



<http://www.diva-portal.org>

Preprint

This is the submitted version of a paper presented at *20th IEEE International Conference on Industrial Technology (ICIT), 13-15 February 2019, Melbourne, Australia.*

Citation for the original published paper:

Mattsson, P., Zachariah, D., Björsell, N. (2019)

Flexible Models for Smart Maintenance

In: *Proceedings 2019 IEEE International Conference on Industrial Technology (ICIT)*
(pp. 1772-1777). IEEE

<https://doi.org/10.1109/ICIT.2019.8754932>

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:hig:diva-30445>

Flexible Models for Smart Maintenance

Per Mattsson
Dept. of Electrical Engineering,
Mathematics and Science
University of Gävle
Gävle, Sweden
per.mattsson@hig.se

Dave Zachariah
Dept. of Information Technology
Uppsala University
Uppsala, Sweden
dave.zachariah@it.uu.se

Niclas Björsell
Dept. of Electrical Engineering,
Mathematics and Science
University of Gävle
Gävle, Sweden
niclas.bjorsell@hig.se

Abstract—Smart maintenance strategies are becoming increasingly important in the industry, and can contribute to environmentally and economically sustainable production. In this paper a recently developed latent variable framework for nonlinear-system identification is considered for use in smart maintenance. A model is first identified using data from a system operating under normal conditions. Then the identified model is used to detect when the system begins to deviate from normal behavior. Furthermore, for systems that operate on separate batches (units), we develop a new method that identifies individual models for each batch. This can be used both to detect anomalous batches and changes in the system behavior. Finally, the two methods are evaluated on two different industrial case studies. In the first, the purpose is to detect fouling in a heat exchanger. In the second, the goal is to detect when the tool in a wood moulder machine should be changed.

Index Terms—Modeling, Nonlinear systems, Applications

I. INTRODUCTION

In the continuing effort to implement Industry 4.0, the move from traditional time-based maintenance towards smart maintenance plays a key role. In time-based maintenance, maintenance is performed based on a calendar schedule. However, as noted in e.g. [4], the time to failure for identical components can vary widely, even under more or less the same conditions. Hence, a pure time-based schedule for maintenance would either lead to more maintenance than needed, that failing components are detected too late, or both. Due to the unreliability of time-based maintenance methods, there could be a large gain from using online monitoring throughout the life-time of the equipment, in order to identify the onset of degradation and failure [12].

Recent development in the industry has also made it possible to deploy Cyber-Physical Systems (CPS) in a systematic way [7]. These are technologies for integration of computational and physical capabilities [1]. In general, a CPS consists of both advanced connectivity for data acquisition and feedback, as well as data analysis and intelligent data management. In the 5-level CPS structure of [7], an important part is played by the so-called cyber-twins, which are responsible for using

This project is conducted in cooperation with SSAB and Svenska Fönster AB within Produktion2030 - a strategic innovation program supported by: VINNOVA, Swedish Energy Agency and Formas. This work was partly supported by the European Commission within the European Regional Development Fund, through the Swedish Agency for Economic and Regional Growth, and partly by Region Gävleborg.

collected data in order to monitor the status of each system, which could be a component or machine.

The main focus of this paper is to show how the cyber-twin can use collected data in order to build up an internal model of the system, and how to use this model in order to warn for degradation and/or failures. The methods discussed are also tested on two real-world industrial cases; a heat exchanger and a wood moulder machine.

There are many ways in which a model for a dynamical system can be constructed. Lets consider the example of a heat exchanger in some detail, before we look into the more general case. One possibility would be to construct a model directly from physical principles. For the heat exchanger, this could be done using the first and second laws of thermodynamics [2]. However, such a theoretical analysis typically involves restrictive assumptions, and has to be redone for each component and/or machine that is introduced into the factory. This is not ideal for the purposes of smart maintenance. Another popular strategy in thermal analysis of heat exchanger is to use artificial neural networks (ANN) [3], [10]. The main advantage is that it avoids assumptions made in physical modeling that are often not satisfied in practical applications, and instead directly learns the nonlinear relations between inputs and outputs from measured data. However, in [10], it is noted that for successful applications of standard ANN-methods the user must carefully choose parameters, such as the number of neurons, learning rate, number of iterations etc., to avoid overfitting the data.

The above discussion indicates the importance as well as limitations of data-driven methods in smart maintenance. In a factory with a diverse set of components and machines, it would be next to impossible to construct models for each part directly from physical laws, without introducing unrealistic assumptions. Hence the need for methods that can learn the model directly from measured data. In order to handle a diverse set of complex components and machines with nonlinearities and correlated noise sources, the considered model structures need to be flexible. However, as noted in [10], a very flexible model structure, such as different types of ANN, may lead to problems with overfitting.

An intuitive way for avoiding problems with overfitting is to use the principle of parsimony, i.e., that the simplest solution is typically the best. So, if a linear model describes the

measured data well enough for our purposes, it is often better than a nonlinear model. While the principle of parsimony mainly is a useful heuristic, it can also often be justified by statistical considerations [13], [5]. Now, the idea of a flexible model structure seems to go against the principle of parsimony. This is one reason for why different regularization techniques, such as LASSO [15], have gained in popularity. Using such techniques, both the model's ability to explain the measured data and the model complexity can be balanced simultaneously.

Another problem with methods using flexible model structures is, as noted in e.g. [10], that they often include a large number of parameters that the user has to tune in order to train the model. Applying a regularization technique like LASSO also introduces extra regularization parameters. There are often no clear guidelines for how these parameters should be selected, since they typically have no direct relation to the physical system. This may lead to time-consuming trial and error of different values for the tuning parameters, and thus reduces the applicability of such methods.

The starting point in this paper is the latent variable framework (LAVA) [9], which can handle flexible model structures using a data-adaptive regularization, meaning that the regularization parameters are learned automatically from the data instead of being tuned by a user. This facilitates the learning of a diverse set of different components and machines. It also follows the principle of parsimony by, in a sense, favoring linear models and only adding nonlinear terms if needed. Furthermore, the implementation of this method can be done in a recursive way, meaning that the model can efficiently be updated every time a new data sample is measured. Hence, in a CPS, this means that the cyber-twin of the system can easily update the internal model every time a new data sample is received.

Furthermore, this paper uses similar ideas as in [9] to develop a new method for identifying a unique model for each batch (or unit) that is processed in some machine, even though only a few data samples are collected from each batch. It is shown how this can be used to determine if the behaviour of the machine changes between batches. This can be used to detect deficient batches, but also degradation of the machine.

The layout of the paper is as follows. In Section II the method in [9] is presented, together with a discussion on how to apply it to smart maintenance. In Section III a similar idea is used in order to develop a new method for detecting dynamical changes between batches/units. In Section IV, the two methods are applied to two real-world industrial cases. The paper ends with a few conclusions in Section V.

II. FLEXIBLE MODEL STRUCTURE

In order to develop a model for the system under consideration, the LAVA-framework [9] has been considered. The model structures in this framework can be very flexible and account for both nonlinearities and correlation in the noise sources. In order to handle the flexibility, data-adaptive regularization is used in a way that promotes parsimonious models. This

framework has already been applied to several real-world systems, see e.g. [8], [9].

In order to describe the model, let \mathcal{D}_t contain the first t samples of the inputs and outputs. The idea here is to start with a simple linear model, which can be written as

$$\mathbf{y}(t) = \Theta\boldsymbol{\varphi}(t) + \boldsymbol{\varepsilon}(t), \quad (1)$$

where $\mathbf{y}(t) \in \mathbb{R}^{n_y \times 1}$ are the outputs at sample t , $\Theta \in \mathbb{R}^{n_y \times n_\varphi}$ contains the unknown parameters, $\boldsymbol{\varepsilon}(t)$ are the prediction errors and $\boldsymbol{\varphi}(t) \in \mathbb{R}^{n_\varphi \times 1}$ is a regression vector that contains past inputs and outputs. A nominal predictor for this model can then be written as

$$\hat{\mathbf{y}}_0(t) = \Theta\boldsymbol{\varphi}(t). \quad (2)$$

In identification of standard linear ARX-models, it is assumed that the prediction error $\boldsymbol{\varepsilon}(t) = \mathbf{y}(t) - \hat{\mathbf{y}}_0(t)$ is a white noise process with a Gaussian distribution. The maximum likelihood-estimate of Θ is then reduced to computing the linear least squares solution. While the white noise-assumption can often be a good approximation, at least around some fixed operation point, this type of identification fails to take into account possible nonlinearities etc. In order to consider more flexible model structures, a different model of the prediction errors was proposed in [9].

A. A data-driven prediction error model

In [9], nonlinearities and data-dependent correlation was included using the following conditional model for the prediction errors,

$$\boldsymbol{\varepsilon}(t) | \mathcal{D}_{t-1}, \mathbf{Z} \sim \mathcal{N}(\mathbf{Z}\boldsymbol{\gamma}(t), \boldsymbol{\Sigma}), \quad (3)$$

where $\boldsymbol{\Sigma}$ is an unknown covariance matrix, $\mathbf{Z} \in \mathbb{R}^{n_y \times n_\gamma}$ is a latent variable, and $\boldsymbol{\gamma}(t) \in \mathbb{R}^{n_\gamma \times 1}$ is a known function of \mathcal{D}_{t-1} . By choosing $\boldsymbol{\gamma}(t)$ as some general basis expansion, $\mathbf{Z}\boldsymbol{\gamma}(t)$ can express a wide range of functions. However, to make the basis expansion long enough, $\boldsymbol{\gamma}(t)$ must have high dimensions, which may lead to problems with e.g. overfitting. This problem is handled by data-adaptive regularization using a prior belief distribution,

$$\text{vec}(\mathbf{Z}) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda}), \quad (4)$$

where $\text{vec}(\cdot)$ is the vectorization operation, and $\boldsymbol{\Lambda}$ is an unknown covariance matrix. The rationale for the prior zero-mean is that, with $\mathbf{Z} = \mathbf{0}$ the error model in (3) reduces to the white noise assumption typically used for linear models. However, when data is collected, the posterior belief distribution of \mathbf{Z} is obtained using Bayes' rule,

$$p(\mathbf{Z} | \mathcal{D}_t) = \frac{p(\mathcal{D}_t | \mathbf{Z})p(\mathbf{Z})}{p(\mathcal{D}_t)}. \quad (5)$$

Finally, with this data-driven prediction error model, it is possible to refine the predictor in (2) as

$$\hat{\mathbf{y}}(t) = \mathbb{E}[\mathbf{y}(t) | \mathcal{D}_{t-1}] = \Theta\boldsymbol{\varphi}(t) + \hat{\mathbf{Z}}\boldsymbol{\gamma}(t) \quad (6)$$

where $\hat{\mathbf{Z}} = \mathbb{E}[\mathbf{Z} | \mathcal{D}_{t-1}]$. Now, in order to use this predictor, we must first identify the parameters Θ and the covariance

matrices Σ and Λ . In [9] it is shown how this can be done in a recursive manner by utilizing the majorization-minimization approach [17] in order to maximize the likelihood $p(\mathcal{D}_N|\Theta, \Sigma, \Lambda)$. It was also seen that the resulting convex optimization problem leads to estimates $\hat{\mathbf{Z}}$ that are sparse, meaning that most elements in $\hat{\mathbf{Z}}$ are typically estimated to equal zero. This property leads to simple explanation of the data, in accordance with the principle of parsimony.

B. Smart maintenance

The method described above can be applied to smart maintenance in several different ways. Since the method can compute the estimated parameters recursively, it is possible to update the parameters each time a new sample is measured from the system in an efficient way. Hence, it is possible to study how the systems parameters change over time, and thus detect when the system starts to deviate from normal conditions. Using black-box models, however, the parameters do not have any explicit physical meaning, and therefore changes in the parameters cannot be readily interpreted.

A more straightforward approach may be to identify a model for the system under normal conditions. Thanks to the flexibility of the model presented in this section, it can be applied to a diverse set of tools and machines, and thus the engineer can avoid time-consuming physical modeling. At the same time, the number of tuning parameters is kept low. The estimated model is then used to simulate the output of the system. The simulated output is finally compared with the actual measured output from the system. If the actual output starts to deviate from the simulated output, this indicate that the system is not operating under normal conditions anymore, and maintenance may be needed. See Section IV for an example where this approach has been tested on an industrial heat exchanger.

III. MULTIPLE MODELS

In manufacturing, it may be interesting to see if a system (machine, component or tool) changes its behavior between batches (or units). This change in behavior could indicate deterioration of the system, or some problem with the current batch. In this section we use a similar idea as in Section II in order to estimate an individual model of the system for each processed batch. By studying deviations in the models, bad batches and system deterioration can be detected.

Let $\mathcal{D}_{i,t}$ contain all inputs and outputs collected during the first t data samples of batch i , and model the system as

$$\mathbf{y}_i(t) = \Theta \varphi_i(t) + \varepsilon_i(t). \quad (7)$$

where $\mathbf{y}_i(t)$ is the output at sample t from batch i , $\varphi_i(t)$ is a regression vector containing past inputs and outputs, and $\varepsilon_i(t)$ is the nominal prediction error.

Let Θ be a nominal model identified during normal operating conditions. However, the dynamics changes when the systems performance deteriorate, or a bad batch is processed.

In order to handle changes in dynamics between batches, consider an error model on the form

$$\varepsilon_i(t)|\mathcal{D}_{i,t-1}, \mathbf{Z}_i \sim \mathcal{N}(\mathbf{Z}_i \varphi_i(t), \Sigma_i). \quad (8)$$

Note that the optimal predictor for this model can be written as

$$\hat{\mathbf{y}}_i(t) = E[\mathbf{y}_i(t)|\mathcal{D}_{i,t-1}] = (\Theta + \hat{\mathbf{Z}}_i) \varphi_i(t), \quad (9)$$

where $\hat{\mathbf{Z}}_i = E[\mathbf{Z}_i|\mathcal{D}_{i,t-1}]$. For batch i , the deviation from dynamics under normal operating conditions is captured by $\hat{\mathbf{Z}}_i$. Hence, we would expect $\hat{\mathbf{Z}}$ to be close to zero as long as the system is behaving in a proper way. Now, the amount of data collected from each batch may be relatively small, so some regularization is needed in order to identify a separate model for each batch. As in Section II, this regularization will be performed using a prior belief distribution on the form

$$\text{vec}(\mathbf{Z}_i) \sim \mathcal{N}(\mathbf{0}, \Lambda_i). \quad (10)$$

Note that the zero mean indicates that we, prior to seeing any data, believe that the dynamics of the system while processing the batch would on average be the same as for the nominal model. However, as data is collected from batch i , the distribution of \mathbf{Z}_i is updated through (5).

In order to determine the full model for each unit, the nominal parameters Θ as well as the individual covariance matrices Λ_i and Σ_i have to be determined. For Θ , the solution in this paper is to identify on batch data where it is known that the system behaves in a proper way, and thus it can be assumed that $\hat{\mathbf{Z}}_i \equiv \mathbf{0}$ for this portion of the data, and the identification can be performed using linear least squares. When a nominal model of Θ has been found, the posterior mean and covariance of \mathbf{Z}_i can be computed using estimates of Λ_i and Σ_i , cf. [9].

A. Smart maintenance

The goal here is to determine whether the dynamics during batch i is significantly different from the dynamics during proper conditions. If they are significantly different, the batch and system should be examined and maintenance may be needed. So, the question is then, how do we determine if there is a significant change in dynamics? Another way to phrase this question is; how large is \mathbf{Z} ?

Using the method described in this section, the posterior of the latent variable $\mathbf{z}_i = \text{vec}(\mathbf{Z}_i)$ is a Gaussian distribution with mean $\hat{\mathbf{z}}_i = \text{vec}(\hat{\mathbf{Z}}_i)$ and covariance \mathbf{R}_i .

The latent variable \mathbf{z}_i falls within the set

$$\mathcal{Z} = \{\mathbf{z}_i : \|\mathbf{z}_i - \hat{\mathbf{z}}_i\|_{\mathbf{R}_i^{-1}} \leq c\} \quad (11)$$

with high probability. We can then test whether $\mathbf{z}_i = \mathbf{0}$ belongs to this set by simply evaluating

$$D_i = \|\hat{\mathbf{z}}_i\|_{\mathbf{R}_i^{-1}}. \quad (12)$$

When this standardized metric exceeds some threshold, we can credibly exclude $\mathbf{z}_i = \mathbf{0}$ and thereby conclude anomalous dynamics. The threshold can be set using the quantile function of the χ^2 -distribution or manually based on application specific tolerances.

IV. CASE STUDIES

The two case studies are a heat exchanger at SSAB and a wood moulder machine at Svenska Fönster AB. Measured data from the plants have been acquired from existing control systems together with a complementary measurement system for the wood moulder machine. The recursive implementation of LAVA enables online monitoring of both systems. Here, however, the data has first been collected and processed offline. Analysis of the results has been done jointly by model developers and staff with process knowledge.

A. The Heat Exchanger

This case study concerns a heat exchanger that needs to be cleaned due to fouling with some regularity. The heat exchanger shall provide the process with cooling water. Water from a nearby river is used as cooling media. Fouling will reduce the cooling efficiency and increase the pressure drop. Of these two, the reduced cooling efficiency is the most dominant effect [16], [6], [11].

Thus, the inputs $\mathbf{u}(t)$ to the predictor model consist of temperature in the process and river water before the heat exchanger and the flow of process and river water. The output $y(t)$ is the outgoing temperature on the process side. For this case study the data was sampled once every minute for 184 working hours. All measurements are taken from an existing control system.

1) *Model identification:* In order to find a model for the heat exchanger, the method presented in Section II was used. The linear regression vector was chosen as

$$\varphi(t) = \begin{bmatrix} y(t-1) \\ \mathbf{u}(t-1) \\ 1 \end{bmatrix}, \quad (13)$$

where the purpose of the 1 in the last row is to model bias in the system. The nonlinear function $\gamma(t)$ is a function of $\varphi(t)$, and the Laplace basis expansion was used due to its good approximation properties [14], [9]. The expansion was chosen so that $\hat{\mathbf{Z}}$ contained 1024 elements. However, the method is insensitive to the number of elements in $\hat{\mathbf{Z}}$, so increasing the size of $\hat{\mathbf{Z}}$ would not make much of a difference in the result.

The first 24 hours of data, i.e. 1440 samples, were used for identifying the model. After the identification, only 5 of the elements in $\hat{\mathbf{Z}}$ was estimated to be non-zero, showing that the identification method finds a parsimonious model for the heat exchanger.

2) *Result:* To illustrate the identified model, the output of the model has been simulated using only inputs. The observed input $y(t)$ and the simulated output $\hat{y}_s(t)$ during the second day after cleaning is shown in Figure 1. It can be seen that the model performs well during this day. However, as time passes the fouling in the heat exchanger increases, and we see in Figure 2 that the model systematically underestimates the output temperature the last day before cleaning. For sake of clarity, Figure 3 shows a moving average of the simulation error $y(t) - \hat{y}_s(t)$. The result provides operational information about how much the cooling in the heat exchanger

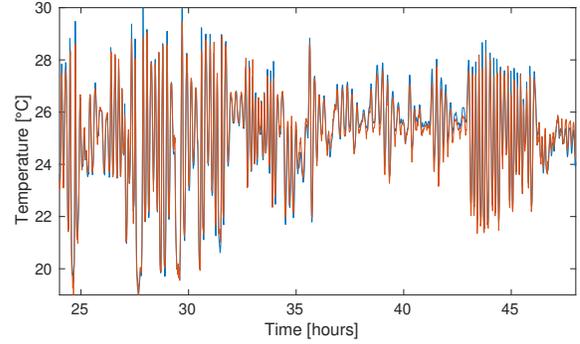


Fig. 1. The measured (blue) and simulated (red) temperature for day 2 after cleaning.

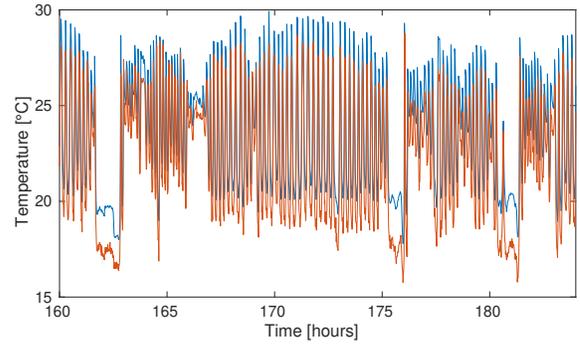


Fig. 2. The measured (blue) and simulated (red) temperature for the last day before cleaning.

can be improved by cleaning. In further studies, data from a longer period should be considered in order to include seasonal variations on the temperature etc. in the data. Since the proposed method is recursive, it would be easy to switch on identification for some time whenever the heat exchanger has been cleaned in order to improve the model.

B. The Wood Moulder Machine

In this section we study a Gabbiani moulder machine, where cutting tools today are changed after a predefined number of meters of machined material. The data collection

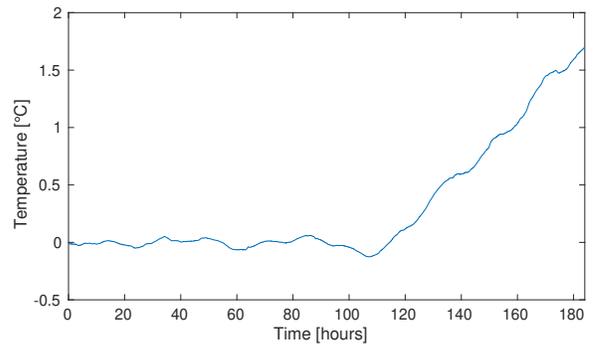


Fig. 3. The moving average of the simulation error $y(t) - \hat{y}_s(t)$.

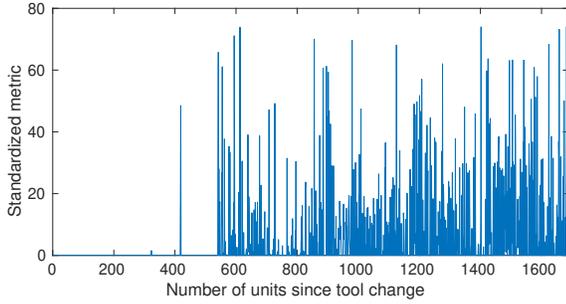


Fig. 4. The standardized metric D_i .

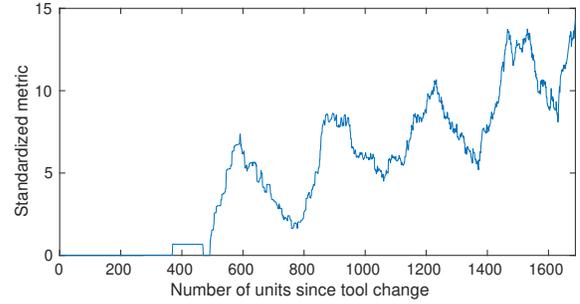


Fig. 5. A moving average of D_i .

is a combination of new measurement and existing signals from the control system. The new measurement signals are motor currents; these were collected with an IbaDAQ-S at a sampling interval of 10 ms. From the control system the following data was collected: Tool number, feed rate, tool speed, presence of material (true/false). In addition a log book for tool replacement was used as input data. Tool numbers are used to sort data and data from the logbook provide information about when a tool is fresh. The two speeds (feed rate and tool speed) are controlled to be constant and these measurements are only used to verify this. Thus, the motor current $y(t)$ is the only signal used in the model. Here we use the approach discussed in Section III, and consider the signal model in (7) with

$$\varphi(t) = \begin{bmatrix} y(t-1) \\ y(t-2) \\ 1 \end{bmatrix}. \quad (14)$$

In order to estimate the nominal parameters Θ , the data collected from the first 100 units processed after a tool change was used. After this, the error model for each of the 1688 units that were processed before the next tool change were identified. As discussed in Section III, the statistic D_i provides a measure of how much the dynamics deviates from normal operating conditions. This is plotted for each unit in Figure 4. It can be seen that for about 500 units after the tool change, the dynamics is well described by the nominal parameters Θ . After this D_i starts to deviate from zero. This does not necessarily mean that the tool must be changed directly, since the tool does not need to be in perfect condition in order to work well enough. However, if the distance starts to be consistently large, this indicates that the tool should be inspected. To see that this happens for the case study, Figure 5 shows the moving average over 100 units of D_i . It can be seen that this distance is increasing as more units are processed by the tool. Since the main difference that happens during this time is that the tool gets more blunt, this indicates that the measure indeed capture the need for changing the tool. This also opens up the possibility to determine a threshold when the tool should be changed. However, in order to determine such a threshold, data from a tool that is too blunt would be needed. At this point we do not have access to such data, since all data was collected during a normal production process. The

four peaks that can be seen in Figure 5 appear to correspond to batch changes.

V. CONCLUSIONS

In this paper, two different modeling techniques for smart maintenance has been proposed and examined in two real-world case studies. The first method uses the LAVA-framework in order to detect dynamical changes in a process, and the second method consider changes between batches (or units). The results are promising, even though more studies would be need for these two specific industrial cases in order to determine the exact levels where maintenance is needed and to handle seasonal variations etc. Since the proposed methods do not contain a lot of tuning parameters, they are relatively straightforward to apply. However, process knowledge is still an important part in determining which signals that should be considered in order to detect maintenance need.

REFERENCES

- [1] R. Baheti and H. Gill. The impact of control technology: Cyber-physical systems. *IEEE Control Systems Society*, pages 1–6, 2011. cited By 1.
- [2] RL Cornelissen and GG Hirs. Thermodynamic optimisation of a heat exchanger. *International journal of heat and mass transfer*, 42(5):951–960, 1999.
- [3] Gerardo Diaz, Mihir Sen, KT Yang, and Rodney L McClain. Simulation of heat exchanger performance by artificial neural networks. *Hvac&R Research*, 5(3):195–208, 1999.
- [4] H. M. Hashemian and W. C. Bean. State-of-the-art predictive maintenance techniques. *IEEE Transactions on Instrumentation and Measurement*, 60(10):3480–3492, Oct 2011.
- [5] William H Jefferys and James O Berger. Sharpening Ockham’s razor on a Bayesian strop. *Dept. Statistics, Purdue Univ., West Lafayette, IN, Tech. Rep.*, 1991.
- [6] S. A. Kumari and S. Srinivasan. Narx models for prediction of reheater fouling. In *2017 23rd International Conference on Automation and Computing (ICAC)*, pages 1–4, Sept 2017.
- [7] J. Lee, B. Bagheri, and H.-A. Kao. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*, 3:18–23, 2015.
- [8] Per Mattsson, Dave Zachariah, and Petre Stoica. Identification of cascade water tanks using a pwarx model. *Mechanical systems and signal processing*, 106:40–48, 2018.
- [9] Per Mattsson, Dave Zachariah, and Petre Stoica. Recursive nonlinear-system identification using latent variables. *Automatica*, 93:343 – 351, 2018.
- [10] M Mohanraj, S Jayaraj, and C Muraleedharan. Applications of artificial neural networks for thermal analysis of heat exchangers—a review. *International Journal of Thermal Sciences*, 90:150–172, 2015.
- [11] Kazi S. N. *Fouling and Fouling Mitigation on Heat Exchanger Surfaces*. InTech, 2012.

- [12] J-H. Shin and H-B. Jun. On condition based maintenance policy. *Journal of computational Design and Engineering*, pages 119–127, 2015.
- [13] T. Söderström and P. Stoica. *System Identification*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.
- [14] A. Solin and S. Särkkä. Hilbert space methods for reduced-rank Gaussian process regression, 2014. arXiv preprint arXiv:1401.5508.
- [15] Robert Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):pp. 267–288, 1996.
- [16] P. Wang and R. X. Gao. Automated performance tracking for heat exchangers in hvac. *IEEE Transactions on Automation Science and Engineering*, 14(2):634–645, April 2017.
- [17] T. T. Wu and K. Lange. The mm alternative to em. *Statistical Science*, 25(4):492–505, 2010.