to a given data set is known, the easiest way to place the labels is, as they were in this case, using *activation maps*, which are obtained as follows. Let $\{\mathbf{x}_k\}_{k=1,\dots,K}$ be a data set. The activation level of SOM unit i for that data set is defined as:

$$A_i = \frac{\sum_k h_{c(k)i}}{\sum_{j=1}^M h_{ij}} \quad i = 1, \dots, M \quad (6)$$

$$c(k) = \arg \min_i \{d(\mathbf{x}_k, \mathbf{m}_i)\} \quad (7)$$

where M is the number of SOM units and $c(k)$ is the index of the winner SOM unit for \mathbf{x}_k . The continuous version is obtained with (5). Activation maps show up which zones in visualization space correspond to a certain data set of input space. In fig. 7, the activation map corresponding to the data segment nr6 (not rolling, segment six) is shown.

E. Residual vector

As it was mentioned in section IV-B, the component planes give the feature values for the projection of a feature vector in visualization space. However, this is only true when the SOM models the projected point, i.e., when this point belongs to a process condition present in SOM training. If this does not

happen, the projected point only shows the most similar process condition that was present in SOM training data. An available method to detect this situations is the *modeling error*. The modeling error is the difference between a point in input space and the point that results when the former is projected in the visualization space and then back to input space. Here, the SOM is a model of the process and this modeling error is a residual vector:

$$\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}} = \mathbf{x} - \mathcal{S}_{\mathbf{g}_i \rightarrow \mathbf{m}_i}[\mathcal{S}_{\mathbf{m}_i \rightarrow \mathbf{g}_i}(\mathbf{x})]. \quad (8)$$

The values of the residual-vector components generated through mappings (autoassociators) based on the support of the data pdf hold a physically insightful relationship with the novel process conditions[19]. Accordingly, a componentwise representation of the residual vector reveals which features¹ are most susceptible to be involved in the fault, as well as the sign of the deviations.

The residual vector can be represented as a graph for each feature, but it can be also displayed with gray shades (see fig. 8), in which the horizontal axis is “time” and vertical axis corresponds to the residuals of each process feature present in SOM training. This way of displaying is very useful when the number of process features is large. Fig. 8 and 9 show a rolling with steel B085G99 and target thickness 4.06 that was not present in SOM training data. From the state trajectory projection it can be inferred that this rolling is more similar to those of C, D and E. However the residual vector unveils some differences of this new situation projects over the border between nr6, cr1 and dr1. Since that is a zone of high neuronal distances, the condition signaled by the KR-SOM seems to be an intermediate situation resulting from the interpolation of the mentioned conditions, and with respect to which the armature current is now higher and the field current is slightly lower (the sign of the residuals is only distinguishable in the color version of the figure).

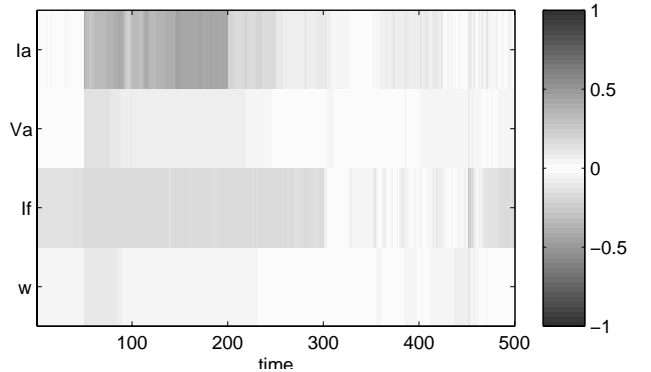


Fig. 8. Residuals for a rolling with steel B085G99 and thickness 4.06. The original is in color and the positive and negative residuals can be distinguished in it.

F. Residual or model maps

We can incorporate knowledge in form of equations (and mathematical models in general) in visualization space [20].

¹Those whose residuals differ significantly from zero.

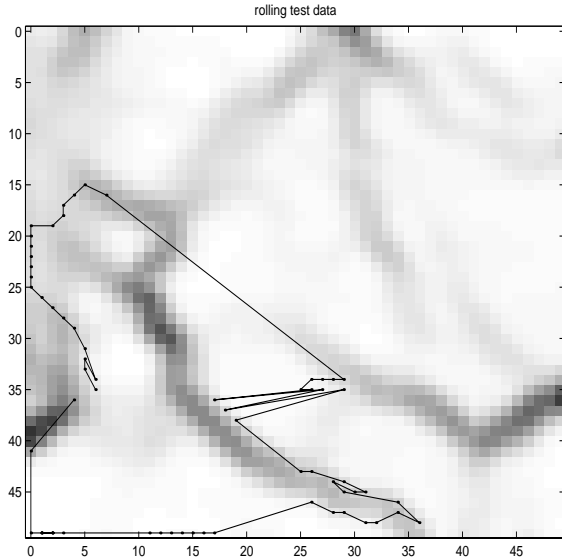


Fig. 9. State trajectory for a rolling with steel B085G99 and thickness 4.06 not present in training data.

Residuals obtained from known models of the process can be represented in visualization space as *residual maps* or *model maps*. A “known model” is usually an equation, which can be converted to the form $f(\mathbf{x}) = 0$. The residual map for that equation is obtained by evaluation of the following equation at each point \mathbf{y} of the visualization space:

$$\epsilon(\mathbf{y}) = f(\mathbf{x}) = f(\mathcal{S}_{\mathbf{g}_i \rightarrow \mathbf{m}_i}(\mathbf{y})). \quad (9)$$

As an example, fig. 10 shows a regression line adjusted for pairs of data armature voltage/speed in the control zone of speed under its rated value. This regression line yields an equation $f(V_a, \omega) = V_a - 4.08\omega + 4.23 = 0$, which corresponds to the DC motor equation:

$$V_a = R_a \cdot i_a + L_a \frac{di_a}{dt} + K \cdot i_f \cdot \omega \quad (10)$$

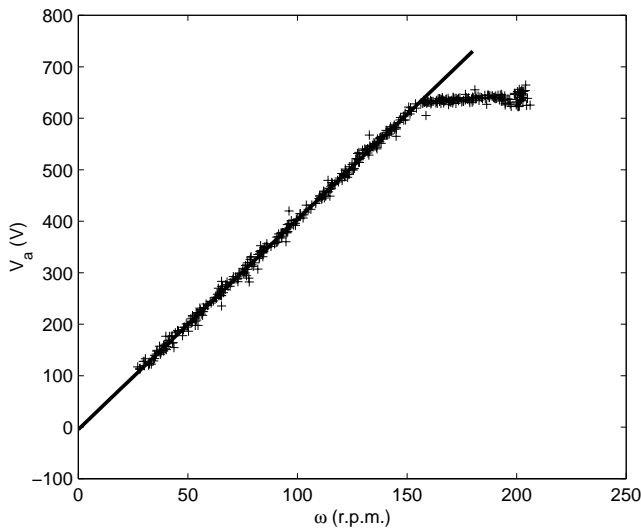


Fig. 10. Graph for speed - armature voltage. The line was adjusted by least squares in the zone of speed control by armature voltage.

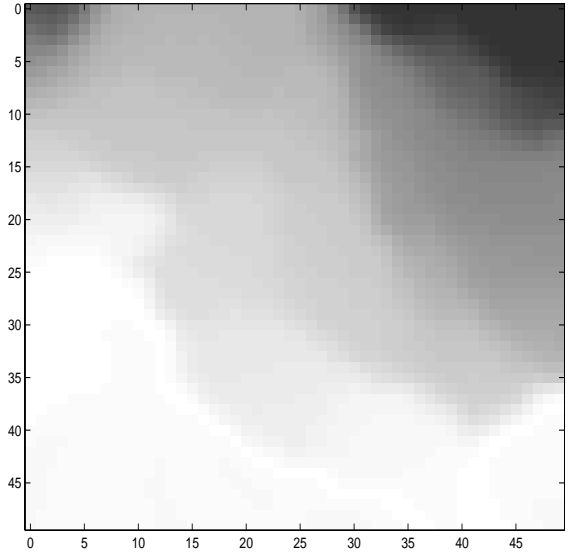


Fig. 11. Residual map for the zone of speed control by armature voltage. White zones are zero residuals.

TABLE III

FUZZY RULES FOR AN ESTIMATION OF SPEED.

if (Va is high) and (If is low) then (Speed is very-high)
 if (Va is high) and (If is medium) then (Speed is high)
 if (Va is high) and (If is high) then (Speed is medium)
 if (Va is medium) and (If is high) then (Speed is low)
 if (Va is low) and (If is high) then (Speed is very-low)

when it is in steady state and the armature current and the field current are both constant. In points of visualization space where this equation does not verify, the residual, namely $\epsilon = f(V_a, \omega)$, is not zero (fig. 11). Therefore, residual maps point out the regions in visualization space where a given equation related to the process holds, and where it does not and in what extent.

G. Fuzzy inference maps

Knowledge in form of fuzzy rules can also be added by means of fuzzy inference maps [20]. Thus, if g is a fuzzy inference system (FIS) with one output τ and as many inputs as process features:

$$\tau(\mathbf{y}) = g(\mathbf{x}) = g(\mathcal{S}_{\mathbf{g}_i \rightarrow \mathbf{m}_i}(\mathbf{y})). \quad (11)$$

As an example, in fig. 12 (left) it is shown an estimation of speed from the variables armature voltage and field current using the rules in table III. This figure is fairly similar to that of the component plane of speed in fig. 5. A fuzzy inference map can describe a degree of fulfilment of a process condition. For example, typing the rules of table IV in a FIS, the regions of the visualization space where the system is rolling are automatically generated—see fig. 12 (right)—.

H. Correlation maps

Looking at the component planes in fig. 5 it can be noted that the component plane for speed is the negative image of the component plane for field current, except for those regions where ar-

TABLE IV
FUZZY RULES FOR “ROLLING” CONDITION.

if (V_a is not low) and ((I_a is high) or (I_a is medium)) then (Rolling is true)
if (V_a is low) or (I_a is zero) or (I_a is negative) then (Rolling is false)

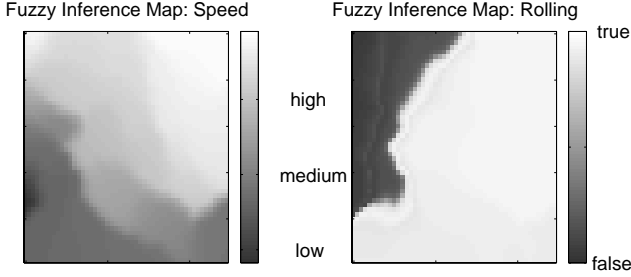


Fig. 12. Fuzzy inference maps. Fuzzy map for speed: linguistic estimation of speed from the variables armature voltage and field current. Fuzzy map for rolling condition: in light regions the system is rolling.

armature voltage drops under its maximum value. This same conclusion can be reached at a glance in the corresponding correlation map of fig. 14. The correlation maps [21] show up the local correlation between process variables at each point of visualization space. Correlation analysis is a powerful technique to discover linear relationships between pairs of variables, but traditionally is applied on a whole data set, providing only global information. However, this is not very useful in data from industrial processes, which typically contain several working points where the correlations between process variables may differ. A local approach is possible by weighting data in input space with a kernel function $w_k(\mathbf{y}) = e^{-\frac{1}{2}\|\mathbf{x}_k - \mathcal{S}_{g_i \rightarrow m_i}(\mathbf{y})\|^2/\sigma^2}$. Each point of the visualization-space grid \mathbf{y} is projected onto input space, and this image $\mathcal{S}_{g_i \rightarrow m_i}(\mathbf{y})$ is used as center of the kernel function to calculate the local mean and the local covariance matrix:

$$\mathbf{m}(\mathbf{y}) = \frac{\sum_k \mathbf{x}_k \cdot w_k(\mathbf{y})}{\sum_k w_k(\mathbf{y})} \quad (12)$$

$$\mathbf{C}(\mathbf{y}) = (c_{ij}) = \frac{\sum_k [\mathbf{x}_k - \mathbf{m}(\mathbf{y})][\mathbf{x}_k - \mathbf{m}(\mathbf{y})]^T w_k(\mathbf{y})}{\sum_k w_k(\mathbf{y})} \quad (13)$$

Then, the local correlation matrix around \mathbf{y} is:

$$\mathbf{R}(\mathbf{y}) = (r_{ij}) \quad \text{where} \quad r_{ij} = \frac{c_{ij}}{\sqrt{c_{ii}c_{jj}}} \quad (14)$$

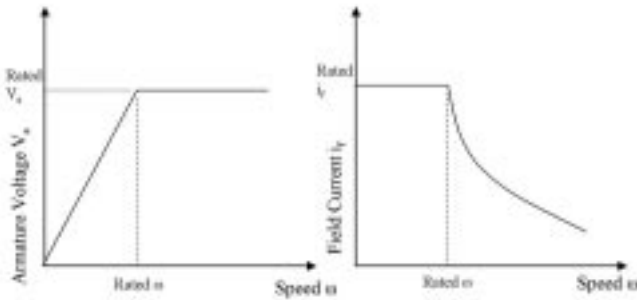


Fig. 13. Field-weakening control scheme.

The correlation map for variables x_i and x_j is represented by the value r_{ij} (or r_{ji} due to the symmetry of covariance and correlation matrices) at each point \mathbf{y} of the visualization-space grid. Note that fig. 14 represents indeed a generalization of a correlation matrix, showing up the local distribution of correlations.

In fig. 14, one can observe the direct correlation between V_a and ω (white area) and the inverse correlation between I_f and ω (dark area) in complementary regions of the visualization space due to the field-weakening control scheme depicted in fig. 13.

V. CONCLUSIONS AND FURTHER WORK

Steel processes are often complex and typically involve many variables. Information about the process is often presented in quite heterogeneous ways such as models, rules, big amounts of data recorded from multiple sensors and stored in large databases or just years of experience which just “stay” in the minds of the technicians and which they do not even know how to formulate. Most of the times none of these descriptions is sufficient by itself and approaches taking advantage of all them can become very useful.

In this paper we describe an approach to model the process on the basis of a nonlinear projection of process data on a 2D space, which can be visualized, while preserving the main geometrical relationships among variables in the input (feature) space. This approach “translates” any reasoning mechanisms (models, fuzzy rules, etc.), which take place in the feature space, into visual metaphors in the visualization space, allowing us to exploit our ability to infer conclusions in a visual way.

We show how to translate several of these “kinds of knowledge” about the process into regions in the 2D space and use them to establish conclusions in a typical steel process (a large DC motor used in a hot-rolling mill).

We believe that this visual representation of knowledge can provide an extremely powerful tool to model and understand many other difficult steel processes. However, much further work still remains to be done to exploit the capabilities of visualization, as well as to discover enhanced ways to represent the available knowledge. Currently, these methods are being tested to investigate the relationships between more than eighty operating parameters in a tandem cold rolling mill and several quality measures of the coils.

REFERENCES

- [1] J. Gertler, “Survey of model-based failure detection and isolation in complex plants,” *IEEE Control Systems Magazine*, vol. 8, pp. 3–11, Dec. 1988.
- [2] D. J. H. Wilson and G. W. Irwin, “RBF principal manifolds for process monitoring,” *IEEE Transactions on Neural Networks*, vol. 10, pp. 1424–1434, Nov. 1999.
- [3] R. Isermann, “On fuzzy logic applications for automatic control, supervision, and fault diagnosis,” *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 28, no. 2, pp. 221–235, 1998.
- [4] T. Harris, “A Kohonen S.O.M. based, machine health monitoring system which enables diagnosis of faults not seen in the training set,” in *International Joint Conference on Neural Networks, IJCNN’93*, (Nagoya, Japan), pp. 947–950, 1993.
- [5] R. Patton and C. Lopez-Toribio, “Artificial intelligence approaches to fault diagnosis,” in *IEE Colloquium on Update on Developments in Intelligent Control (Ref. No. 1998/513)*, p. 312, Oct. 1998.

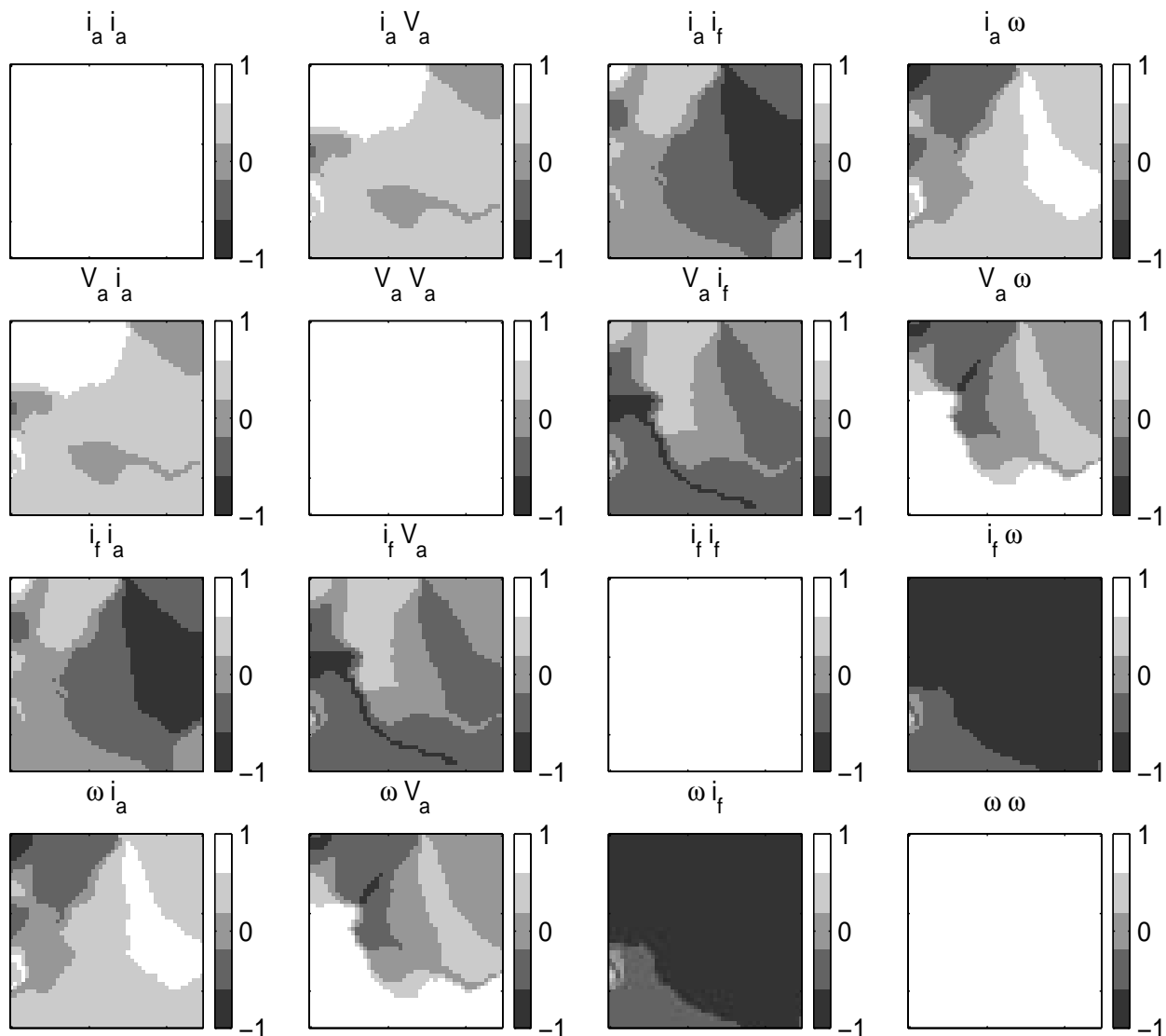


Fig. 14. Correlation maps. The gray levels have been reduced to five for easy identification of zero correlations. In color this is not necessary.

- [6] P. P. Bonissone, Y.-T. Chen, K. Goebel, and P. S. Khedkar, "Hybrid soft-computing systems: Industrial and commercial applications," *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1641–1667, 1999.
- [7] T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas, "Engineering applications of the self-organizing map," *Proceedings of the IEEE*, vol. 84, pp. 1358–1384, october 1996.
- [8] O. Simula, J. Vesanto, E. Alhoniemi, and J. Hollmén, "Analysis and modeling of complex systems using the self-organizing map," in *Neuro-Fuzzy Techniques for Intelligent Information Systems*. (R. Kasabov, N. & Kozma, ed.), pp. 3–22, Physica-Verlag, 1999.
- [9] J. Vesanto, "Som-based data visualization methods," *Intelligent Data Analysis*, vol. 3, no. 2, pp. 111–126, 1999.
- [10] W. Pedrycz and H. C. Card, "Linguistic interpretation of self-organizing maps," in *IEEE Int. Conf. on Fuzzy Systems*, (Piscataway, NJ), pp. 371–378, IEEE Service Center, 1992.
- [11] C. M. Bishop, M. Svensen, and C. K. I. Williams, "GTM: The generative topographic mapping," *Neural Computation*, vol. 10, no. 1, pp. 215–234, 1998.
- [12] P. Demartines and J. Hérault, "Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets," *IEEE Transactions on Neural Networks*, vol. 8, pp. 148–154, January 1997.
- [13] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, Dec. 22 2000.
- [14] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, Dec., 22 2000.
- [15] T. Kohonen, *Self-Organizing Maps*. Springer-Verlag, 1995.
- [16] I. Díaz, A. B. Diez, and A. A. Cuadrado, "Complex process visualization through continuous self organizing maps using radial basis functions," in *International Conference on Artificial Neural Networks (ICANN'01)*, (Vienna, Austria), pp. 443–450, 2001.
- [17] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, pp. 1464–1480, september 1990.
- [18] D. F. Specht, "A general regression neural network," *IEEE Transactions on Neural Networks*, vol. 2, pp. 568–576, Nov. 1991.
- [19] I. Díaz and J. Hollmén, "Residual generation and visualization for understanding novel process conditions," in *International Joint Conference on Neural Networks (IJCNN, WCCI'2002)*, (Honolulu, Hawaii, USA), pp. CD-ROM, 2002.
- [20] I. Díaz, A. B. Diez, A. A. Cuadrado, and M. Domínguez, "Prior knowledge integration in self organizing maps for complex process supervision," in *International Federation of Automatic Control 15th IFAC World Congress*, (Barcelona, Spain), pp. 000–000, 2002. in press.
- [21] I. Díaz, A. A. Cuadrado, and A. B. Diez, "Correlation visualization of high dimensional data using topographic maps," in *International Conference on Artificial Neural Networks (ICANN'02)*, (Madrid, Spain), pp. 000–000, 2002. in press.