International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:1, No:5, 2007

Genetic Algorithms in Hot Steel Rolling for Scale Defect Prediction

Jarno Haapamäki and Juha Röning

***Abstract*—** Scale defects are common surface defects in hot steel rolling. The modelling of such defects is problematic and their causes are not straightforward. In this study, we investigated genetic algorithms in search for a mathematical solution to scale formation. For this research, a high-dimensional data set from hot steel rolling process was gathered. The synchronisation of the variables as well as the allocation of the measurements made on the steel strip were solved before the modelling phase.

***Keywords*—** Genetic algorithms, hot strip rolling, knowledge discovery, modelling.

I. Introduction

MACHINE learning methods were applied to hot strip rolling since the more traditional methods have not proved feasible in scale defect modelling or prediction. Scale defects are a common group of surface defects in hot steel rolling, and due to stringent surface quality requirements it is important to recognize the risk factors that cause scale on the surface of steel products.

Oxidation of steel results in a three-layer scale consisting of wüstite, FeO, magnetite Fe_3O_4 and hematite Fe_2O_3. Wüstite, which is the innermost layer, is stable only at temperatures above 570°C. In hot rolling process conditions it constitutes roughly 95% of scale [1,2,3]. There are several scale types with different mechanisms of formation. Rolled-in or black scale develops when harder oxides are rolled into [1] the surface during the finishing process. Red scale is mostly associated with a high Si-content, although this is not a necessary condition [2]. In red scale detection, there is only a small possibility for confusing red scale with some other defect [4]. When the two types coincide, it is possible that some rolled-in scale defects will be ignored by the detection system, and therefore strips with over 0.1% Si-content were therefore investigated only for red scale and the other parts of the data for rolled-in scale.

The origin of scale defects has been a topic of interest in many research projects, but it is still hard to find literature on the modelling of defects The first published modelling study of surface quality related to scaling, that gave rise to a practical application (VAI-Q Strip), was published recently [5]. However, no physical model for scale formation has been formulated so far. The problem for modelling is due to high dimensional variable group with their interactions.

Neural networks have been traditionally used for modelling purposes when the formulation of a physical model has not been possible. Such models suffer from a lack of informativity and interpretability. However, a search for a formal model is possible by using methods derived from the Genetic Algorithms, hereafter GA [6].

Several reasons for scale formation are mentioned in the literature, including the effects of temperature and time [1,3,5,7], rolling forces and reduction [1,5], steel composition [8] and gas atmosphere. Silicon content and reheating temperature [3] are also relevant factors, since molten fayalite, Fe_2SiO_4, accelerates the scaling rate [2]. Furthermore, uneven cooling was mentioned as a contributing factor in [7]. However, the current knowledge about how the process conditions and the steel grade affect the final properties of tertiary scale is fragmentary [1].

Three different machine learning methods have been applied and compared. Two prediction models were implemented with neural networks (multilayer perceptrons and self-organizing maps) in an earlier study [9]. The original goal was to recognize high-risk process conditions and to identify the part of the process that causes the scale defects. Therefore, further knowledge in the form of a mathematical solution was searched for in this study by means of genetic programming. The application constructs a simple mathematical model, which is used to predict scale defects. Moreover, continuous learning was applied in this study. High-frequency measurements of the rolling process enabled a local analysis of scale defects in different parts of the strips. The response of the video camera-based machine vision system was utilised in the training of models.

II. Data

At the beginning of the research, data was gathered and analysed iteratively. The final data set was collected at Rautaruukki Oyj, Raahe, Finland during 15.8.2003 – 27.8.2003, and it consisted of 1326 steel strips and 59 variables from diagnostics measurements and 127 variables containing averaged values. Average values reflect product properties such as the composition of steel, target values, dimensions and oven identification number. Mill diagnostics time series information was gathered from roughing mill to

Manuscript received March 31, 2005. The research was funded by national technology agency of Finland, TEKES.

Jarno Haapamäki and Juha Röning are with Neurogroup / ISG, Department of Electrical and Information Engineering, University of Oulu, Finland (e-mail: {jarno.haapamaki, juha.roning}@ee.oulu.fi).

finishing mill temperature. The pre-processing and selection of variables were done using data sets consisting of over 15000 strips from two separate one-month periods. The diagnostics data consisted of temperatures, rolling forces and speeds, cooling water flows and pressure of the scale breaker.

The measurement interval in the time series data set varied between 0.0025s and 0.05s. Therefore, there are tens of thousands of measurements of high-frequency variables per strip. Hot strip rolling is quite a complex process from the point of view of measurements and data: During the process, a steel slab goes through plastic deformation, which increases its length between the points at which different variables are measured. Furthermore, rolling speed varies when it is used as a means to control temperature. Data synchronisation and allocation on the steel strip had to be solved before the data analysis. Therefore, the data were modified from time-indexed signals to a distance-based axis common for all variables.

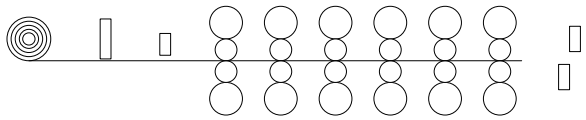

Fig. 1. Finishing mill layout.

The target variable, i.e. the scale defect count, was recorded between the last finishing stands and the cooling section. The video camera-based machine vision system used in surface quality detection, Parsytec ODIS 4.2, is shown on the right in Fig. 1. The machine vision system detects scale defects and indicates the exact location of each defect. There was a possible drawback, however, since only the count of defects was measured, but neither the size nor the severity of the defect area was recorded.

III. Data Mining and Modelling

The model should be continuously learning, since there was a significant difference in the number of scale defects between the data sets from different time periods. A prediction system should predict errors before they occur. Since most of the diagnostics measurements are done at the finishing stands, however, it is not possible to make predictions on-line in order to adjust the process parameters for the strip being rolled. Instead, the estimation given by the prediction system could be compared with the measured values. The outcome of the comparison could be used to facilitate decision-making concerning the mill set-up for the following strips or the need for service (predicting maintenance).

Careful data pre-processing is essential before the modelling phase. The application area involves several problems that restrict the usability of feature extraction methods. Firstly, the process contains some controversial properties. For instance, the surface temperature is affected by scale thickness, because the oxide layer reduces heat transfer [1]. Since the crop shear temperature is measured just before the scale breaker, the measurements may be inaccurate. Moreover, scale thickness is reduced in proportion to rolling reduction, but reduction increases simultaneously the amount of harder oxides and the probability of defects due to scale cracking. Secondly, the combination of variables is huge. Thirdly, the lack of severe process faults resulted in a small number of scale defects, which is a problem from the viewpoint of data mining. The approach was to collect data without process experiments, e.g. by reducing scale breaker pressure. However, intentionally generated fault situations would not have been useful when normal process conditions were modelled. Since the preconditions for modelling were quite demanding and due to the low scale content of the samples, the first task was to identify the variables with maximum influence on the amount of scale.

A. *Synchronisation*

Process state varies constantly and causes difficulties in modelling if there is no information about process dynamics available [10]. High-frequency measurements are necessary to find out the conditions within the entire strip length. Mill sensors are located at different positions along the production line, and measurements are therefore recorded at different times. Synchronisation of the variables and allocation of the measurements on the steel strip had to be solved before data analysis. There was a time stamp at the beginning of each time series, but since the measurement series was triggered before the strip arrived at the appropriate position, the indication of which part of the measurement series was within the length of the strip had to be solved. Feature extraction techniques (e.g. derivation) were used for temperature and rolling force signals to locate the leading edge of the strip. Furthermore, the strip edge extracted from the rolling force signals was used to proportionate the strip speed between stands to the threading speed, which was an absolute value. Measured speeds at various locations were synchronised by searching for the maximum correlation between signals at different lag values. Based on the knowledge of the speed over strip length and the time of the leading edge of the strip, the other measurements could be located. Each variable value is presented as a (measurement, position) pair for subsequent utilisation in modelling.

B. *Feature Extraction and Variable Count Reduction*

Measurement-position pairs were further processed by calculating features for each 10m section by means of descriptive statistics and Fourier transform. The features were mean, standard deviation, minimum, maximum, derivative, energy at a frequency interval of 1-5Hz and 5-20Hz and maximum amplitude with a related frequency between 1-25Hz. Furthermore, the absolute magnitude of scale defects was proportioned to the area size of the strip segment.

The features had to be combined with the scale defect information in order to find out the critical conditions for scale formation. A SOM [11] was used to visualize the dependencies of the variables on scale defect by mapping variable combinations with scale defect. SOM is able to

visualise non-linear dependencies, and it has been used in many industrial applications e.g. to model product quality in steel rolling [12,13]. Redundant and non-affecting variables were removed based on visualisation. As a result, the number of variables was reduced from 186 to 10 at the modelling phase.

C. Analysis and Scale Defect Prediction

The behaviour of scale at a roll gap is difficult to predict [14]. Thus, the use of machine learning modelling methods instead of physical models is justified. Predictive modelling has been done earlier by using neural networks [9]. The uneven target variable distribution, which is a consequence of the small number of defects, affects learning negatively. Training algorithms aim to minimise the average error, which leads to poor generalisation on rare occasions. However, biasing of the distribution was not possible without wasting too much data. Thus, only the number of samples without defects was limited in some analyses. Both GA and MLP suffered from the unbalanced data, which is why the output had a limited range near zero. However, there are possibilities to compensate for the "zeroing effect". GA can be based on freely selected cost function and they can utilise various penalizing functions [16]. Although the particular application allowed direct calculation of cost function, it is possible to penalize a solution candidate when, for instance, a solution is impossible (e.g. a negative number of defects). In this application, negative values were penalized. In addition, if the outcome of the equation was too close to zero for the entire training set, the similar action was applied. Furthermore, excessively complicated solutions were penalized.

A chromosome structure consists of variables with a multiplier factor and arithmetic basic operations between terms. The following operators were implemented: addition, multiplication, division, exponent function and special no-operation, which ends the formula. Since no single variable was known to have a linear effect on the modelled function, the formula length was limited to be between 2 and 10 terms. The genotype was coded with real valued numbers. A chromosome can be written $k_1m_1o_1$, $k_2m_2o_2$, ..., $o_9k_{10}m_{10}$, where k_i is a scaling factor, m_i is a variable and o_i is a operator. The other option would have been a variable length chromosome, which has been suggested by Kotani *et al.* [17].

In this application the operator genes select the program path to be executed to calculate a cost function, which is based on RMS error in this case. This resembles genetic programming [18], although the code structure is static, lacking code evolution.

A standard set of genetic operators with minor modifications were used in the algorithm. Selection was based on combined fitness calculated from a cost function and from chromosome diversity. Diversity was measured as a sum of Euclidean distances to all neighbours.

The real value coded genes were not the lowest order presentation for genes unlike the binary coded ones. Nevertheless, the effect of the schema theorem [6] has been put into practice by selecting the crossover points so that variable and scaling factor constitute variable length groups which are kept together in the calculations. Therefore a structured information change takes place in the reproduction. Good structures will have increasing numbers of samples in the subsequent generations as shown in [6,19]. However, without exchanging the locations of those building blocks, some of the power goes unused, since the different good solutions can compete for the same location in a string. Furthermore "super building blocks" could occupy several locations in distinct samples, restricting the production of other good solutions. The reason to code a string as a combination of separate real valued blocks is to avoid the bit position effect. Furthermore, the use of real values instead of binary coding allows the search space to increase.

The range of gene values was not restricted within the range of the initial population, since the selected crossover method can extend the population range occasionally. Therefore, new areas were searched and the population diversity was maintained. The value of the offspring gene o at time t+1 was calculated from the equation:

$$o(t+1) = \begin{cases} 0{,}5\,p_1(t) + 1{,}5\,p_2(t) \\ 1{,}5\,p_1(t) + 0{,}5\,p_2(t) \\ 0{,}5\,p_1(t) + 0{,}5\,p_2(t) \end{cases} \quad (1)$$

Actually, it became necessary to limit the excessive divergence of gene values by limiting maximum diversion. In addition, too similar chromosomes were deleted from the population before fitness calculations.

Population size was varied between 100 and 1000 in the modelling efforts. A continuously learning version was also tested. The model was trained with the data from 10-200 consecutive strips at a time and after each training period, the next strip in the sequence was modelled. To test the algorithm, an artificial data set consisting of three random variables was generated. The fit between the estimation and the generated value from function $f = 0.5\exp(2x) - 1.5y^3 - z$ is shown in Fig 2.

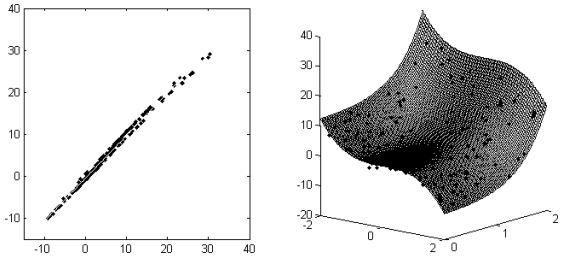

Fig. 2. Test function correlation with estimation on the left and estimated values with reference to the surface at z=0 on the right.

IV. Results

One steel quality, which was studied for rolled-in scale, was selected as an illustrative example. On an average the number of rolled-in-scale observations was $0.0488/m^2$ for this quality.

Independent training and testing sets were selected in such a way that there could not be samples from the same strip in both sets. The "alarm rate", selected for convenience and unrelated to the rejection risk, was set at 0.5 observations/m^2.

Different numbers of parameters and various combinations were tested. The most promising variables found with SOM in the data mining phase were used in the GA analysis. The average error in the test data was 0,0957 [1/m^2], which is almost equal to the value achieved with MLP. There are many other formulas that are almost as good as the one that gave the best result. One further possibility would be the study the frequencies of various building blocks.

When continuous learning was applied, the testing error was reduced to 0.0640. However, the result is not as good as that obtained with SOM, which can be seen in Fig. 3, where estimations have been plotted on (measurement, estimate) space. The alarm rate is not reached although there are 81 measurements above it. Furthermore, the interpretability of the model decreases, since the on-line learning model is a product of a short period. However, the history in the form of the best genotype for each time step was recorded, which made it possible to search for similarities and to study the learning. Furthermore, the history knowledge was applied to the initialisation of a new population in continuous learning. The test errors of GA are compared to the multilayer perceptron, MLP [15] and SOM-type neural networks in table I.

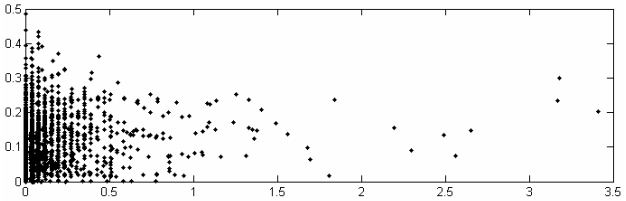

Fig. 3. Measured (on x –axis) versus estimated (on y –axis) values.

TABLE I
COMPARISON OF RESULTS

Method	Test error	Training type
GA	0.0640	continuous
GA	0.0957	batch
SOM	0.0597	batch
MLP	0.1032	batch

V. CONCLUSION

The data synchronisation was successfully accomplished before the data analysis. The focus in the modelling was on the quantity of scale after the finishing mill, since the available scale defect inspection system detected scale only at that stage. Modelling of the entire scale formation procedure would have required scale detection during the whole rolling process

Andorfer et al. pointed out in their article [6] that the classification accuracy of the surface inspection system is questionable. Indeed, the performance of the scale detection system was highly dependent on the environmental conditions.

As a result it can be stated that scale content modelling is possible using high-frequency data synchronisation. The interpretability of the solution made the GA based model a feasible tool for extracting knowledge, even though the results obtained with real world data were not as good as expected, although the continuous learning was applied.

Since the proposed GA method is applicable to continuous functions and the data are divided in different and probably even non-continuous functions, there is a challenge for further development.

REFERENCES

[1] P.H. Bolt, F. Friedel, H. Pircher, X. Cornet, S. Ehlers, and F. Steinert, "Investigation of the formation, constitution and properties of scale formed during the finishing rolling, cooling and coiling of thin hot strips," ECSC steel RTD project tech. rep.

[2] T. Fukagawa, H. Okada, and Y. Maehara, "Mechanism of red scale defect formation in Si-added hot-rolled steel sheets," *Tetsu-to-Hagane (Iron and Steel Institute of Japan)*, vol. 34 no. 11, 1994, pp. 906-911.

[3] V. Ginzburg, *Steel-rolling technology: theory and practice.* Marcel Dekker Inc. New York ,1989.

[4] *Surface defects in hot rolled flat steel products.* Verlag Stahleisen Gmbh, 1996.

[5] J. Andorfer, D. Paesold, R. Puntigam, K. Rendl, J. Zeindl, D. Auzinger, G. Hubmer, "New achievements in quality control systems for hot band," *METEC 2003, 6th international metallurgy trade fair.*

[6] J.H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. The University of Michigan Press, Ann Arbor, 1975.

[7] R.Y. Chen, and W.Y.D. Yuen, "Effects of finishing and coiling temperatures on the scale structure and picklability of hot rolled strip," *Iron and Steelmaker,* vol. 27 no. 4, 2000, pp. 47-53.

[8] K.S. Tan, M. Krzyzanowski, and J.H. Beynon, "Effect of Steel Composition on Failure of Oxide Scales in Tension under Hot Rolling Conditions," *Steel Research,* vol. 72 no. 7, 2001, pp. 250-258.

[9] J. Haapamäki, S. Tamminen, and J. Röning, "Data mining methods in scale defect prediction for hot steel rolling," in *Proc. 23rd IASTED Int. Conf. Artificial Intelligence and Applications, Feb.* 2005 pp. 90–94

[10] Y. Bisseur, E.B. Martin, A.J. Morris, and P. Kitson, "Fault detection in hot steel rolling using neural networks and multivariate statistics," *IEE Proc. Control Theory and Applications*, 147(6), 2000, pp. 633-640.

[11] T. Kohonen, *Self organizing maps.* Springer, Berlin, 1995.

[12] E. Alhoniemi, J. Hollmén, O. Simula, and J. Vesanto, "Process monitoring and modeling using the self-organizing map," *Integrated Computer-Aided Engineering*, vol. 6 no. 1, 1999, pp. 3-14.

[13] L. Cser, A.S. Korhonen, J. Gulyás, P. Mäntylä, O. Simula, Gy. Reiss, and P. Ruha, "Data mining and state monitoring in hot rolling," *Proc. of the 2nd Int. Conf. on Intelligent Processing and Manufacturing of Materials,* Honolulu, Hawaii, July 10-15, 1999, Vol. 1, IEEE, pp. 529-537.

[14] M. Krzyzanowski, and J.H. Beynon, "The tensile failure of mild steel oxides under hot rolling conditions," *Steel Research* vol. 70 no. 1, 1999, pp. 22-27.

[15] S. Haykin, *Neural networks: a comprehensive foundation.* Macmillan Publishing Company, 1994.

[16] Z. Michalewicz, *Genetic algorithms + data structures = evolution programs.* 3. ed. 1996 Springer-Verlag.

[17] M. Kotani, M. Ochi, S. Ozawa, and K. Akazawa, "Evolutionary discriminant functions using genetic algorithms with variable-length chromosome," *Proc. Int. joint conference on Neural Networks, 2001,* IJCNN pp. 761-766.

[18] J.R. Koza, Genetic programming: "A paradigm for genetically breeding populations of computer programs to solve problems," Stanford University, 1990. Available http://www.genetic-programming.com/jkpdf/tr1314.pdf

[19] D.E. Goldberg, *Genetic algorithms in search optimization and machine learning.* Addison-Wesley, 1989.