Modelling the Strip Thickness in Hot Steel Rolling Mills Using Least-Squares Support Vector Machines

Yuri A. W. Shardt,[1] Siamak Mehrkanoon,[2] Kai Zhang,[3] Xu Yang,[3]* Johan Suykens,[2] Steven X. Ding[4] and Kaixiang Peng[3]

1. *Department of Chemical Engineering, Faculty of Engineering, University of Waterloo, Waterloo, ON, Canada and Institute of Automatic Control and Complex Systems (AKS), University of Duisburg-Essen, Germany*
2. *ESAT-STADIUS, Catholic University of Leuven, Leuven, Belgium*
3. *Key Laboratory of Knowledge Automation for Industrial Processes of the Ministry of Education, School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing, China*
4. *Institute of Automatic Control and Complex Systems (AKS), University of Duisburg-Essen, Germany*

The development and implementation of better control strategies to improve the overall performance of a plant is often hampered by the lack of available measurements of key quality variables. One way to resolve this problem is to develop a soft sensor that is capable of providing process information as often as necessary for control. One potential area for implementation is in a hot steel rolling mill, where the final strip thickness is the most important variable to consider. Difficulties with this approach include the fact that the data may not be available when needed or that different conditions (operating points) will produce different process conditions. In this paper, a soft sensor is developed for the hot steel rolling mill process using least-squares support vector machines and a properly designed bias update term. It is shown that the system can handle multiple different operating conditions (different strip thickness setpoints, and input conditions).

Keywords: soft sensors, steel mill, support vector machines, process systems engineering

INTRODUCTION

With the increasing demands placed on industry in terms of tight production schedules, decreased profit margins, and increased emphasis on safety, there has been an increase in the need to implement and develop methods that can improve the overall control strategy.[1] One industrial area where these concerns are increasingly being felt is the hot steel rolling mill process. In this process, the fast nature of the system implies that even small delays in detecting faults can lead to large amounts of wasted steel. This implies that any methods that can improve the detection of faults will have a large industrial benefit for this process.

In the hot steel mill rolling process, the key performance indicator is the final thickness of the steel, which is measured at the exit using an X-ray sensor. Unfortunately, this sensor is located at some distance from the overall process, which implies that there is a significant time delay before the value can be obtained. One approach to this problem has been to develop intricate models of the process in order to predict the required setpoints for the process.[2] However, in many cases, obtaining all the relevant information has proven difficult. Thus, data-driven methods, which do not require knowledge of process parameters, have been developed.[3] However, the initial conditions of the steel and the desired steel thickness can vary from batch to batch, implying that different models for each of the possible combinations would be required. Furthermore, it is often impossible to know all the initial conditions for the given steel roll, so that it is not possible to properly assign the given steel roll to the given model.

Recently, soft sensors have been considered for application to the hot steel rolling mill process.[4,5] Soft sensors are a mathematical framework for forecasting process values where there are issues with missing, inaccurate, or random process variables, which can result from lack of sensors, design of sensor networks, or other similar reasons.[6] A typical soft sensor system consists of two components: a process model and a bias update term.[6] In general, it is assumed that the process model exists to provide accurate and detailed updates about the process using all relevant inputs about the process and is able to accurately track changes in the overall process. However, the process model may not be able to get the correct absolute value, that is, the process model may be biased compared with the true value. It is the role of the bias update term to correct any bias by comparing the true values with the soft sensor values. It should be noted that the reason for this separation of terms lies in the fact that measuring the true process value is often performed at a much slower rate than that of the main system. This implies that the bias update term can only be updated when new data is available. The process model can be obtained using any of the standard methods for modelling of chemical processes including simple regression analysis,[7] principal component analysis (PCA),[8] partial least squares (PLS),[9] neural networks,[10] Kalman filter,[11] and support vector machines (SVM).[12,13]

In soft sensor design, SVM, which seek to remove the nonlinearities present in the data set by projecting the data into a higher-dimensional, feature space,[14] can be used to reduce the complexity of the resulting model. One approach to this problem is

* *Author to whom correspondence may be addressed.*
E-mail address: yangxu@ustb.edu.cn
Can. J. Chem. Eng. 96:171–178, 2018

DOI 10.1002/cjce.22956
Published online 24 August 2017 in Wiley Online Library (wileyonlinelibrary.com).

to use a least-squares support vector machine (LS-SVM)-based approach that allows for a simple solution of the problem without excessive computations or overfitting problems.[13] This approach has been used in such fields as function estimation,[13] determining solutions of differential equations,[15,16] and parameter estimation of dynamic systems.[17] However, its application to soft sensor design has not been tested.

Therefore, this paper seeks to analyze the problem of developing a soft sensor for the hot steel rolling process using LS-SVM that can deal with multibatch cases arising from different operating conditions. The soft sensors will be developed and tested using industrial data.

THEORY

Soft Sensor System

Assume that the soft sensor system can be represented as shown in Figure 1, which consists of two parts: the process model, $\widehat{G}_p$, and the bias update term, G_B. It can be noted that y_t is the true process output, u_t the process input (manipulated variables), d_t the disturbance, $y_{m,\,t}$ the forecast output value from the soft sensor system, G_p the true process function, and G_l the disturbance model.

In terms of soft sensor design, most of the attention has focused on developing and analyzing appropriate methods for developing accurate and reliable process models. The overall design philosophy can be written as follows:

1) **Gather Process Knowledge**, which consists of data, process schematics, and process information from the various operators.
2) **Preprocess the Information**, which performs a preliminary analysis of the data set to determine if there are any obvious issues, as well as providing the user with a good picture of the overall data properties.
3) **Develop the Process Model**, which uses all the available information, appropriate data modelling techniques, and analysis to obtain a good model for the process.
4) **Testing the Model**, which involves determining and showing that the given model reflects the given process well and can accurately forecast future values of the system.

It can be noted that this process is circular, in that it is common to go back a step or two whenever there are problems. This makes the modelling exercise quite involved and the final result can take time to obtain. However, following this procedure ensures that the final model is accurate and can perform satisfactorily in an industrial setting.

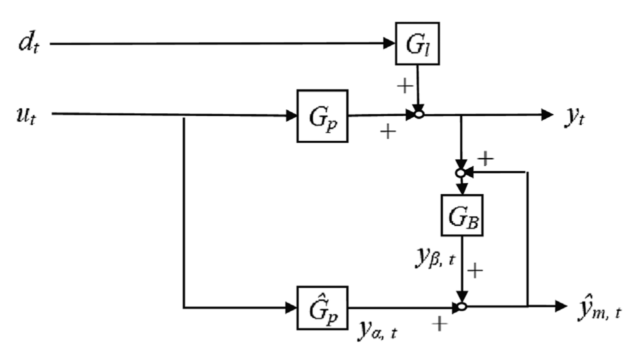

Figure 1. Schematic of an Open-Loop Soft Sensor with Bias Update (after Shardt and Huang[18]).

Least-Squares Support Vector Machines

In many modelling problems, the initial data set is highly nonlinear. Fitting a model to such data can be difficult, so methods have been developed that allow the nonlinear data to be projected onto a suitable, higher dimensional space. Modelling is then performed in this higher dimensional space. In order to better understand the application of support vector machines to the modelling problem, the application of support vector machines to the 2-class problem will be first considered.

One such method is support vector machines, which seek to determine the largest margin between two data sets. In the linear case, this margin will cleanly separate the two data sets. However, in the nonlinear case, this may no longer be feasible.

Consider a training data set, $\{x_i, y_i\}_{i=1}^{N}$ where x_i is the input data point y_i is the output data point, and N is the number of data points. The standard, linear support vector machine optimization problem can be stated as follows:[14]

$$\begin{aligned} &\min_{w,b,\xi}\ \frac{1}{2}\,w^T w + c\sum_{i=1}^{N}\xi_i \\ &\text{subject to} \\ &y_i(w^T x_i + b) \geq 1 - \xi_i \quad i = 1, \ldots, N \\ &\xi_i \geq 0\ i = 1, \ldots, N \end{aligned} \tag{1}$$

where w is a vector of weights, c a tuning parameter, ξ the slack variables, and b the intercept of the hyperplane. It should be noted that in the standard support vector machine problem it is assumed that the output data point is simply 1 or -1, which denotes which of the two sets the given input data belongs.

By allowing the output values to vary, the standard support vector regression problem is obtained. It assumes that the model for the process can be written as follows:

$$y_i = w^T x_i + b \tag{2}$$

This gives an optimization problem that can be written as follows:

$$\begin{aligned} &\min_{w,b}\ \frac{1}{2}\,||w||^2 \\ &\text{subject to} \\ &y_i - w^T x_i - b \leq \epsilon \\ &w^T x_i + b - y_i \leq \epsilon \end{aligned} \tag{3}$$

where ε is the threshold. This threshold plays a similar role to the slack variables in the original problem. However, the problem with both of these approaches is that they require that a quadratic programming problem with inequality constraints be solved. In practice, this can be a difficult proposition. Thus, the least-squares support vector machine (LS-SVM) framework was proposed by Suykens et al.[13]

In the least-squares support vector machine (LS-SVM) framework, one works with a L_2-loss function and equality constraints, instead of the inequality constraints present in SVM, which allows for solving a linear system rather than a quadratic programming problem. In the LS-SVM framework, it is assumed that the underlying function describing the relationship between the input and output of the system has the following form:

$$y_t(u_t) = w^T\phi(u_t) + b \tag{4}$$

where ϕ is the feature map that projects the inputs into the feature space. The data are embedded into the feature space using a nonlinear feature map. The optimal solution is sought in the new space by minimizing the residual between the model outputs and the measurements. The primal LS-SVM formulation has the following form:[13]

$$\begin{aligned} &\min_{w,b,e} \frac{1}{2} w^T w + \frac{\gamma}{2} e^T e \\ &\text{subject to} \\ &y_i = w^T\phi(u_i) + b + e_i,\ \forall i = 1, \ldots, N \end{aligned} \tag{5}$$

where γ is strictly positive, b is a constant, w are the weights, and:

$$e = \langle e_1, e_2, \ldots, e_N \rangle \tag{6}$$

The complexity of the model is controlled by γ and thus overfitting can be avoided.[13] In the LS-SVM approach, the feature map ϕ is not explicitly known in general and can be infinite dimensional. Therefore, the kernel trick is used and the problem is solved in the dual form.[13] The Lagrangian of the constrained dual optimization problem becomes the following:

$$\mathfrak{L}(w, b, e, \alpha_i) = \frac{1}{2} w^T w + \frac{\gamma}{2} e^T e - \sum_{i=1}^{N} \alpha_i [w^T\phi(u_i) + b + e_i - y_i] \tag{7}$$

where α are the Lagrange multipliers. In this case, the Karush-Kuhn-Tucker (KKT) optimality conditions are:

$$\begin{aligned} &\frac{\partial \mathfrak{L}}{\partial w_j} = 0 \rightarrow w_j = \sum_{i=1}^{N} \alpha_i \phi(u_i) \\ &\frac{\partial \mathfrak{L}}{\partial e_i} = 0 \rightarrow \alpha_i = \gamma e_i \\ &\frac{\partial \mathfrak{L}}{\partial b} = 0 \rightarrow \sum_{i=1}^{N} \alpha_i = 0 \\ &\frac{\partial \mathfrak{L}}{\partial \alpha_i} = 0 \rightarrow w^T\phi(u_i) + b + e_i = y_i \end{aligned} \tag{8}$$

Eliminating the primal variables e_i and w leads to the following linear system in the dual problem:

$$\left[\begin{array}{c|c} \Omega + I_N/\gamma & 1_N \\ \hline 1_N^T & 0 \end{array}\right] \left[\begin{array}{c} \alpha \\ \hline b \end{array}\right] = \left[\begin{array}{c} y \\ \hline 0 \end{array}\right] \tag{9}$$

where $\Omega_{ij} = K(u_i, u_j) = \phi(u_i)^T\phi(u_j)$ is the ij-th entry of the positive definite kernel matrix, 1_N is a $N \times 1$, column vector of 1's, α is the vector containing all the individual α_i, y is the vector containing all the measurements, and I_N is the $N \times N$ identity matrix.

The model in the dual form becomes:

$$y_t(u_t) = w^T\phi(u_t) + b = \sum_{i=1}^{N} \alpha_i K(u_t, u_i) + b. \tag{10}$$

Dealing with Interbatch Variation and Time Delay

In many complex, batch, or semi-batch processes, unexpected process changes can occur that will perturb the overall process. These changes will then need to be accounted for in the development of the model. For example, in the hot steel rolling mill process, the final thickness of individual batches depends not only on the process itself, but also on the variable initial conditions of the roll, such as the temperature, thickness, and cooling rate. The impact of such changes on the process can be viewed as a change in the initial offset or bias in the problem with the overall changes being similar.

In practical terms, during online implementation of the soft sensor, the direction of change may be accurately forecast in such cases, but the actual, absolute value would be shifted from that expected, leading to a bias in the terms.

In order to eliminate this bias, there is a need to include the available process measurements in order to provide an estimate of the bias. The easiest approach to take here is to simply consider the difference between the forecast value and the measured value and update the future process values based on this approach. Although this approach will work in an open-loop situation, it will not provide tracking in closed-loop.[19] Thus, there is a need to design the bias update term appropriately in order to consider the overall impact on the system.

Furthermore, in many practical systems, the process values are not immediately available due to such factors as time delay or sampling rate. This lack of information can introduce additional limitations on both the design parameters and the overall bias update system. As well, since different batches can potentially have different delay characteristics, it can happen that until process information is available, the soft sensor model may not be very accurate. This situation can easily arise, in such examples as steel rolling mills, where the speed at which the steel passes through the mill depends on such factors as initial thickness, desired thickness, and potentially other factors.

It has been shown that the behaviour of the bias update term strongly depends on the manner in which the soft sensor will be used. In open-loop cases, then a simple bias update term is sufficient to provide good process tracking, while in closed-loop cases, the bias update term will need to contain an integrating term in order to track all possible situations.

Since the bias update term provides an easy method for updating and incorporating measured information into the soft sensor, it will be used in the proposed soft sensor system to handle any uncertainties arising from changes in the initial conditions, as well as to account for unmeasured disturbances and the inevitable plant-model mismatch.

PROCESS DESCRIPTION

The hot rolling process (HRP) consists of 6 key parts: the reheating furnace, the rough mill, the transfer table and crop shear, the finishing mill, the run-out table cooling, and the coiler.[20,21] A single batch consists of a coil of rough steel, which enters the reheating furnace to be reheated to the appropriate temperature. Next, the strip passes through the rough mill, where its thickness and width are reduced to close to the desired value. Then, the strip passes through the transfer table and enters the finishing mill section, where the strip is carefully milled to the required width and thickness. Afterwards, the strip passes through the run-out table to be cooled to an appropriate final temperature. Finally, the strip is coiled and is ready for shipment.

In the HRP, a key performance indicator is the final strip thickness, which is determined by the finishing mill process. Thus, it makes sense to focus the modelling and control on the finishing mill.

Figure 2 shows a diagram of the finishing mill rolling process (FMRP) that consists of 7 groups of stands. In each group of stands, there are 4 rollers: two that work directly on the strip and two that support the working rollers.[22] Before a strip reaches a given stand, the rolling force for the upper supporting roller is computed based on the desired rate of thickness reduction, while the bending force is found using an empirical equation. The actual change in the steel thickness is determined by the rolling force, temperature, bending force, and other physical properties that depend on the specific steel batch.[22] Thus, a first principles model can be difficult to obtain. In practice, the individual stands are not autonomous, but are combined together using various control methods. After the last stand, the steel thickness is measured using an online, X-ray device, that is located at a distance from the stands.[22] The difference between the measured and setpoint thicknesses can then be fed back to the current or previous stands to adjust the milling force. It should be noted that the distance between the X-ray measurement sensor and the stands introduces time delay into the feedback control system. Furthermore, since the thickness cannot be calculated between two stands, the gap measurement between the two working rollers can be used as a proxy variable. It has been found that the thickness of the steel strip after each stand is approximately equal to the gap between the working rollers plus the ratio between the applied force and the roller's stiffness.[22] However, since the stiffness can be difficult to calculate precisely, it is impossible to adjust the downstream stands based on the upstream thickness.

RESULTS AND DISCUSSION

The soft sensor model developed using the LS-SVM method will be developed for the finishing mill process using industrial data available from the system. Comparison of the results against different batches with and without a bias update term will be considered. Furthermore, the impact of time delay on the system will be considered.

Modelling Parameters

In the hot steel rolling mill process, each batch is assumed to be a single coil of steel. Each batch, or coil of steel, is assumed to have had a similar preparation and target specifications, for example, a single batch may contain a specific type of steel that has been processed in a given manner that will be rolled into a given thickness. This implies that the model must be able to handle different initial conditions, as well as varying requirements.

When performing SVM analysis, there are three key steps: model training, model validation, and model testing.[13] In model training, the model parameters are determined, while model validation seeks to determine if the given model structure is appropriate. Finally, model testing seeks to test the final model on another data set to determine its predictive properties. Ideally, for each step a different data set would be used. However, in practice, this may not be feasible. In general, the model training and validation steps often use the same data set. In such cases, other methods, such as k-fold cross-validation, are used to provide an approximation to a different data set. In k-fold cross-validation, the data set is split into k equally sized subsets. One of the k subsets is selected as the validation data set, while the remaining subsets are used as the training data set. This is repeated so that all k subsets are used once as the validation subset.

A single batch, that is, the result of processing a single complete coil, was selected to be both the training and validation data sets. Selecting a complete run corresponding to a single coil eliminates the need to consider the changes of the process over the course of a single coil run. The data from batch #4 was used for training and validating the model, while testing was performed using batches #3, 7, and 8. A Gaussian radial basis function kernel was selected as the basis function. Other basis functions were tested, but they did not provide any significant improvement to the results. Tenfold cross-validation was used to select the best model given the training data set.

For each batch, a total of 42 variables are available, that is, for each of the 7 stands, 6 separate variables are measured: the working roll force, the driving roll force, the setpoint for the working roll force, the total force placed on the top of the stand, and the speed feedback value. The sampling time is 0.1 s. The model parameters were set to $\gamma = 1.6709 \times 10^5$ and the final dimension was selected to be 1253.

Single Batch Investigation

The first set of examples will consider the issue of modelling the hot steel rolling mill using a single batch of data to build the model and compare the predictions against the different batches.

Batch #4 was used to develop the model since it was representative of the largest group of similar thicknesses. For each batch comparison, the mean forecast error (MFE), defined as follows:[23]

$$\mathrm{MFE} = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \hat{y}_{m,i}\right| \tag{11}$$

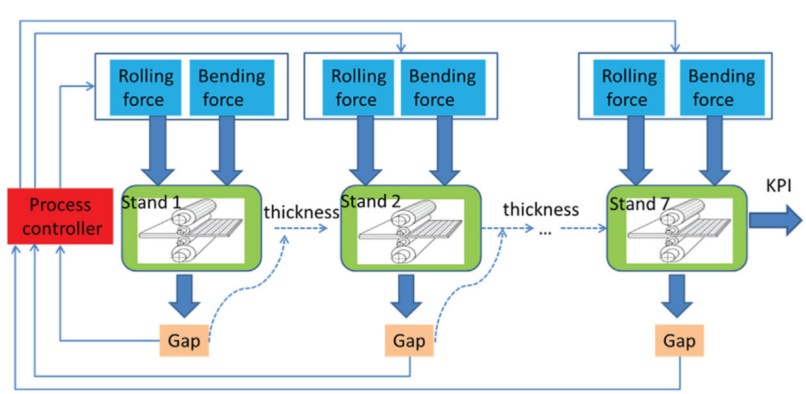

Figure 2. Diagram of the finishing mill process.

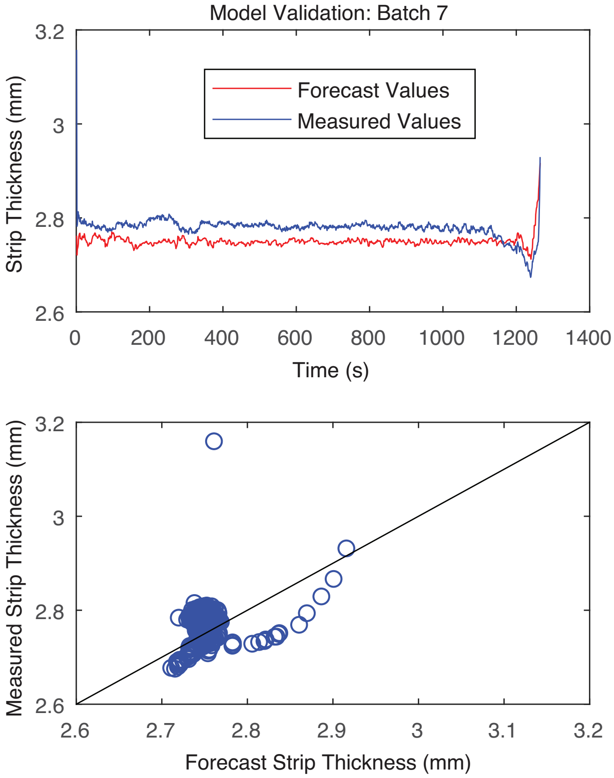

Figure 3. Model validation using batch 7. (top) Time series plot of the forecast and measured values; (bottom) measured values as a function of the forecast ones.

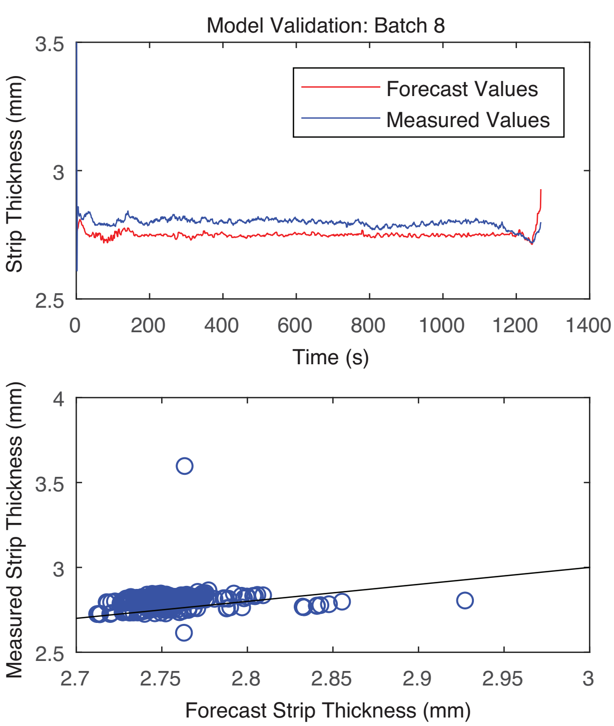

Figure 4. Model validation using batch 8. (top) Time series plot of the forecast and measured values; (bottom) measured values as a function of the forecast ones.

where N is the total number of data points, was computed. The smaller the mean forecast error, the better the fit.

Model testing was performed using batch #7, which had the same nominal setpoint thickness. The testing results are presented in Figure 3. From Figure 3, it can clearly be seen that, although the developed model in general tracks the overall changes in the process well, the predicted values are offset from the measured values even though the strip thickness setpoints were similar.

Figure 4 and Figure 5 show the forecast values using the above model for two different batches. Both a time series plot of the data, and a scatter plot where the measured values are on the x-axis and the forecast values are on the y-axis, are shown. This implies that if the fit is perfect all the values will lie along the $y = x$ line (shown as the solid black line). Deviations from this behaviour can be used to determine the underlying issues with the model. From Figure 4, it can be seen that for batch #8, the model can accurately capture the dynamics of the process, but there is a substantial offset. On the other hand, Figure 5 shows that for batch #3, there are significant deviations from the expected behaviour. However, even in this case, it would seem that it could be a case of determining the appropriate nonconstant bias update.

Table 1 shows the mean forecast errors, computed using Equation (12). As expected the mean forecast error is smallest for batch #7 and largest for batch #3. These results agree well with the observations made from both the time series and scatter plots.

Based on the above results, it can be concluded that the model can, in general, obtain an accurate estimate of the process direction or change. However, the model is unable to correctly predict the absolute process value.

Single Batch Investigation with Bias Update Term

In this case, a simple bias update term will be designed assuming that there are no time delays in the system. In order to design the bias update term, it is necessary to first note that specific conditions must

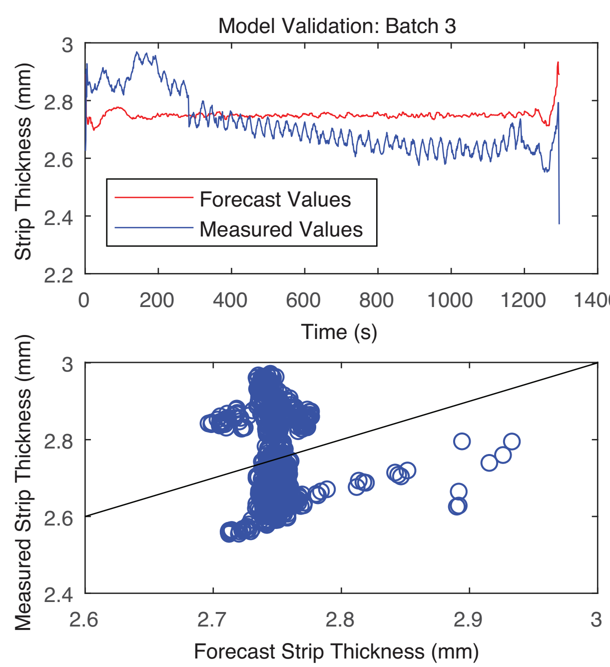

Figure 5. Model validation using batch 3. (top) Time series plot of the forecast and measured values; (bottom) measured values as a function of the forecast ones.

Table 1. Mean forecast error for the base case	
Batch	Mean Forecast Error
#3	0.0941
#7	0.0330
#8	0.0488

Table 2. Mean forecast error for the bias update case	
Batch	Mean Forecast Error
#3	0.0937
#7	0.0328
#8	0.0487

be satisfied in order that the bias update term provide appropriate tracking. Assuming that the bias update term can be written as follows:

$$G_B = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}}{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_m z^{-m}} = \frac{\sum_{i=0}^{n} a_i z^{-i}}{\sum_{i=0}^{m} b_i z^{-i}} \tag{12}$$

and $b_0 \neq 0$, then the following conditions need to be satisfied for satisfactory tracking:[18]

1) $\Sigma b_i = 0$, and
2) An integrator must be present in the bias update term, that is, the denominator must contain the root $z = -1$.

The two additional constraints can be rewritten as follows:

$$\begin{cases} \alpha_k - b_k \neq 0 \\ \alpha_1 - b_1 = 0 \ \forall l \neq k \end{cases} \tag{13}$$

where k is an arbitrarily selected integer located between 0 and $\max(m, \ n)$. Simplifying the condition allows for a faster determination of an optimal bias update term. For the time delay case, setting k equal to the delay provides the best tracking and a simple solution.

Therefore, using these constraints and assuming that the time delay is not relevant, the bias update term can be written as follows:

$$G_B = \frac{z^{-1}}{1 - z^{-1}} \tag{14}$$

Using the same model as before, the current measured values as a function of the current forecast values is shown in Figure 6. Unlike in the previous case, most of the data points now lie along the $y = x$ line. This suggests that a simple bias update term can help eliminate the bias problem previously noted. Furthermore, Table 2 shows the MFE for this situation. As expected, the MFE has decreased compared to the original case. This confirms that including the bias update term improves performance. However, there are practical issues with such an implementation, since the measured process values are often delayed and not available as required. Therefore, it is important to consider this final problem in determining the implications of using a single model for modelling the process.

Single Batch Investigation with Bias Update and Time Delay

The final example will consider the impact on forecast accuracy of adding a time delay to the measurement of the key performance indicator of 10 samples. This is close to the mean value of the time delay in the actual process. It can be noted that the true time delay varies depending on the process conditions. Using the conditions provided, the bias update term in this case can be written as follows:

$$G_B = \frac{z^{-10}}{1 - z^{-10}} \tag{15}$$

Figure 7 shows the measured values as a function of the forecast values. Firstly, it should be noted that until the first measured value arrives there is no way of knowing the appropriate correction to apply. Therefore, the first 10 samples, although shown, will be equivalent to the case where there is no bias update. Secondly, even in the presence of time delay, the bias update term is able to accurately track the process changes. In fact, the overall accuracy has not changed from the previous case. This suggests that the bias correction is almost constant for a given batch and only varies between batches depending on the initial conditions and other external factors. It can be noted that Batch 3 seems to display some rather peculiar behaviour in that there are two regions with more or less constant bias. The reason for this difference is not known.

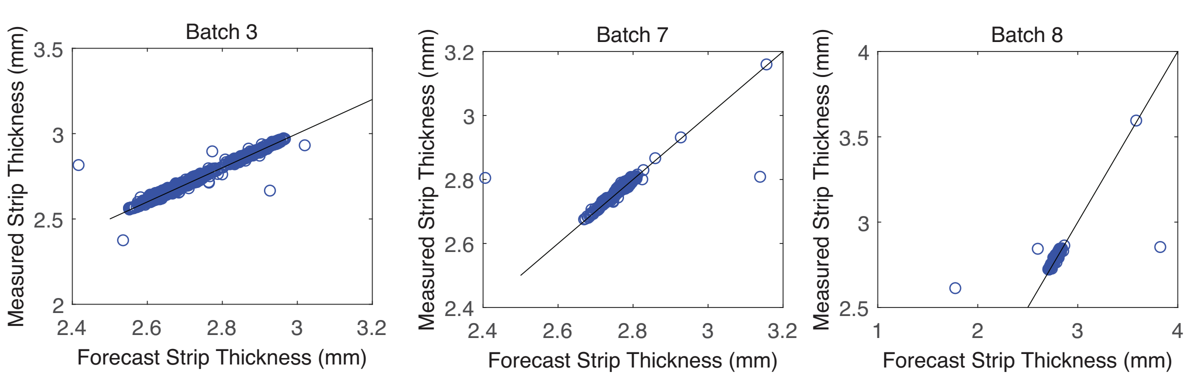

Figure 6. Current measured strip thickness as a function of the current forecast values for the case where a bias update term is present.

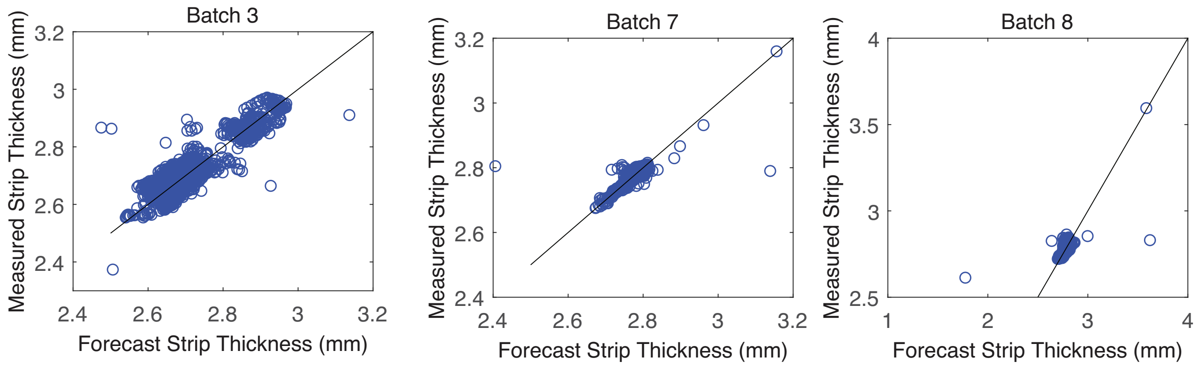

Figure 7. Measured strip thickness as a function of forecast values for the case where a bias update term is present and a delay of 10 samples is assumed.

Table 3. Mean forecast error for delay and bias update case

Batch	Mean Forecast Error
#3	0.0926
#7	0.0324
#8	0.0481

Table 3 gives the mean forecast error for this case. It can be seen that the mean forecast errors have decreased. The largest drop can be observed for batch #3, since initially the performance was much lower.

Thus, implementing an appropriate bias update term can have the potential of allowing a single process model to be applied over a wide range of different process conditions and still obtain a good fit.

CONCLUSIONS

This paper has examined the design of soft sensors for the hot steel rolling mill using least-squares support vector machines and bias updates. It was shown that, although the individual batches may have different behaviours, implementing a soft sensor with a bias update term can accurately model the overall process. Furthermore, the presence of time delay in the measured data was shown not to have an impact on the ability to obtain accurate process values. Future work will consider the problem of trying to estimate the time delay for an individual batch using available process information.

ACKNOWLEDGEMENTS

Yuri Shardt would like to thank the “Programms zur Förderung des exzellenten wissenschaftlichen Nachwuchses” of the University of Duisburg-Essen for funding parts of this research. Xu Yang would like to thank the National Natural Science Foundation of China (Grant #61673053), the Beijing Natural Science Foundation (Grant #4162041), and the Fundamental Research Funds for Central Universities (Grant #FRF-BR-16-025A) for funding. Johan Suykens and Siamak Mehrkanoon acknowledge support from KU Leuven, the Flemish government, FWO, the Belgian Federal Science Policy Office, and the European Research Council (ERC Project AdG A-DATADRIVE-B (290923), CoE PFV/10/002 (OPTEC), GOA MANET, IUAP DYSCO, FWO Grant No. G.0377.12).

REFERENCES

[1] Y. A. W. Shardt, Y. Zhao, F. Qi, K. H. Lee, X. Yu, B. Huang, S. Shah, *Can. J. Chem. Eng*. **2012**, *90*, 217.
[2] D. Slaughter, *Strip Crown Prediction: Developing a Refined Dynamic Roll-Stack Model for the Hot Rolling Process*, PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg **2009**.
[3] K. Peng, K. Zhang, B. You, J. Dong, *Neurocomputing* **2015**, *168*, 1094.
[4] Y. A. W. Shardt, H. Hao, S. X. Ding, *IEEE T. Ind. Electron*. **2015**, *62*, 3843.
[5] Y. A. W. Shardt, X. Yang, “Development of Soft Sensors for the Case Where the Time Delay is Random,” *DYCOPS, IFAC*, Trondheim, 6–8 June **2016**.
[6] X. Shao, B. Huang, J. M. Lee, F. Xu, A. Espejo, *AIChE J*. **2011**, *57*, 1514.
[7] F. Souza, R. Araujo, *IEEE T. Ind. Inform*. **2015**, *10*, 937.
[8] J. Zhu, Z. Ge, Z. Song, *J. Process Contr*. **2015**, *32*, 25.
[9] Z. X. Wang, Q. P. He, J. Wang, *J. Process Contr*. **2015**, *26*, 56.
[10] Y. He, Y. Xu, Z. Geng, Q. Zhu, *Chinese J. Chem. Eng*. **2015**, *23*, 138.
[11] Z. Sun, J. Zhao, Z. Shi, S. Yu, *Mechatronics* **2014**, *24*, 186.
[12] W. Yan, H. Shao, X. Wang, *Comput. Chem. Eng*. **2004**, *28*(8), 1489.
[13] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, *Least Squares Support Vector Machines*, World Scientific Publishing, Singapore **2002**.
[14] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York **1998**.
[15] S. Mehrkanoon, T. Falck, J. A. K. Suykens, *IEEE T. Neur. Net. Lear*. **2012**, *23*, 1356.
[16] S. Mehrkanoon, J. A. K. Suykens, *Neurocomputing* **2015**, *159*, 105.
[17] S. Mehrkanoon, T. Falck, J. A. K. Suykens, “Parameter Esitmation for Time Varying Dynamical Systems using Least Squares Support Vector Machines,” *16th IFAC Symposium on System Identification (SYSID 2012)*, IFAC, Brussels, 11–13 July **2012**.
[18] Y. A. W. Shardt, B. Huang, *Ind. Eng. Chem. Res*. **2012**, *51*, 4958.
[19] Y. A. W. Shardt, B. Huang, *Ind. Eng. Chem. Res*. **2012**, *51*, 4968.

[20] K. Peng, H. Zhong, L. Zhao, K. Xue, Y. Ji, *Int. J. Adv. Manuf. Tech*. **2014**, *72*, 589.

[21] S. X. Ding, S. Yin, K. Peng, H. Hao, B. Shen, *IEEE T. Ind. Electron*. **2013**, *9*, 2339.

[22] K. Peng, K. Zhang, B. You, J. Dong, Z. Wang, *IEEE T. Ind. Electron*. **2016**, *63*, 2615.

[23] J. Liu, D.-S. Chen, J.-F. Shen, *Ind. Eng. Chem. Res*. **2010**, *49*, 11530.

Manuscript received February 11, 2017; revised manuscript received June 15, 2017; accepted for publication June 26, 2017.