Control and Cybernetics

Cold Rolling Mill Thickness Control Using The Cascade-Correlation Neural Network

by

Lipo Wang[1], Yakov Frayman[2] and Chunru Wan[1]

[1] School of Electrical and Electronic Engineering
Nanyang Technological University
Block S2, Nanyang Avenue, Singapore 639798

[2] School of Computing and Mathematics
Deakin University
Geelong VIC 3217 Australia
Email: elpwang@ntu.edu.sg
URL: http://www.ntu.edu.sg/home/elpwang/

Abstract:
The improvements in thickness accuracy of a steel strip produced by a tandem cold-rolling mill are of substantial interest to a steel industry. In this paper, we design a direct model-reference adaptive control scheme with a cascade-correlation neural network (CCNN) as a controller for a rolling mill thickness control which exploits the natural level of the excitation existing in the closed-loop. Simulation results show that the combination of a such a direct model-reference adaptive control scheme and the CCNN significantly improves the thickness accuracy comparing to the conventional PID controller.
Keywords: Direct MRAC, cascade-correlation neural network, dynamic neural network construction, cold mill thickness control

1. Introduction

A tandem cold-rolling mill (Fig. 1) is designed to reduce the thickness of the incoming strip, supplied in a coil at room temperature, by a factor of 2 to 10, so the outgoing strip has a uniform thickness with certain dimensions, typically 20-50 *inches* in width and 0.007-0.012 *inches* in thickness. The product is vital for the automotive industry and consumer goods manufacturers.

As the steel industry strives to improve product quality and reduce production costs, a viable scheme to achieve accurate thickness is of substantial interest to the industry. Considerable industrial research effort (Dutton and

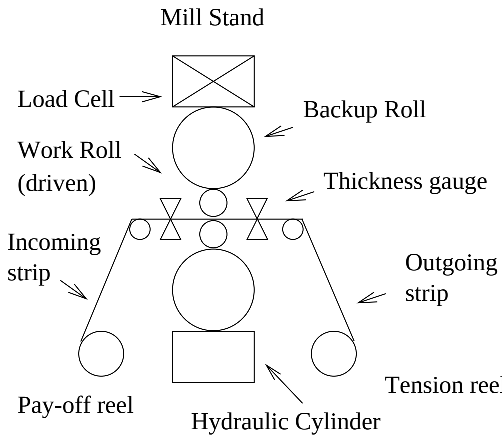

Figure 1. A single-stand reversing strip mill.

Groves, 1996; Grimble, 1995; Postlethwaite and Geddes, 1994) has been devoted to finding the best possible solution. Conventional thickness control systems are mainly based on the standard PID controllers, which are simple to implement but have limited accuracy. The rolling process, however, is a complex system with strong mutual interactions between the strip thickness, the roll gap position, the rolling force, and the strip hardness. Consequently, improvements in the design strategy are necessary to overcome the problem of mutual interactions.

Neural networks is a promising new technology which is becoming popular for control applications. However, most of the existing research in neural control has been concentrating on indirect control schemes, where the neural network is used to identify the process, and a controller is subsequently synthesized from this model (Brown and Harris, 1994; Harris *et al.*, 1993; Hunt *et al.*, 1992; Lin and Lin, 1996; Narendra and Pathasarathy, 1990; Narendra and Mukhopadhyay, 1994).

Such approach is very prominent in the applications of neural networks to steel manufacturing, where the main research efforts are concentrated on using neural network models to predict the process parameters such as the rolling force (for example, Hwu and Lenard, 1996; Pichler and Pffaffermayr, 1996).

The indirect control schemes mentioned above are following the traditional route better process models lead to better control. However, there is no warranty that even a very good model may lead to a good control. The inherent conflict between the identification and the control is well-recognized (see, for example, Koussoulass and Dimitriadis, 1989). The objective of the control is to minimize (make zero) the error between the actual process outputs and the desired

outputs (set-points).

However, when the error is zero, or a constant, one has no influence on the outputs and thus cannot identify the process parameters (Guez *et al.*, **1992). This refers to the lack of the persistency of excitation (Anderson, 1985). Tsakalis (1997) pointed out that the typical control law attempts to minimize the approximation error by driving the parameter estimations towards the manifold where the error becomes zero, thus allowing for the bursting phenomena. In the worst case, an appropriate arbitrarily small disturbance can be found to induce a parameter drift on this manifold and thereby cause a persistent bursting.**

Guez *et al.* **(1992) argue that a successful control requires the whole environment that acts systematically towards the goal of accumulating the knowledge and using it. In this paper, an attempt is made to design such environment that includes not only the learning in the controller but also the overall control scheme that permit such learning to occur.**

Consequently, the control should imply both the learning (identification) and the tracking (control in its traditional form). The authors argue that if one tries to optimize both objectives (the control and the identification), the identification solution alone would indeed play the role of the persistent excitation.

From the other angle, there are many types of neural networks **which are possible to use as a controller** (Haykin, 1999), among which the most popular one is the multilayer perceptron (MLP). However, with the MLP, there are no simple ways to determine in advance the **minimal structure of the network (number of hidden layers and the minimal size of each hidden layer)** necessary to achieve a desired performance . It is not uncommon to test many architectures to find the appropriate one by trial-and-error, although there are some algorithms for constructing an MLP during learning (Kavzoglu, 1999; Kwok and Yeung, 1997). In this paper, we use the cascade-correlation neural network (CCNN) (Fahlman and Lebiere, 1990) **for the simplicity of its implementation**.

This paper is organized as follows. Section 2 describes the control scheme and briefly reviews the CCNN. Section 3 describes the simulation model. Computer simulations using the CCNN and comparisons with the standard PID controller are shown in section 4. Section 5 summarizes the results in this paper.

2. The Control Scheme and the Cascade Correlation Neural Network

Note. Reversed the order of presentation: first the control scheme and then the CCNN.

A general multi-input multi-output (MIMO) nonlinear dynamical process can be represented by the following state-space representation:

$$\vec{x}(t+1) = \vec{f}[\vec{x}(t), \vec{u}(t), \vec{d}(t)] \ , \qquad (1)$$

$$\vec{y}(t) = \vec{g}[\vec{x}(t), \vec{\nu}(t)] \ . \qquad (2)$$

where $\vec{x} \equiv (x_1, x_2, .., x_n)$ is the process inputs, $\vec{u} \equiv (u_1, u_2, .., u_m)$ are the control signals (the manipulated variables), $\vec{d} \equiv (d_1, d_2, .., d_p)$ are disturbance inputs, $\vec{y} \equiv (y_1, y_2, .., y_m)$ are the process outputs, $\vec{\nu} \equiv (j_1, j_2, .., j_m)$ are the measurement noises, and t is the sampling time. Here n, m, and p are the dimensions of corresponding vectors.

In this paper we are using a direct model reference adaptive control (MRAC) (Åström and Wittenmark, 1990) where a neural network is the controller and no model of the process are required. The overall control scheme exploits the natural level of the excitation in the process under control by applying the same noisy inputs (states) signals to both the controller and the reference model.

The persistency of excitation (or the signals rich in frequency) is required for the exponential stability of an adaptive algorithm. Failure to satisfy this requirement may result in a bursting phenomena, also known as the parameters drift (Anderson, 1985). This means that in the absence of such excitation the parameters of the controller would grow extremely large thus resulting in a bursting. Consequently, the input and the output signals need to be rich in frequencies for the learning to be successful.

As pointed out by Tsakalis (1997), the parameters drift can be interpreted as a non-robustness of an ill-posed optimization problem. The error bursts in this framework are the immediate consequence of a Lipschitz continuity of the parameter approximation (finite adaptation gains). The estimation/approximation of the time-varying parameters in the absence of the sufficient excitation poses a challenging theoretical problem. In such a case achieving a *limsup* performance is as hard as achieving a L_∞ performance from the initial conditions that are zero in the output errors but arbitrary in the parameters. Consequently, the burst suppression in a general case requires the controllers with the infinite adaptation gains or the injection of the excitation.

However, it is impractical/dangerous to inject the artificial excitation signals into the closed-loop system, as such injection could not only result in the excitation of the high-order dynamic modes of the process, but could also result in the product/equipment losses. It would be more practical to exploit the noise/disturbances already existing in the process to provide such persistent excitation. While the

noise/disturbances are unmeasurable, their effects on the measurable variables are possible to measure.

A fundamental obstacle in overcoming the persistency of excitation problem is that the designer has limited or no control over the external inputs and, consequently, the level of excitation. This means, that a high level of excitation (frequency rich and large amplitude signals) are required in order to obtain the accurate parameters of the controller. However, a low-level excitation required by a typical control objective, for example, the regulation, the disturbance rejection, and the tracking of the low-frequency reference signals (Tsakalis, 1996).

One may look on the inference canceling in the adaptive signal processing for the analogy. To filter the noise from the signal an optimal Kalman or Wiener filter are not well suited as they introduce some inevitable phase distortion. A better solution is to introduce an additional reference input x_nr containing the noise which is correlated with the original corrupting noise x_n. The network filters the reference noise x_nr to produce an estimate of the actual noise $x*_n$. Then the network subtracts the noise from the primary input $s + x_n$ which acts as the desired response to produce the estimate of the signal $s*$ (Zaknich and Attikiouzel, 1995).

Now in the context of the adaptive control, one may try to exploit a natural level of excitation that exists in the process under control and apply the same noisy inputs (states) signals to both the controller and the reference model. If a reference model were in the form of a filter (for example, a Butterworth filter), the output of a reference model would be a desired signal with the acceptable level of the noise in it. Using the error between the actual response of the process and the desired one from a reference model, it should be possible to find a needed auxiliary function (controller) f_c. This auxiliary function would transfer the original process function f_p into a desired one between the noisy inputs (states) and the desired outputs of the reference model. In such a case a reference model represents our desired process (the controller plus the original process) f_d.

In view of the above discussion, the squared difference between the desired output set-point $\vec{y}^{sp}$ and the process output $\vec{y}$ cannot be used as the objective function to be minimized for the learning in a neural control:

$$\varepsilon_l(t) = \frac{1}{2}(y_l^{sp}(t) - y_l(t))^2 \quad . \tag{3}$$

Here $l = 1, 2, ..., m$. With such objective function the neural controller learning may proceed to a physically unrealizable situation, since the set-points obviously are not persistently exciting (Anderson, 1985). In order to overcome the lack of the persistency of excitation, one may

obtain the desired output response $\vec{y}_d$ from the output of a reference model with the state variables $\vec{x}$ being inputs to the model. In this case the excitation is due to the actual process signals (states) affected by the disturbance signals $\vec{d}$ and the noise $\vec{n}$. See also Tsakalis (1996) for discussion on the injection of the persistently excited signals in the closed-loop systems.

The objective function in such a case becomes the squared difference between the outputs of the process $\vec{y}$ and the reference model $\vec{y}_d$.

$$\varepsilon_l = \frac{1}{2}(y_l - y_{dl})^2 \quad . \tag{4}$$

In addition, a reference model in the form of a filter with a desired transfer function may be used to ensure a variance in the desired dynamic characteristics of the process. The frequency response of the closed-loop may be adapted in line with the changes in the frequency responses of the filter. Such reference models may be used when the required performance of the time-varying process is to be achieved by the change in the overall process transfer function. This is an optimization-based design method using a Modulus Optimum, also called a loop-shaping method (Hågglund and Åström, 1996).

One may use a linear stable reference model, for example, a Butterworth filter (Åström and Wittenmark, 1990), since the general well-behaved nonlinear models are not yet available. The coefficients of a Butterworth filter are thus selected to correspond to the Modulus Optimum criteria for a desired performance in terms of the standard control objectives such as the overshoot, the settling time, and the steady-state error. While the use of such a general reference model would not permit us to achieve an ideal control, this should guarantee the adequate control performance for a wide range of the processes in which it is to be used. The controller (CCNN in our case) is designed to transfer the original process transfer function to a desired one.

The learning algorithm is designed to obtain the correct control signals (manipulated variables) u_l $(l = 1, 2, ..., m)$ corresponding to the desired process outputs y_{dl} by minimizing the learning error ε_l, defined as the difference between the desired process responses y_{dl} and the measured process outputs y_l

$$\varepsilon_l = \frac{1}{2}(y_l - y_{dl})^2 \quad . \tag{5}$$

A block diagram of the overall control system is presented in Fig. 2.

Fig. 3 shows the architecture of the CCNN (Fahlman and Lebiere, 1990), **used as a controller in the the above control scheme,** whose construction and learning algorithms can be summarized as follows:

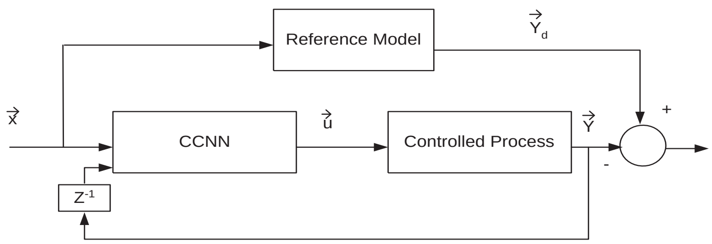

Figure 2. A block diagram of the overall control system.

1. Starts with a minimal network consisting only an input layer and an output layer. Both layers are fully connected with adjustable weights. There is also a bias unit, set permanently to +1. Linear output units are used.
2. Train all connections to the output layer using the quickpropagation learning algorithm (Fahlman, 1988) until the overall error of the network no longer decreases.
3. If the network performance satisfies a prescribed accuracy target, the algorithm stops. In such case, as there is no hidden layer, the problem at hand is linear.
   We note that CC can thus be used to test if the problem at hand is *really* nonlinear. There is no benefit in applying neural controller to a linear or linearizable plant, as this will result in degradation of performance in terms of computation time and controller performance: the solution should not be more complex than the problem at hand (Mars *et al*, 1996).
4. If the network performance is not satisfactory (and therefore the problem is really nonlinear), generate candidate nodes. Every candidate node receives trainable connections from all inputs nodes and from all pre-existing hidden nodes. There are no connections between the candidate nodes and the output nodes.
5. Maximize the correlation between the activation of the candidate nodes and the residual error of the network by training all connections leading to a candidate node. The training stops when the correlation no longer improves.
6. Choose the candidate node with the maximum correlation and add it to the network. To change the candidate node into a hidden node, connect it to all output units. Return to step 2.

The algorithm is repeated until the overall error of the network falls below a pre-specified threshold.

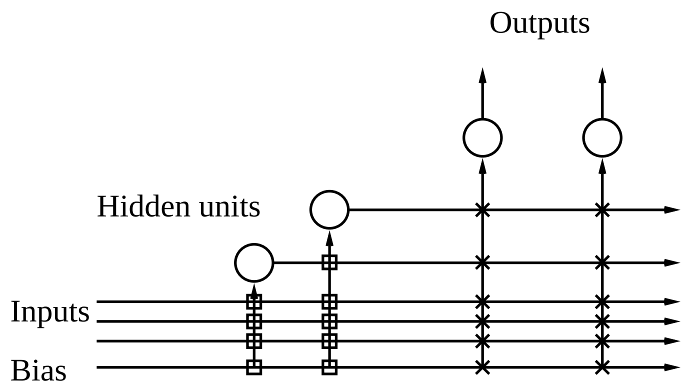

Figure 3. The structure of the cascade-correlation neural network (CCNN).

3. Simulation Model of a Cold Rolling Mill

As shown in Fig. 1, in a single-stand reversing strip mill, the incoming strip is supplied from a pay-off reel at one side of the mill, reduced in thickness as it is passed between the mill work rolls, and recoiled by a tension reel at the other side of the mill. The roll gap is then reduced, and the process is repeated in the reverse direction. The process continues until the outgoing strip is of the desired final thickness. The thickness of the rolled strip is predominantly determined by the gap between the work rolls, although there are other contributing parameters, such as the tension in the strip, hardness variations in incoming materials, and hardening of the material during rolling. The roll gap is initially set by electrical screw-down drives. Once the strip is threaded, changes to the roll gap are carried out by extending/contracting hydraulic cylinders. Automatic gauge control (AGC) for cylinder control is used in two modes, pressure (load) control or position control. Fig. 4 (Dutton and Groves, 1996) shows a typical arrangement of a control loop for cylinder position. For pressure control, position transducers are replaced by pressure transducers.

In practice, the cylinder position reference signal will have contributions from several other control loops not shown in **Fig. 4**. For example, the outgoing strip thickness will be measured and any deviation from the desired value requires reduction of the reference signal for cylinder position. However, for physical reasons, it is impossible to measure the outgoing thickness until some distance downstream of the roll gap. This introduces a transport lag, which severely degrades the gauge performance when compensating for short-duration errors. In high performance mills, the incoming thickness at the roll gap is also measured and used in a feedforward control loop, so that the cylinders are adjusted in line with variations in the roll gap. Another common disturbance to the operation of the position control loop is the roll eccentricity, arising from imperfect roll

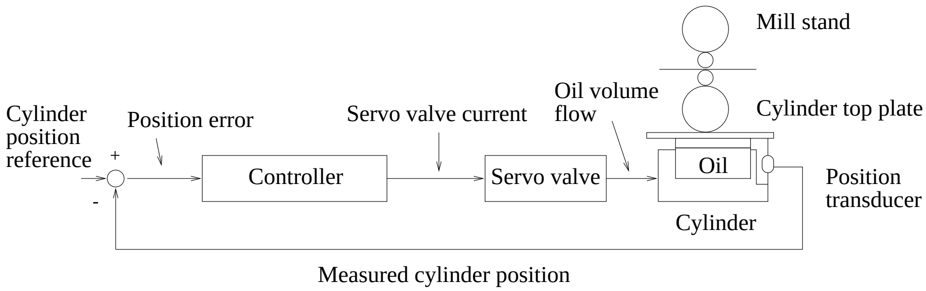

Figure 4. Automatic gauge control (AGC) gap position loop.

grinding and roll wear.

In our simulation studies presented in the next section, we have used an AGC position control servo valve and capsule model (Fig. 5) developed from the physical insight of the process (Dutton and Groves, 1996) to replace the servo valve and capsule in Fig. 4 (the position transducer is assumed to have negligible dynamics).

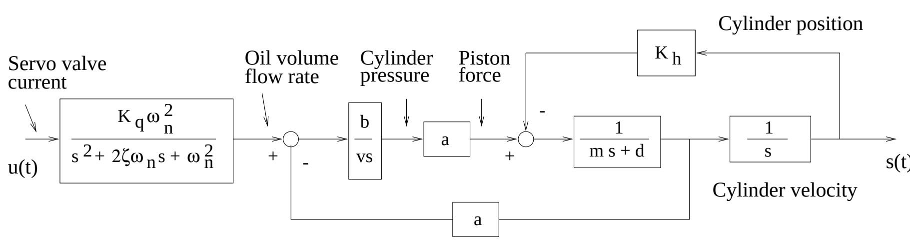

Figure 5. Servo valve and cylinder model of a position loop. Here $s = dx/dt$, $\omega_n = 120\ rad/s$ is the servo valve natural frequency, $\zeta = 1.1$ is the valve damping ratio, $K_q = 25\ m^3/(sA)$ is the flow gain, $b = 1.4 \times 10^9\ N/m^2$ is the oil compressibility, $a = 0.58\ m^2$ is the capsule cross-sectional area, $v = 0.0232\ m^3$ is the capsule volume, $m = 1 \times 10^5\ kg$ is the mass of the mill, $K_h = 4.5 \times 10^9\ N/m$ is the mill housing stiffness, and $d = 5 \times 10^6$ is a damping term.

Although the main emphasis is on the gap position control, the other parts of the mill need to be taken into account (such as the force control loop), as their disturbances and noise affect the performance of the position control. This results in a complete model representation of the combined mill position control, disturbances and noise (Fig. 6) (Grimble, 1995).

The parameter values were selected based on the experimental and the physical property data (Grimble, 1995). The frequencies and the gain values of the model were scaled appropriately to match the experimental data.

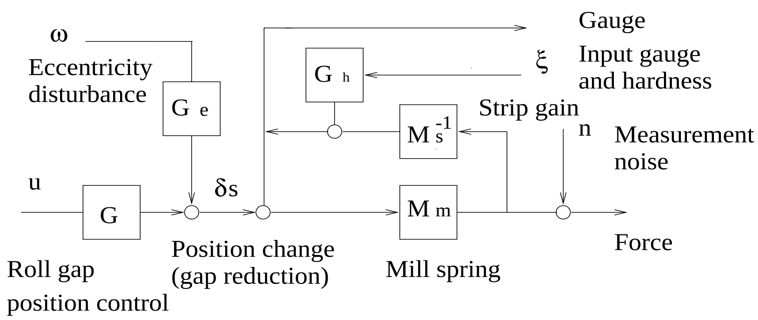

Figure 6. An overall single-stand model for the cold rolling mill.

In Fig. 6, G represents a hydraulic roll-gap model for closed-loop position control of Fig. 5. G_e is an eccentricity disturbance model:

$$G_e = 852.224/((s^2 + 0.0314s + 985.96) \times (s^2 + 0.0294s + 864.36)) \ , \quad (6)$$

and G_h is an input thickness and hardness disturbance model:

$$G_h = 0.333/(s + 0.333) \ . \quad (7)$$

Here $s = \partial x / \partial t$.

Grimble (1995) experimentally verified the use of an eccentricity model comprising two lightly damped oscillators driven by zero-mean white noise with covariance $E\{\omega(t)\omega(\tau)\} = x_1^2$, where $x_1 = 0.00012$.

Since it is difficult to obtain data for hardness variation in cold rolled strip mills, we used a first order lag driven by zero-mean white noise (Grimble, 1995) with covariance $E\{\xi(t)\xi(\tau)\} = x_2^2\delta(t-\tau)$, where $x_1 = 0.00007$ and δ is the Kronecker delta function. The disturbances and noise were applied concurrently to represent a real situation when the disturbances and noise in the rolling mill are present at the same time.

A combination of these models results in a complete model representation of the combined mill and the disturbance system.

A small change model is used (Grimble, 1995) to generate roll force and gauge variations, with gauge $h(t)$ satisfies:

$$h(t) = \frac{M_m M_s^{-1}}{1 + M_m M_s^{-1}} \delta s(t) + \frac{1}{1 + M_m M_s^{-1}} \delta H(t) \ , \quad (8)$$

where $M_m = 1.039 \times 10^9 \ N/m$ and $M_s = 9.81 \times 10^8 \ N/m$ are the mill and the strip moduli, respectively, and $s(t)$ is the roll gap setting. Thus the measured roll force $z(t)$ is:

$$\delta z(t) = M_m(\delta h(t) - \delta s(t)) + n(t) \ , \quad (9)$$

where $n(t)$ represents measurement noise with covariance $E\{n(t)n(\tau)\} = x_3^2\delta(t-\tau)$, where $x_3 = 1000$.

Thickness control in a cold rolling mill requires the output gauge to be regulated in the presence of disturbances using the measured roll gap position and the roll force. Here we have an inferential control problem where it is not the measured variables that are controlled, but the measured variables are used to achieve control of another system variable - the strip gauge (thickness).

4. Simulation Results

We used Butterworth's (Åström and Wittenmark, 1990) characteristic equation for the 5th order system as a reference model

$$F(s) = s^5 + 3.24\omega_n s^4 + 5.24\omega_n^2 s^3 + 5.24\omega_n^3 s^2 + 3.24\omega_n^4 s + \omega_n^5 \ , \qquad (10)$$

where ω_n is a natural frequency of the system $\omega_n = 200\ rad/s$. This form of characteristic equation gives us a damping ratio $\xi = 0.71$, and the settling time can be determined through approximate relationship $t_s \approx 4/\xi\omega_n$. The delay time t_d can also be approximated from the following relationship $t_d \approx (1 + 0.7\xi)/\omega_n$. Consequently, the number of delays in eq. 5 $c = 2$. A 4 Hz square wave with the amplitude of 10 μm **same** as in Dutton and Groves,(1996) applied to cylinder position reference was used as a test signal.

Removed the Fig. 7 - square wave signal

We used the input-output data generated using above reference model for training and testing of the CCNN. We have generated 1000 input-output data pairs using fifth order Runge-Kutta integrator (Åström and Wittenmark, 1990) with a sampling time $t = 0.005s$ normalized in the range [-1, 1]. **The states of the process (inputs to the CCNN controller) are the strip gauge, the rolling force, the eccentricity disturbance, and the roll gap position.** We used first 500 samples for training, the other 500 for validation of controller's ability to generalize, and the whole data set for final testing of the resulted controller.

The selection this particular evaluation approach was based on the following reasoning:

1. **Both the CCNN and the PID controllers were trained/tuned based on the first 500 samples of the input-output data.**
2. **For both controllers used the next 500 samples of the input-output data were used to test the controllers ability to perform under the changing process conditions.**
3. **The final run of both controller over the whole set of input-output data was used to test that the controllers are able to perform well both for the known data and the unknown data. While the performance of the controller is traditionally demonstrated on the testing (unseen) data only this prevents the comparison of**

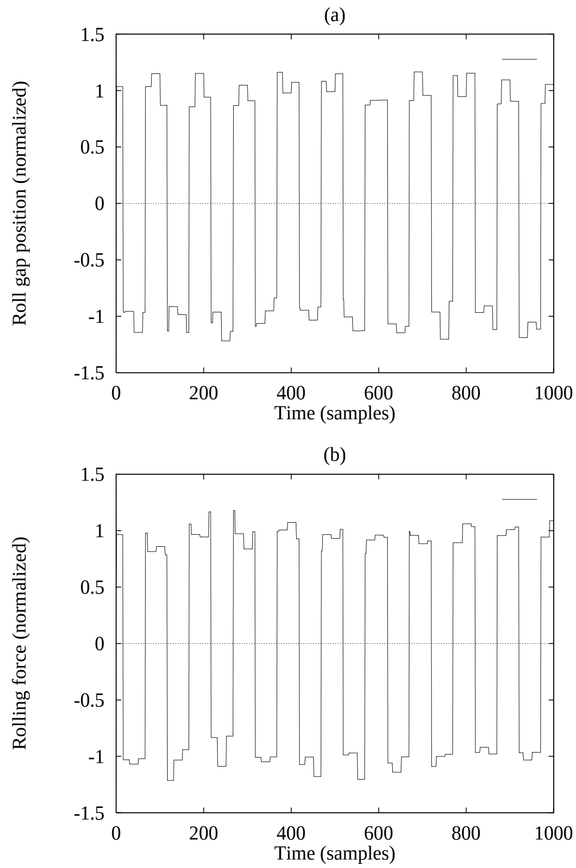

Figure 7. The time response for uncontrolled (a) roll gap position and (b) rolling force.

> **any degradation of the performance when the controller is used on unseen data in comparison to its performance on the known (training) data. The controller ability to generalize is at it best where there is a minimal difference between its performance on known data and its performance on unseen data. While this well known, it is, for some reasons, rarely evaluated.**

We selected **as a CCNN parameters** the learning rate $\eta = 0.005$, maximum growth parameter $\mu = 1.75$, weight decay term $\nu = 0.0001$, maximum tolerated difference between a teaching value and an output $d_{max} = 0.1$). **The parameters of the CCNN were selected based on preliminary tests, using as a measure the minimum RMSE between the cylinder position reference signal and the ones obtained using CCNN controller**

for both the position and the rolling force. For comparison, two PID controllers was used for the same **simulated** process (one for the position loop, another for the force loop). The parameters of PID controllers, i.e., gain $K_c = 15.5$, integral time constant $\tau_I = 1.6$, and derivative time constant $\tau_D = 2$, were selected according to the industry standard Ziegler-Nichols' method (Åström and Wittenmark, 1990). **The design bandwidths for both the CCNN and the PID controllers were the same as they only depend on a natural frequency of the process and a damping ratio. As there is a well known difficulty in optimization of the PID controller to control of such complex process as a rolling mill, an additional fine tuning of the controller gains were utilized to obtain the best possible performance of the PID controllers.**

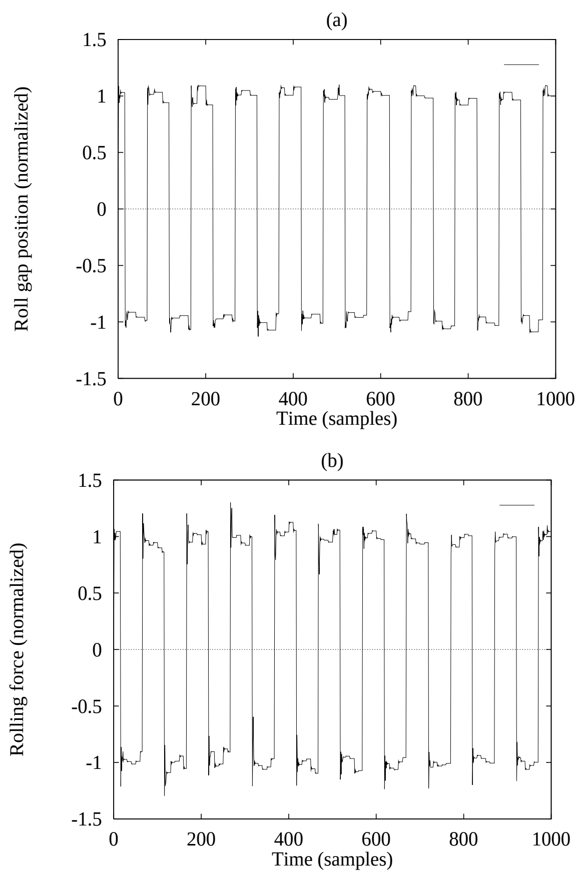

Figure 8. The performance of the PID controllers for (a) the roll gap position and (b) the rolling force.

We present the simulation results in Figs. 7-9 and Table 1. Fig. 7 shows the uncontrolled time response (root mean squared error RMSE = 0.1724 for the position and 0.1232 for the rolling force). While reducing the deviations due to disturbances and noise of the position of roll gap and the rolling force to some extend, the PID controller showed rather poor performance (RMSE = 0.0897 for the position and 0.0787 for the rolling force in Fig. 8). The reasons for the poor performance of the conventional "optimal" PID controller may be attributed to non-linearity and/or the coupling in the rolling process. Fixed PID controllers are thus not able to capture the underlying process behavior well.

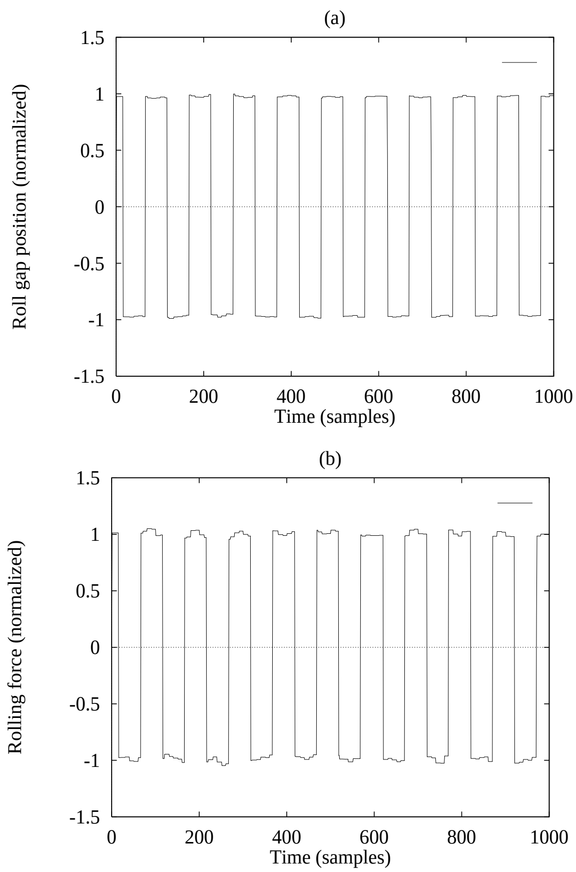

Figure 9. The performance of the cascade-correlation neural network (CCNN) controller for (a) the roll gap position and (b) the rolling force.

In contrast, the performance of the CCNN is much better (RMSE = 0.0312 for the position and 0.0518 for the rolling force in Fig. 9). The CCNN generated

37 hidden nodes, which signifies that the process is indeed a non-linear one. **Moreover, the CCNN produced practically identical results in regards to both RMSE and the network size for different runs. The Figs. 7-9 show a typical result of the application of the CCNN controller as it is our belief that the averaging the results of a controller's application is inappropriate**.

	Uncontrolled	PID	CCNN
Roll gap position	0.1724	0.0897	0.0312
Rolling force	0.1232	0.0787	0.0518

Table 1. A comparison of the root mean squared errors (RMSE) of an uncontrolled system, a system controlled by a PID controller, and a system controlled by a cascade-correlation neural network (CCNN) controller.

5. Conclusion

In this work we used a direct model-reference adaptive control scheme with the cascade-correlation neural network (CCNN) as a controller for cold rolling mill thickness control which exploits the natural level of the excitation existing in the closed-loop. We have demonstrated that such direct MRAC scheme with a CCNN as a controller significantly increased the control precision and robustness compared to the linear PID controller in this important real-world problem. We argue that a direct MRAC scheme designed in this paper with a CCNN controller using both structure and parameter learning can provide a computationally efficient solution to control of many real-world nonlinear processes in the presence of process disturbances and measurement noise.

References

Anderson, B. D. O. (1985)
Adaptive systems, lack of persistency of excitation and bursting phenomena.
Automatica, **21(3)**, 247–258.

Åström, K.and Wittenmark, B. (1990)
Computer controlled systems, theory and design.
2nd ed. Prentice-Hall.

Brown, M.and Harris, C. (1994)
Neurofuzzy Adaptive Modelling and Control.
Prentice-Hall.

Dutton, K. and Groves, C. N. (1996)
Self-tuning control of a cold mill automatic gauge control system.
Int. J. Control, **65**, 573–588.

Fahlman, S. (1988)
Faster-learning variations on back-propagation: An empirical study.
1988 Connectionist Models Summer School, 38–51.

Fahlman, S. and Lebiere, C. (1990)
The cascade-correlation learning architecture.
Advances in Neural Information Processing Systems, **2**, 524–532.

Grimble, M. J. (1995)
Polynomial Solution of the Standard H_∞ Control Problem for Strip Mill Gauge Control.
IEE Proc.-Control Theory Appl., **142**, 515–525.

Guez, A., Rusnak, I. and Bar-Kana, I. (1992)
Multiple objective optimization approach to adaptive and learning control.
International Journal of Control, **56(2)**, 469–482.

Hågglund, T. and Åström, K. (1996)
Automatic tuning of PID controllers.
The Control Handbook, CRC Press, 817–826.

Harris, C., Moore, C.and Brown, M. (1993)
Intelligent Control: Aspects of Fuzzy Logic and Neural Nets.
World Scientific.

Haykin, S. (1999)
Neural Networks: A Comprehensive Foundation.
Prentice Hall, 2nd ed.

Hunt, K., Sbarbaro, D., Zbikowski, R. and Gawthrop, P. (1992)
Neural networks for control systems - a survey.
Automatica, **28**, 1083–1112.

Hwu, Y.J. and Lenard, J. G. (1996)
Application of Neural Networks in the Prediction of Roll Force in Hot Rolling.
Proceedings of the 37th Mechanical Working and Steel Processing Conference, 549-554.

Kavzoglu, T. (1999)
Determining Optimum Structure For Artificial Neural Networks.
Proceedings of the 24th Annual Technical Conference and Exhibition of the Remote Sensing Society (Earth Observation: From Data to Information), 675–682.

Koussoulass, N. T. and Dimitriadis, E. K. (1989)
Computational techniques for multi-criteria stochastic optimization and control.
Control and dynamic systems, advances in theory and applications, **30(3)**, 1–17.

Kwok, T. Y. and Yeung, D. Y. (1997)

Constructive algorithms for structure learning in feedforward neural networks for regression problems.
Transactions on Neural Networks, **8(3)**, 630–645.

LIN, C.-J. and LIN, C.T. (1996)
Reinforcement learning for an ART-based adaptive learning control network.
Trans. Neural Networks, **7**, 709–731.

MARS, P., CHEN J. R. and NAMBIAR R. (1996)
Learning algorithms: theory and applications in signal processing, control and communications.
CRC Press, Boca Raton, FL.

NARENDRA, K. and PATHASARATHY, K. (1990)
Identification and control of dynamical systems using neural networks.
IEEE Trans. Neural Networks, **1**, 4–27.

NARENDRA, K. and MUKHOPADHYAY, S. (1994)
Adaptive control of nonlinear multivariable systems using neural networks.
Neural Networks, **7**, 737–752.

PICHLER, R. and PFFAFFERMAYR, M. (1996)
Neural Networks for On-Line Optimisation of the Rolling Process.
Iron and Steel Review, **August**, 45–56.

POSTLETHWAITE, I. and GEDDES, J. (1994)
Gauge Control in Tandem Cold Rolling Mills: A Multivariable Case Study Using H^∞ Optimization.
Proceedings 3rd IEEE Conference on Control Applications, **3**, 1551–1556.

TSAKALIS, K. S. (1996)
Performance limitations of adaptive parameter estimation and system identification algorithms in the absence of excitation.
Automatica, **32(4)**, 549-560.

TSAKALIS, K. S. (1997)
Bursting scenario and performance limitations of adaptive algorithms in the absence of excitation.
Kybernetika, **33(1)**, 17–40.

ZAKNICH, A. and ATTIKIOUZEL, Y. (1995)
Application of artificial neural networks to nonlinear signal processing.
Computational intelligence, a dynamic system perspective, IEEE Press, New York, 292–311.