COBEM 99
XV Congresso Brasileiro de Engenharia Mecânica
15th Brazilian Congress of Mechanical Engineering
22 - 26 de Novembro de 1999 / November 22 - 26, 1999 Águas de Lindóia, São Paulo,

NEURAL NETWORK REPRESENTATION OF ALEXANDER'S MODEL FOR THE ROLLING PROCESS

L.E. Zárate, Dr.
Deptº de Ciências Quânticas, FCG-UNA
Belo Horizonte - MG, Brasil, E-mail: zarate@brhs.com.br
H. Helman, Ph. D.
Deptº de Engenharia Metalúrgica e de Materiais, EEUFMG
Belo Horizonte-MG, Brasil. E-mail: hhelman@demet.ufmg.br
J.M. Gálvez, Ph.D.
Deptº de Engenharia Mecánica, EEUFMG
Belo Horizonte-MG, Brasil. E-mail: jmgm@dedalus.lcc.ufmg.br

Abstract. *The rolling process mathematical modeling involves nonlinear parameters and relationships that usually lead to nonlinear equations of difficult numerical solution. Such is the case of Alexander's model (1972), considered one of the most complete regarding the rolling theory. For simulation purposes, Alexander's model requires too much computational time, which prevents its use in on-line control and supervision systems.*
In this work, two neural network structures are trained using process and operation data respectively, generated by Alexander's models. The neural models are validated through simulation. Finally, the neural network models are used to obtain the sensitivity factors of the process by differentiating the network outputs.
It is shown that the new neural network representations allow to obtain process equations for different operation points. Results of the representations are presented.

Keywords: Steel industry, Rolling process, Neural networks

1. INTRODUCTION

The existent theoretical models for the rolling process allow to calculate the rolling load by unit width (P) and torque (T_q) of non-linear expressions as: Eqs. (1) and (2).

$$P = f(h_i, h_o, t_b, t_f, \mu, \bar{y}, E, R) \tag{1}$$

$$T_q = f(h_i, h_o, t_b, t_f, \mu, \bar{y}, E, R) \tag{2}$$

with $h_i =$ input thickness; $h_o =$ output thickness; $t_b =$ back tension; $t_f =$ front tension; $\bar{y} =$ average yield stress; $\mu =$ friction coefficient; E =Young modulus of the strip; R= roll radius.

It is convenient to have equations to calculate the outgoing thickness and the rolling load, Eq. (3), in terms of the other parameters. This will permit to analyze the effect of disturbances in the entry thickness, front or back tensions, average yield stress and friction coefficient on the rolling load and, therefore, on the outgoing thickness.

$$(h_o, P_w) = f\,(W, g, M, \bar{y}, h_i, t_b, t_f, \mu, E, R) \qquad (3)$$

where: W = strip width; M = rigidity rolling mill modulus; g = roll gap.

Equations. (1), (2) and (3) involve non-linear parameters and relationships that usually lead to non-linear equations of difficult numerical solution. Such is the case of the Alexander's model (1972), considered as one of the most complete in the rolling theory. For simulation purposes, Alexander's model requires a significant computational effort, which prevents its use in on-line control and supervision systems.

In Eqs. (1), (2) and (3), the rolling load value depends on the output thickness value and vice versa. Therefore, an "algebraic loop" exists in Eq. (3) that prevents the analytical calculation of those parameters.

To solve this problem, a numerical solution involving successive iterations can be used. This procedure may demand a great computational effort in order to calculate the new operation point when a disturbance takes place in the rolling process. The computational time varies according to the different operation points. This prevents the use of this type of numerical solution in on-line control and supervision systems.

Other forms to represent the rolling process use neural networks and sensitivity factors.

In this paper, two neural network representations for the cold rolling process based on Alexander's model, are presented.

In section 2, the neural network based representation of the rolling process, is presented. In section 3, the calculation of the sensitivity equations through the differentiation of the neural network is shown. In section 4, the neural models are validated through simulation. Finally, conclusions of the representations are discussed.

2. REPRESENTATION OF THE ROLLING PROCESS THROUGH ARTIFICIAL NEURAL NETWORKS

Artificial neural networks (ANN) have been the focus of great deal of attention during the last decade, due to their capabilities in solving non-linear problems by learning (Hunt et. al., 1992 and Sbarbaro-Hofer et.al., 1993). Actually, these nets are presently widely used in metallurgical process, as in Andersen, et. al. (1992), Smart (1992), Zárate (1998) and Zárate et. al. (1998a,b).

In this section, two representations for the rolling process (Eqs. (1) and (2)) and rolling mill operation (Eq. (3)) through ANN, are presented.

2.1 Representation of the rolling process

The set of parameters to represent the rolling process is given by:

$$(h_i, h_o, \mu, t_b, t_f, \bar{y}) \xrightarrow{NeuralNetwork} (P, T_q) \qquad (4)$$

Figure 1, shows the structure to represent the rolling process by an ANN.

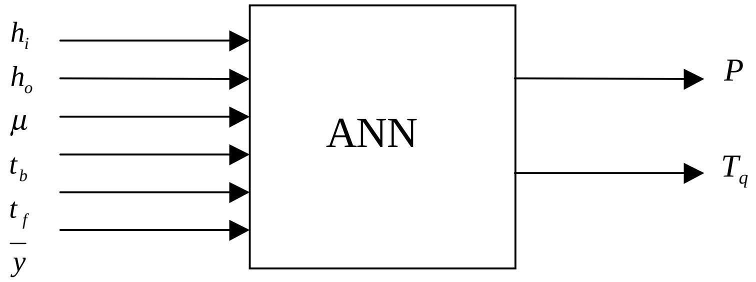

Figure 1- Structure to represent the rolling process by an ANN

The representation has 6 entries, generating 729 training sets, assuming that each entry may have three different values. Alexander's model (Alexander, 1972) was used to generate a database for the cold rolling mill process.

2.2 Representation of the rolling mill operation.

The set of parameters to represent the rolling mill operation is given by:

$$(h_i, g, \mu, t_b, t_f, \bar{y}) \xrightarrow{NeuralNetwork} (h_o, P_w) \tag{5}$$

Note that the rolling load (P_w), the strip width (W) and the roll gap (g) values are related to output thickness through the elastic equation of the mill, Eq. (6).

$$h_f = g + \frac{P.W}{M} \tag{6}$$

Figure 2 shows the structure to represent the rolling mill operation by an ANN.

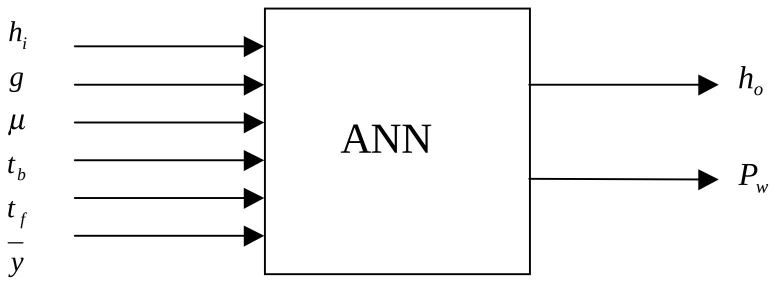

Figure 2- Structure to represent the rolling mill operation by an ANN

Observe that the representation of the rolling mill operation considers an ANN trained in the neighborhood of an operation point, assuming the mill modulus and strip width as constants.

Observe that the representation of the rolling mill operation considers the roll gap as an entry. This entry is calculated through Eq. (6), considering the mill modulus and strip width as constants for all the training sets. This constitutes a limitation for the representation, since the strip width is a dependent variable.

The introduction of the strip width as a new entry in the ANN, results in 7 entries, generating 2187 training sets, assuming that each entry may adopt three different values.

When some deviation from the operation point (not considered in the network training) is expected, an adaptive configuration may be used in order to operate the on-line representation. In this case, the neural network will remain training, aiming at updating the weights. Normally, the computational time involved in training an ANN is big and a new entry can prevent its use in on-line control and supervision systems, if an adaptive structure is used.

A strategy that permits to consider the strip width as an independent variable represents the output thickness (h_o) as a function of the sensitivity factors: Eqs. (7) and (8).

$$\Delta P = \frac{\partial P}{\partial h_i}\Delta h_i + \frac{\partial P}{\partial h_f}\Delta h_f + \frac{\partial P}{\partial t_r}\Delta t_r + \frac{\partial P}{\partial t_f}\Delta t_f + \frac{\partial P}{\partial \mu}\Delta \mu + \frac{\partial P}{\partial \overline{y}}\Delta \overline{y} \tag{7}$$

$$\Delta Tq = \frac{\partial Tq}{\partial h_i}\Delta h_i + \frac{\partial Tq}{\partial h_f}\Delta h_f + \frac{\partial Tq}{\partial t_r}\Delta t_r + \frac{\partial Tq}{\partial t_f}\Delta t_f + \frac{\partial Tq}{\partial \mu}\Delta \mu + \frac{\partial Tq}{\partial \overline{y}}\Delta \overline{y} \tag{8}$$

As the rolling mill is not perfectly rigid, the outgoing thickness is influenced by the elastic equation of the rolling mill Eq. (6). Any variation in the roll gap can be expressed as:

$$\Delta h_f = \Delta g + \frac{\Delta P.W}{M} \tag{9}$$

Combining the Eqs. (7) and (9), the expression to calculate the outgoing thickness, Eq. (10) can be obtained. This equation requires the calculation of the sensitivity factors.

$$\Delta h_o = \frac{W}{M - W\frac{\partial P}{\partial h_o}}\{\frac{M}{W}\Delta g + \frac{\partial P}{\partial h_i}\Delta h_i + \frac{\partial P}{\partial tr}\Delta t_b + \frac{\partial P}{\partial t_f}\Delta t_f + \frac{\partial P}{\partial \mu}\Delta \mu + \frac{\partial P}{\partial y}\Delta y\} \tag{10}$$

where $\Delta h_o = h_o^* - h_o$

Normally, the sensitivity factors are calculated from equations of difficult analytical solution, as in Alexander's model. In this paper, the sensitivity factors are calculated through the differentiation of a neural network. To that aim, the neural network based representation of the rolling process (Eq. (4)) is used.

In this case, a back-propagation neural network with six inputs (N=6), two outputs (M=2) and one hidden layer with 13 neurons (2N+1), is used. A sigmoid function was selected as the activating function.

Generally, the largest effort to train a neural network lies on collecting and pre-processing the input data. The pre-processing operation consists in the normalization of the data, in such a way that the inputs and outputs values be within the 0 to 1 range.

The following procedure was adopted to normalize the input data before using it in the ANN structure:

- In order to improve convergence of the ANN training process, the normalization interval [0, 1] was reduced to [0.2, 0.8].
- The data was normalized through the following formula:

Ln = (Lo - Lmin) / (Lmax - Lmin) (11)

Where Ln is the normalized value, Lo value to normalize, Lmin and Lmax are minimum and maximum variable values, respectively.

- Lmin and Lmax were computed as follows:

Lmin = (4 x LimiteInf. - LimiteSup) / 3 (12)
Lmax = (LimiteInf. - 0.8 x Lmin) / 0.2 (13)

3. SENSITIVITY EQUATIONS

In this section, the calculation of the sensitivity factors through the differentiation of a general neural net, with N entries, M exits and L neurons in the hidden layer, is presented.

The currently used symbols are:

$U_i, \; i = 0,...,N$ are the net entries and $U_0 = 1$ is a polarization entry

$f_i^a(.) \; i = 0,...,N$ are the normalization entry functions and $f_0^a(.) = 1$

$X_i, \; i = 0,...,N$ are the normalized entries $X_0 = U_0$

$W_{ij}^h \; i = 1,...,L$ e $j = 0,...,N$ is the weight corresponding to the neuron i and entry j

$net_j^h = \sum_{i=0}^{N} W_{ji}^h X_i \quad j = 1,...,L$ are product of weights times entries

$f_j^h(net_j^h) \quad j = 0,...,L$ with $f_0^h(net_0^h) = 1$ is the sigmoid function of the hidden layer.

$I_j, \; j = 0,...,L$ are the corresponding values of the sigmoid function $I_0 = 1$

$W_{ij}^o \; i = 1,...,M$ e $j = 0,...,L$ is the weight of the neuron i and entry j for the hidden layer.

$net_j^o = \sum_{i=0}^{L} W_{ji}^o I_i \quad j = 1,...,M$ are product of the weights times entries for the hidden layer.

$f_j^o(net_j^h) \quad j = 1,...,M$ is the value of the sigmoid function for the exit layer

$Y_j, \; j = 1,...,M$ are the normalized exits of the net, obtained from the sigmoid function

$f_i^b(.) \; i = 1,...,M$ are the denormalization functions of the exits

$Z_i, \; i = 1,...,M$ net exits values

$e\max_k, e\min_k \; k = 1,..,N$ higher and lower value of the entries

$s\max_k, s\min_k \; k = 1,..,M$ higher and lower value of the exits

The procedure to obtain the expressions of the net sensitivity will now be described. Be

$$\begin{aligned} Z_1 &= f_1^b(Y_1) \\ Z_2 &= f_2^b(Y_2) \\ &\vdots \\ Z_M &= f_M^b(Y_M) \end{aligned} \tag{14}$$

By substituting the corresponding values for the functions $f^a(.), f^b(.), f^o(.), f^h(.)$, Eq. (15) is obtained:

$$Z_k = \frac{1}{1 + \exp^{-V_k}} \left[\text{smax}_k - \text{smin}_k \right] + \text{smin}_k \tag{15}$$

para k = 1,.., M

where:

$$V_k = \sum_{j=0}^{L} W_{kj}^{o} f_j^{h} \left(\sum_{i=0}^{N} W_{ji}^{h} f_i^{a}(U_i) \right) \quad \text{para } k = 1, \ldots, M \tag{16}$$

The sensitivity factors will be calculated from Eq. (17):

$$\frac{\partial Z}{\partial U} = \begin{bmatrix} \frac{\partial Z_1}{\partial U_1} & \frac{\partial Z_1}{\partial U_2} & \cdots\cdots & \frac{\partial Z_1}{\partial U_N} \\ \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots \\ \frac{\partial Z_M}{\partial U_1} & \frac{\partial Z_M}{\partial U_2} & \cdots\cdots & \frac{\partial Z_M}{\partial U_N} \end{bmatrix} \tag{17}$$

where each term of the sensitivity matrix is calculated in the form:

$$\frac{\partial Z_k}{\partial U_i} = [\text{smax}_k - \text{smin}_k] \frac{\exp^{-V_k}}{(1 + \exp^{-V_k})^2} \frac{\partial V_k}{\partial U_i} \tag{18}$$

Manipulating the derivative term of Eq. (18) and taking into account Eq. (16), the following expression is obtained:

$$\frac{\partial V_k}{\partial U_i} = \frac{\partial}{\partial U_i} \left(W_{k0}^{o} + \sum_{j=1}^{L} W_{kj}^{o} f_j^{h} \left(\sum_{i=0}^{N} W_{ji}^{h} f_i^{a}(U_i) \right) \right) \tag{19}$$

By differentiating Eq. (20) and substituting that expression in Eq. (18), the Eq. (20) is obtained, which allows to calculate the sensitivity factors starting from the net:

$$\frac{\partial Z_k}{\partial U_i} = \begin{bmatrix} R_1 W_{11}^{o} & R_1 W_{12}^{o} & \cdots & R_1 W_{1L}^{o} \\ R_2 W_{21}^{o} & R_2 W_{22}^{o} & \cdots & R_2 W_{2L}^{o} \\ \vdots & \vdots & \vdots & \vdots \\ R_M W_{M1}^{o} & R_M W_{M2}^{o} & \cdots & R_M W_{ML}^{o} \end{bmatrix} \begin{bmatrix} \frac{Q_1 W_{11}^{h}}{\text{emax}_1 - \text{emin}_1} & \frac{Q_1 W_{12}^{h}}{\text{emax}_2 - \text{emin}_2} & \cdots & \frac{Q_1 W_{1N}^{h}}{\text{emax}_N - \text{emin}_N} \\ \frac{Q_2 W_{21}^{h}}{\text{emax}_1 - \text{emin}_1} & \frac{Q_2 W_{22}^{h}}{\text{emax}_2 - \text{emin}_2} & \cdots & \frac{Q_2 W_{2N}^{h}}{\text{emax}_N - \text{emin}_N} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{Q_L W_{L1}^{h}}{\text{emax}_1 - \text{emin}_1} & \frac{Q_L W_{L2}^{h}}{\text{emax}_2 - \text{emin}_2} & \cdots & \frac{Q_L W_{LN}^{h}}{\text{emax}_N - \text{emin}_N} \end{bmatrix} \tag{20}$$

with:

$$Q_k = \frac{\exp^{-\left(\sum_{i=0}^{N} W_{ki}^{h} X_i\right)}}{\left(1 + \exp^{-\left(\sum_{i=0}^{N} W_{ki}^{h} X_i\right)}\right)^2} \qquad k = 1, \ldots, L \tag{21}$$

$$R_k = (\text{smax}_k - \text{smin}_k) \frac{\exp^{-V_k}}{(1 + \exp^{-V_k})^2} \qquad k = 1, \ldots, M \tag{22}$$

$$V_k = \sum_{j=0}^{L} W_{kj}^{o} f_j^{h} \left(\sum_{i=0}^{N} W_{ji}^{h} f_i^{a}(U_i) \right) \quad \text{para } k = 1, \ldots, M \tag{23}$$

The Eq. (20) provides a linear form of the equation process, in the neighborhood of an operation point (Ui) and it is valid for small variations in the process parameters. Notice that the sensitivity factors are calculated directly from the network inputs and the net weights.

4. SIMULATION RESULTS

To asses the performance of the proposed representations, Alexander's model (Alexander 1972) was used to generate a database for the cold rolling mill process. To obtain the data sets for ANN training, the parameters variations were chosen as: $h_i = \pm 8\%$; $h_o = \pm 3\%$; $\mu = \pm 20\%$; $t_f = \pm 30\%$; $t_b = \pm 30\%$ and $\bar{y} = \pm 10\%$. Three different values were chosen for each parameter resulting in 729 training sets. The load rolling was obtained through Alexander's model and the roll gap by the elastic equations for the rolling mill, Eq. (6).

The nominal values of the parameters for the rolling process and rolling mill operation were chosen as: h_i = 5.0 mm; h_o, = 3.6 mm; g = 1.846 mm; μ = 0.12; t_f = 9.098 kgf/mm^2; t_b=0.441 kgf/mm^2; $\bar{y}$ =46.918 kgf/mm^2; W = 500 mm; E=20,400 kgf/mm^2; R = 292.1 mm; M=500,000 kgf/mm ; P= 875.31tf and $y = 26.138 + 47.742\varepsilon^{0.4275}$.

The final weights for the hidden and output layers, with its polarization weight for the rolling process and rolling mill operation are:

Case 1: Representation of the rolling process

$$W^h = \begin{bmatrix} -2.3472 & 0.7327 & -1.6876 & -0.0036 & 0.6561 & -2.6032 \\ 3.9055 & 3.0808 & -5.1249 & -2.8544 & -5.3232 & -5.6772 \\ -8.4711 & 5.1593 & 6.1539 & -1.5899 & -4.3587 & 3.4279 \\ -11.7598 & 0.2278 & -0.9994 & -0.3439 & 0.1404 & 6.6674 \\ -8.4249 & -9.2360 & -5.5525 & 1.1392 & 2.4098 & -5.1437 \\ -1.3988 & 1.6645 & 9.0436 & 0.0068 & 5.0951 & -7.2585 \\ 3.4416 & 0.4525 & 8.6002 & -6.6928 & 5.2036 & 1.8066 \\ -2.0861 & 0.6307 & -0.6206 & 0.0006 & 0.8530 & -1.8555 \\ -4.1920 & 0.5245 & -4.9311 & 7.3975 & 1.3536 & 6.6874 \\ 10.5998 & 3.1857 & -4.8274 & 1.8625 & -3.3112 & -2.8462 \\ 3.8174 & 13.3982 & -3.9206 & -1.5036 & -1.7811 & 0.6947 \\ 3.9012 & -2.6002 & 0.5488 & 2.1261 & -5.7294 & -5.7072 \\ 8.5121 & -6.7925 & -2.3503 & 2.8321 & 4.0080 & -3.0901 \end{bmatrix} \quad W^h_{bias} = \begin{bmatrix} 5.2510 \\ 6.0337 \\ -3.8967 \\ 5.1585 \\ 5.4216 \\ -1.0687 \\ -7.2570 \\ 0.2414 \\ -3.2777 \\ -0.9663 \\ -5.8385 \\ 3.7317 \\ -3.0088 \end{bmatrix} \quad W^o = \begin{bmatrix} -3.1772 & -2.9064 \\ -0.0878 & 0.0705 \\ -0.0626 & -0.0884 \\ 0.1237 & 0.0306 \\ 0.0126 & 0.0401 \\ 0.0767 & -0.0340 \\ 0.0418 & -0.0196 \\ -2.4975 & -2.2320 \\ 0.0412 & -0.0437 \\ -0.0568 & 0.0361 \\ 0.0205 & 0.0008 \\ -0.1528 & 0.1381 \\ -0.0468 & 0.0634 \end{bmatrix} \quad W^o_{bias} = \begin{bmatrix} 3.2107 \\ 2.8861 \end{bmatrix}$$

Figures 3 and 4 compare the results of the operation of the ANN trained with the data obtained from Alexander's model. The global error was 0.039 after 555.000 iterations.

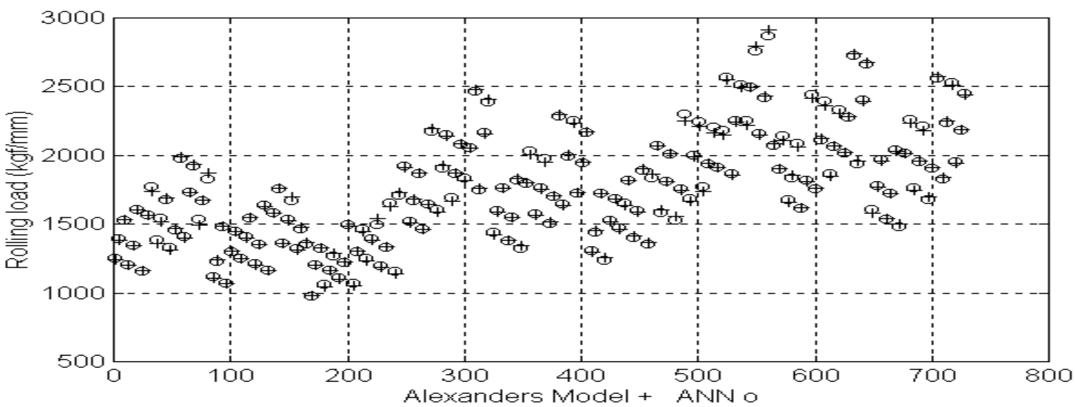

Figure 3- Comparison between ANN and Alexander's Model (case 1) for Rolling Load

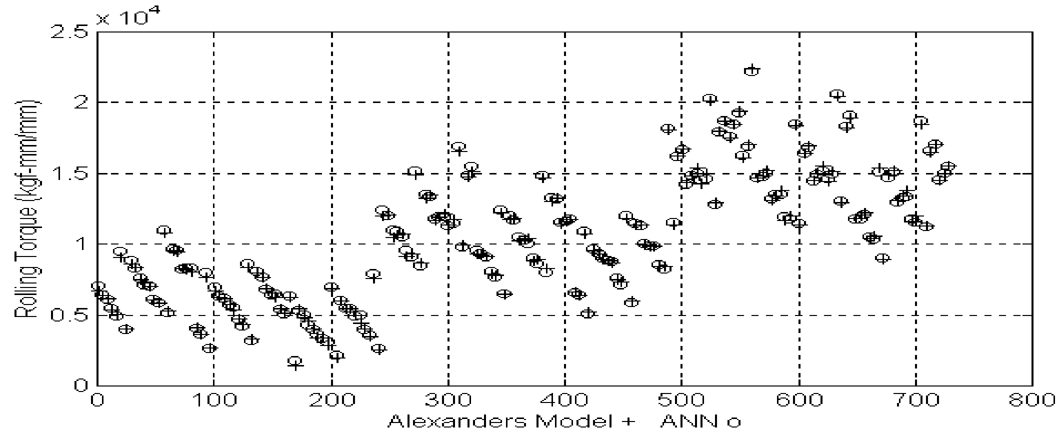

Figure 4- Comparison between ANN and Alexander's Model (case 1) for Rolling Torque

For the nominal operating point, the error in the rolling load was 0.64% and in the rolling torque 0.90%.

Case 2: Representation of the rolling mill operation

$$W^h = \begin{bmatrix} 6.2010 & 9.4161 & -1.7278 & -0.2642 & 1.8475 & 1.3078 \\ -1.5239 & -9.2425 & -2.4238 & 0.0734 & 0.7813 & -6.0229 \\ -11.4365 & -1.7129 & 1.2562 & -0.2171 & -0.6941 & 3.3125 \\ -0.2767 & -8.3214 & -5.5920 & -0.2737 & -3.3595 & -0.8293 \\ 3.3504 & 5.2707 & 3.4006 & 0.0343 & -0.5938 & 4.7363 \\ 8.9832 & 3.9726 & 0.7990 & 9.7730 & 0.9096 & 5.5016 \\ -6.1621 & 3.4810 & 7.4542 & 0.5558 & -7.0447 & -0.9006 \\ -4.8199 & -3.2388 & 6.5208 & -3.4559 & -0.6086 & 8.9722 \\ 0.8914 & -8.2071 & -7.6049 & -0.1518 & -2.2323 & -1.5593 \\ -1.2378 & -10.6099 & -0.0338 & 0.0399 & 0.4850 & -2.1458 \\ 8.3913 & 2.3521 & 5.9967 & -8.1868 & -4.0553 & 1.8826 \\ -1.4947 & -11.9576 & -1.6808 & 0.0733 & 0.7387 & -4.2602 \\ 2.9411 & 4.8319 & -11.5151 & -1.5484 & -4.0985 & -4.2330 \end{bmatrix} \quad W^h_{bias} = \begin{bmatrix} -11.4784 \\ 8.9265 \\ 2.5654 \\ 10.9804 \\ -10.4911 \\ -15.1094 \\ 5.3887 \\ -4.5520 \\ 7.1306 \\ 8.8376 \\ -0.4583 \\ 8.7774 \\ 2.5081 \end{bmatrix} \quad W^o = \begin{bmatrix} 0.5503 & 0.4083 \\ -1.7551 & -2.1593 \\ -0.9245 & -0.1960 \\ -0.2976 & 0.2163 \\ 4.2976 & 0.9087 \\ 0.0021 & 0.0045 \\ 0.1479 & 0.1406 \\ 0.1378 & 0.0480 \\ -1.1542 & -0.0074 \\ -6.4508 & 1.0470 \\ -0.0276 & -0.0106 \\ -3.9372 & 2.4463 \\ 0.1814 & 0.0746 \end{bmatrix} \quad W^o_{bias} \begin{bmatrix} 6.4599 \\ -1.4106 \end{bmatrix}$$

Figures 5 and 6 compare the results of the operation of the ANN, trained with the data obtained from Alexander's model. The global error was 0.040 after 330.000 iterations.

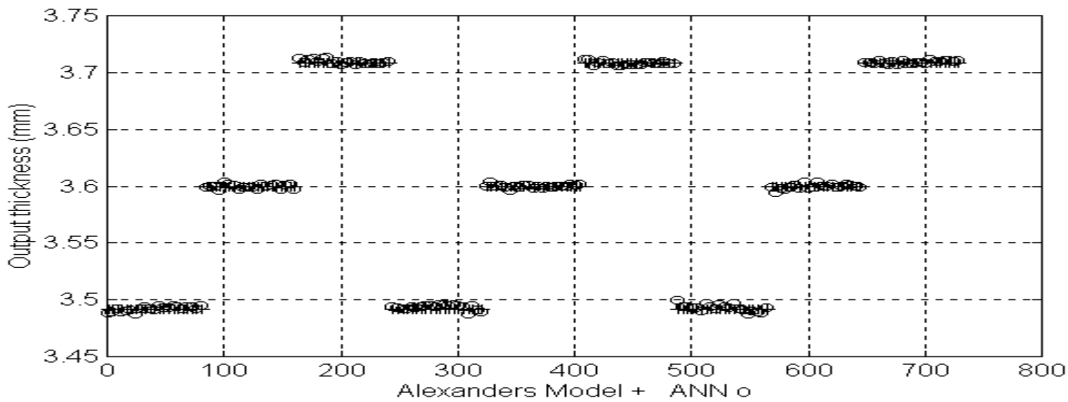

Figure 5- Comparison between ANN and Alexander's Model (case 2) for Output Thickness

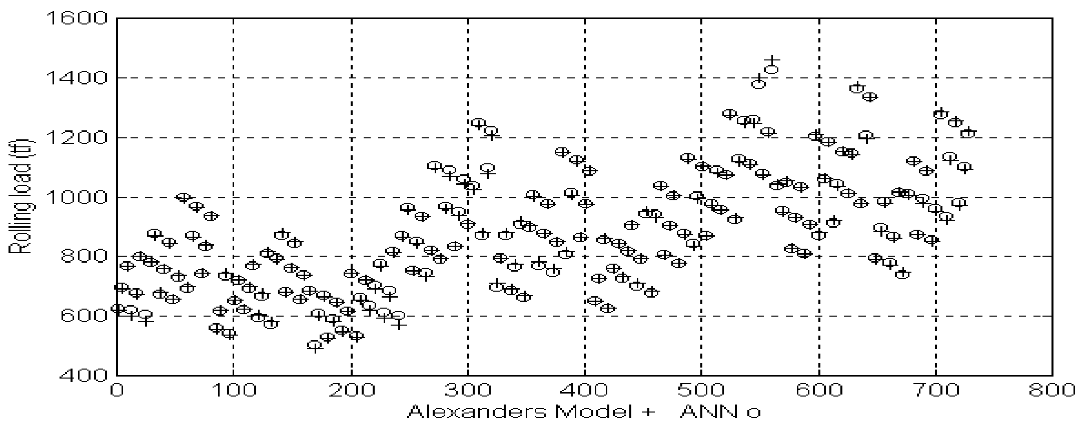

Figure 6- Comparison between ANN and Alexander's Model (case 2) for Rolling Load

For the nominal operation point, the error in the output thickness was 0.06% and in the rolling load 0.23%.

The representation of the rolling mill operation takes into account the strip width as a dependent variable. Another form, that permits to consider the strip width as an independent variable is Eq. (10), through the sensitivity factors and the neural network Eq. (4). Table 1 shows the nominal values for the sensitivity factors at the operation point, obtained by differentiating the neural network previously trained (Eq. (20)). Table 2, shows the output thickness obtained through the sensitivity factors for perturbations of 1, 3 e 5% in the parameters. In this case, the output thickness is kept with an error below 0.50%.

Table 1. Sensitivity factors

$\frac{\delta P}{\delta h_i}$	$\frac{\delta P}{\delta h_o}$	$\frac{\delta P}{\delta \mu}$	$\frac{\delta P}{\delta t_b}$	$\frac{\delta P}{\delta t_f}$	$\frac{\delta P}{\delta \bar{y}}$
498.66	-631.16	7909.25	-86.56	-5.83	56.75

Table 2. Output thickness obtained through the sensitivity factors

Var %	h_i	μ	t_b	t_f	$\bar{y}$	h_o ANN	h_o Alexander	$\lvert Err \rvert$ %
0,0	5,00	0,12	0,441	9,098	46,918	3,600	3,590	0,28
-1,0	4,95	-	-	-	-	3,585	3,572	0,36
+1,0	5,05	-	-	-	-	3,615	3,623	0,22
	-	0,119	-	-	-	3,595	3,588	0,19
	-	0,121	-	-	-	3,605	3,602	0,08
	-	-	0,436	-	-	3,600	3,599	0,03
	-	-	0,445	-	-	3,599	3591	0,22
	-	-	-	9,007	-	3,600	3,597	0,08
	-	-	-	9,189	-	3,599	3,594	0,14
	-	-	-	-	46,449	3,584	3,582	0,06
	-	-	-	-	47,387	3,616	3,606	0,28
-3,0	4,85	-	-	-	-	3,554	3,519	0,99
+3,0	5,15	-	-	-	-	3,646	3,670	0,65
	-	0,116	-	-	-	3,581	3,580	0,03
	-	0,124	-	-	-	3,619	3,613	0,17
	-	-	0,428	-	-	3,600	3,599	0,03
	-	-	0,454	-	-	3,599	3,591	0,22

Var %	h_i	μ	t_b	t_f	$\bar{y}$	h_o ANN	h_o Alexander	$\lvert Err \rvert$ %
	-	-	-	8,825	-	3,601	3,599	0,03
	-	-	-	9,371	-	3,599	3,591	0,22
	-	-	-	-	45,510	3,551	3,550	0,03
	-	-	-	-	48,26	3,647	3,632	0,41
-5,0	4,75	-	-	-	-	3,524	3,459	1,88
+5,0	5,25	-	-	-	-	3,676	3,723	1,26
	-	0,114	-	-	-	3,571	3,572	0,03
	-	0,126	-	-	-	3,629	3,621	0,22
	-	-	0,419	-	-	3,601	3,598	0,06
	-	-	0,463	-	-	3,599	3,592	0,19
	-	-	-	8,643	-	3,602	3,602	0,00
	-	-	-	9,553	-	3,598	3,589	0,28
	-	-	-	-	44,572	3,518	3,530	0,34
	-	-	-	-	49,264	3,681	3,659	0,60
-3,0	4,85	0,116	0,428	8,825	45,510	3,487	3,459	0,81
+3,0	5,15	0,124	0,454	9,371	48,326	3,713	3,733	0,54

In Table 2, for perturbations of 1% on the operational parameters, in the output thickness the minimum error was 0.03% and maximum error was 0.59%. When the deviations in the parameters increase to 3 and 5%, the error in the calculation of the output thickness is 2%. This is so because the method of the differentiation of the neural network to obtain the sensitivity factors behaves as Taylor's development (Zárate et. al. 1998b).

Notice that the smallest errors are present in the back tension, in the front tension and in the average yield stress, because the sensitivity factors in these parameters are, in absolute value, the smallest ones for the chosen operation point, as shown in Table 1.

4. CONCLUSIONS

In this paper two neural network and sensitivity factors based representations for the cold rolling systems through Alexander's model, were presented. The results show the good performance of both representations.

The calculated sensitivity factors were valid in the neighborhood of the operation point. However, it was observed that the results behave satisfactorily, even for large deviations from the operation point (Table 2). These neural network based representations can be used in on-line control and supervision systems, since the computational effort of the representation by neural networks and sensitivity factors is minimum (Zárate et.al. 1998c).

REFERENCES

Alexander, J.M., (1972). On the theory of Rolling. *Proc. R. Soc. Lond.* A. 326, pp. 535-563.

Andersen, K., Cook, G.E. and Barnett, R.J., (1992). Gas Tungsten Arc Welding Process Control Using Artificial Neural Networks, *International Trends in Welding, Science and Technology, ASM*, pp. 877-1030.

Hunt, K.J., Sbarbaro, D. Zbikowski and Gawthrop, P.J. (1992). Neural Network for Control Systems-A Survey. *Automatica*, Vol. 28, N° 6, pp. 1083-1112.

Sbarbaro-Hofer, D., Neumerkel, D. and Hunt, K., (1993). Neural Control of a Steel Rolling Mill, *IEEE Control Systems*, June, pp. 69-75.

Smartt, H.B. (1992) Intelligent Sensing and Control of Arc Welding. *International Trends in Welding, Science and Technology, ASM*, pp. 843-851.

Zárate, L.E., (1998). Um Método de Análise para Laminadores Tandem a Frio. Doctoral Thesis, Federal University of Minas Gerais, Belo Horizonte, MG, Brazil.

Zárate, L.E., J.M. Gálvez and H. Helman, (1998a). A Neural Network Based Controller Steel Rolling Mills by Using Sensitivity Functions. *XII CBA*, Vol. I, pp.29-34, September 14-18, Uberlândia, MG, Brazil.

Zárate, L.E.; Helman H. e Gálvez, J.M. (1998b). "Um Método para Linearização de Modelos Utilizando Redes Neurais e sua Aplicação em Processos de Laminação". VIII Congreso Latinoamericano de Control Automático", Vol. II, pp.709-714 - Nov. 9-13, Viña del Mar, Chile.

Zárate, L.E.; Song, M. e Helman H. (1998c). "Algoritmo para Linearização de Processos via redes neurais e sua análise de comportamento assintótico". V Simpósio Brasileiro de Redes Neurais, Dez. Belo Horizonte, Brasil, pp.199-204