Multivariate Fault Detection using Vector Autoregressive Moving Average and Orthogonal Transformation in Residual Space

Francisco Serdio[1], Edwin Lughofer[2], Kurt Pichler[3], Thomas Buchegger[4], Markus Pichler[5], and Hajrudin Efendic[6]

[1,2] *Department of Knowledge-Based Mathematical Systems, Johannes Kepler University Linz, Austria*
francisco.serdio@jku.at edwin.lughofer@jku.at

[3,4,5] *Austrian Competence Center of Mechatronics, Linz, Austria*
kurt.pichler@lcm.at thomas.buchegger@lcm.at markus.pichler@accm.co.at

[6] *Institute for Design and Control of Mechatronical Systems, Johannes Kepler University Linz, Austria*
hajrudin.efendic@jku.at

Abstract

We propose the use of multivariate orthogonal space transformations and Vector Autoregressive Moving-Average (VARMA) models in combination with data-driven system identification models to improve residual-based approaches to fault detection in rolling mills. Introducing VARMA models allows us to build k-step ahead multi-dimensional prediction models including the time lags that best explain the target. Multivariate orthogonal space transformations provide estimates for the dynamical parameters by rewriting the equation set of the system at hand, decomposing the measured data into process and residuals spaces. Modeling in the process space then produces much more accurate models due to dimensionality (noise) reduction. Since we use an unsupervised scheme that requires a priori neither annotated samples nor fault patterns/models, both model identification and fault detection are based solely on the on-line recorded data streams. Our experimental results demonstrate that our approach yields improved Receiver Operating Characteristic (ROC) curves than methods that do not employ vector autoregressive moving-average models and multivariate orthogonal space transformations.

1. Introduction

Unscheduled machine downtime could be reduced significantly by accurate condition monitoring and early detection of faults. Thus, expensive repair costs could be minimized and production efficiency increased. The concept of *fault detection* was formally defined by the "SAFEPROCESS" IFAC Technical Committee as the "Determination of the faults present in a system and the time of detection". Further, they also defined a *fault* as "An unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable/usual/standard condition" (Isermann & Ballé, 1997). At the time when the Committee introduced the terminology to the fields of supervision, fault detection and diagnosis, most applications supported fault detection by simple threshold logic or hypothesis testing, but the number of publications describing much more complex techniques (such as fuzzy logic or neural networks) was steadily growing. The main challenge in our application is the *detection of faults*, without (i) an analytical description of faults and process models and (ii) a collection of typical fault patterns. Approaches using model-based techniques that rely on analytical process (system) models (Dong, Liu, & Li, 2010) or employ models deduced from the physical definition of the appearance of a fault (Bolt et al., 2010) or take advantage of a robust observer design (Theilliol, Mahfouf, Ponsart, Sauter, & Gama, 2010) are unsuited to our application. Neither are time-series analysis (Chandola, Banerjee, & Kumar, 2009) and vibration monitoring in frequency spaces (spectrograms, etc.) (Pichler, Lughofer, Buchegger, Klement, & Huschenbett, 2012) because the measurement signals are not completely smooth and continuous in their time line appearance but may show jumping patterns even in fault-free states due to, for instance, varying systems states. Pattern recognition and classification approaches (Bishop, 2006) cannot be employed (i) due to the absence of fault patterns and (ii) because annotated samples are not available since simulating real faults directly at the system involves high costs and the risk of component breakdown.

For fault detection in rolling mills, we propose a residual-based approach that builds upon data-driven techniques, combined with vector autoregressive moving-average models and multivariate orthogonal space transformations. Further, we use a dynamic threshold based on a tolerance band that tracks the residuals of the models over time after being normalized by a local error bar. The model-based FD approach takes advantage of the nominal model of the system to generate residuals (deviations between model predictions and measured targets) containing information about the faults. Clearly, the quality of the model is essential for fault detectability and isolability and the avoidance of "false alarms" (Frank, Alcorta, & Köppen-Seliger, 2000).

We conducted our experiments using *process history*-based methods which only require large amounts of historical process data. There are different ways in which this data can be transformed and presented as a priori knowledge to a diagnostic system. Principal Component Analysis (PCA) and Partial Least Squares (PLS) together with statistical pattern classifiers form a major component of statistical feature extraction methods (Venkatasubramanian, Rengaswamy, Kavuri, & Yin, 2003). Section 2 explains how we handle VARMA models and describes both PCA and PLS as preprocessing and dimension reduction tools and how to combine them with linear or fuzzy methods. Section 4 comments briefly on preliminary results which show that the proposed techniques yield better Receiver Operating Characteristic (ROC) curves than previous fault detection methods. Section 5 concludes the paper and presents future research directions.

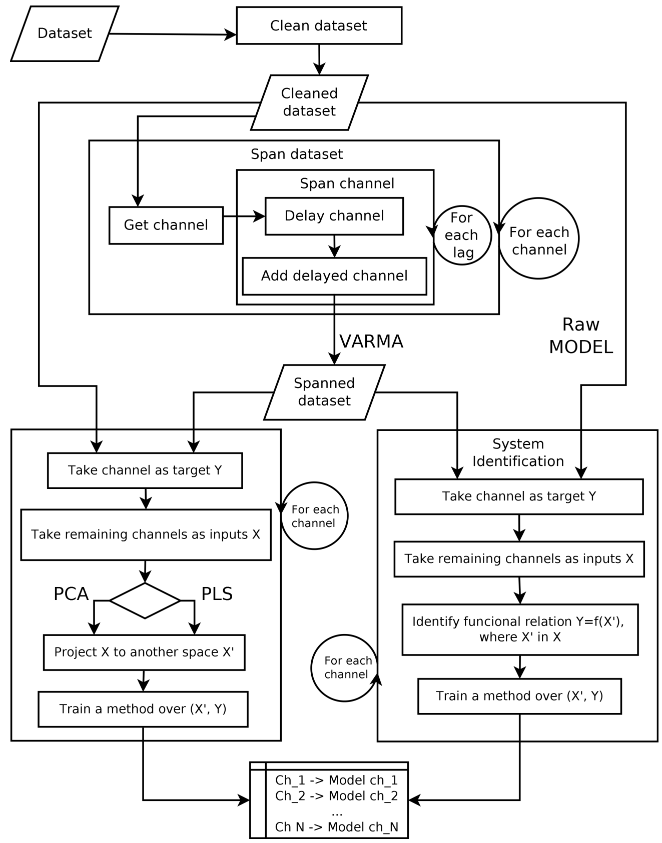


Figure 1. Methodology workflow. All combinations of methods can be obtained following the four different pathways.

2. MODELING

2.1. Methodology step by step

Our methodology at a glance is shown in Figure 1. In the workflow, four different pathways can be follow, thus obtaining the different combinations of models, i.e. the models without any additional issue (raw models), the VARMA models (including the lags of the channels), the models trained on the projected spaces and the VARMA models trained on the projected spaces. Along these pathways, how a dataset is spanned to include lags and how a model can be trained in a projected space is also depicted. The methodology also include a preprocessing step cleaning the data.

2.2. VARMA (Multi-regressive) Models

Inspired by Auto-Regressive Moving Average (ARMA) models, which model a channel by its own lags, we use lags in our multidimensional and multichannel approach. Thus, the dataset containing the candidate channels to explain a target is first spanned with the candidate channels' lags (Figure 1). To this end, each channel in the dataset is delayed several times by different amounts (1 to L in our experiments). Then, the model of each channel including all the channels' lags is learned by means of a modified variant of forward selection (Miller, 2002) over the spanned dataset. Thus, we obtain separate and independent multivariate time series models, $f_i, i = 1, ..., M$.

Starting from the general definition of the model of a channel ch_i as a function f_i of other channels (where no lags are used), and using the notation [t] to express the lag operator (also known as backshift operator, *at time t*), it becomes straightforward to pass from the particular model definition where the set of candidate channels for explaining the target is obtained from the spanned dataset including lags (cf. Figure 1). The functional relation without lags is then

$$ch_i[t] = f_i(ch_1[t], ..., ch_{i-1}[t], ch_{i+1}[t], ..., ch_M[t])\,, \quad (1)$$

while the functional relation with lags is

$$ch_i[t] = f_i(ch_p[t-k], ..., ch_q[t-l], ..., ch_r[t-m])\,, \quad (2)$$

where k varies from 0 (i.e., allowing channels without lags to participate in the model definition) to L, and $p, q, r \neq i$ (lags of ch_i are not included) but not necessarily $p \neq q \neq r$. Thus, models are potentially obtained where no lags, different lags from the same channel, or different lags from different channels appear as input. To do not allow lagged versions of ch_i will avoid introducing the information of the (potentially faulty) channel as an input of the final model.

Previous work using autoregressive models in the fault detection area can be found in Schöener, Moser, and Lughofer (2008), M. Yang and Makis (2010) and T. Yang (2006). These use AR, ARX and ARMA models respectively, but we have not found evidence of VARMA models applied for the purpose of fault detection. However, our results demonstrate that this new functional relation enriches the model set since they yield better ROC curves (see section 4). A detailed description of ARMA models and its variations is provided in Holan, Lund, and Davis (2010).

2.3. (Fuzzy) Principal Components Regression

Principal component analysis (PCA) (Jolliffe, 2002) is a vector space transformation that identifies the most meaningful basis to re-express the original space while preserving maximum variance in a minimum number of dimensions and filtering out the noise. When starting from correlated data, PCA is a good technique to transform the set of original process variables into a new set of uncorrelated variables that explain the main trends of the process. The PCA decomposition is

$$X = \sum_{h=1}^{m} v_h r_h^T + G = VR^T + G, \quad (3)$$

where V, R and G are scores, loadings and residuals, respectively. Since PCA was first introduced as a tool (Jackson & Mudholkar, 1979), numerous publications have described fault detection using PCA, comprising both practical applications (Odgaard, Lin, & Jørgensen, 2008) and theoretical studies (Tamura & Tsujita, 2007). We, however, do not use PCA directly as a fault detection method; instead we go a step further and use a Principal Component Regression (PCR) technique –see Jolliffe (2002). PCR exploits the PCA capabilities as a dimensionality reduction tool in order to produce a new set of regressors to train a linear method with them (see Figure 1). Since the model trained with the PCA scores can be changed (a linear model is used in PCR), several variants are possible. We used both PCR and Fuzzy PCR (FPCR) in our experiments. For FPCR we used a Takagi-Sugeno fuzzy model (TSK) trained with an extended version of *SparseFIS* as explained in Lughofer and Kindermann (2010).

SparseFIS relies on a top-down approach that takes a maximum number of (allowed, feasible) rules as inputs and discards out as many rules as possible to find a compact rule set through an enhanced numerical optimization process which is a non-linear version of the *projected gradient descent algorithm* (Daubechies, Defrise, & Mol, 2004). SparseFIS also uses an extended form of TS fuzzy models, since it integrates rule weights $\rho \in [0,1]$ as rule importance levels, thus preferring some local regions over others.

$$\hat{f}(\vec{x}) = \hat{y} = \sum_{i=1}^{C} \Psi_i(\vec{x}) \cdot l_i(\vec{x}) \quad \Psi_i(\vec{x}) = \frac{\rho_i \varphi_i(\vec{x})}{\sum_{j=1}^{C} \rho_j \varphi_j(\vec{x})}, \quad (4)$$

where l_i are the linear consequent functions and $\varphi(\vec{x})$ denotes the activation degree of the i-th rule. The analytical model of FPCR results from substituting the inputs in (4) by those transformed (and usually reduced) by the PCA orthogonal projection in (3).

2.4. (Fuzzy) Partial Least Squares

Partial Least Squares (PLS) handles correlated inputs, reducing the dimensionality of the input and target variables by projecting them to the directions maximizing the covariance between them. The projection combines highly correlated variables into new one-dimensional variables. The PLS projection explained in terms of scores and loadings corresponds to Equations (5) and (6).

$$X = \sum_{h=1}^{m} t_h p_h^T + E = TP^T + E. \quad (5)$$

$$Y = \sum_{h=1}^{m} u_h q_h^T + F = UQ^T + F. \quad (6)$$

These equations are called PLS outer relation, where T and U are input and target scores, P and Q are input and target loadings, and E and F are input and target residuals. The relation between score vectors t_h and u_h is called inner relation.

Some previous work combined PLS with fuzzy systems in what is called Fuzzy Partial Least Squares (FPLS) (Bang, Yoo, & Lee, 2002), a subset of Nonlinear Partial Least Squares (NPLS) techniques. This FPLS approach takes the PLS outer relation as a reduction tool to remove collinearity and then applies a Takagi fuzzy model to capture and model the nonlinearity in the projected latent space. To the best of our knowledge, FPLS has not been used in fault detection, though there is published literature on PLS for process monitoring describing its theoretical properties (Li, Qin, & Zhou, 2010), on PLS and its variants with practical applications in fault detection (Muradore & Fiorini, 2012), Wang, Kruger, and Lennox (2003), and on fuzzy systems successfully applied in process monitoring tasks (Angelov, Giglio, Guardiola, Lughofer, & Luján, 2006). Our results demonstrate that it is also feasible to use the combination of PLS+TSK (i.e., FPLS) in the fault detection domain. As described in Section 2.3, we used a TSK model trained with an extended version of *SparseFIS*. Analogously, the analytical model of FLSR results from substituting the inputs in (4) by those resulting from the PLS projection (5).

2.5. Dynamic Residual Analysis (On-line)

The online fault detection uses a dynamic analysis of the *residuals* of the system. Figure 2 shows how the residual signal may be affected by a fault in a monitored channel. The (real) example illustrates how the dynamic band *mimics* the fault when the fault is too small and how the tolerance band is exceeded and a fault is signaled when the fault intensity is sufficiently great to affect the residual signal. This strategy provides a flexible way of handling the residuals, reacting more dynamically to their basic trend.

In order to formulate the tolerance band for each new incoming data sample $\vec{x}(k)$ at time instance k, its affected residuals $res_1(k),...,res_{m^*}(k)$ to all m^* models $f_1,...,f_{m^*}$ are calculated and normalized according to the models' confidence

$$res_m(k) = \frac{|f_m(\vec{x}(k)) - y_m(k)|}{conf_m(k)} \tag{7}$$

where f_m is the model prediction, y_m is the observed target and $conf_m$ is a *local error bar* capturing different certainty levels (confidence) for different regions of the input space according to distribution, noise level, density, etc. of our training data. The inverse covariance matrix of the model inputs X^TX provides a reliable representation of dense and sparse data regions (Nelles, 2001), and by multiplying it with the noise level σ^2, the noise intensity is incorporated

$$cov\{par_m\} = \sigma^2(X^TX)^{-1}\,. \tag{8}$$

Since the noise level of the training data is usually unknown, we use an unbiased estimator (Söderström & Stoica, 1988)

$$\widehat{\sigma}^2 = \frac{2\sum_{k=1}^{N}(y(k)-\widehat{y}(k))^2}{N-deg}, \tag{9}$$

where N is the number of data samples in the training data and deg is the degrees of freedom. For new test data samples X_{test}, the covariance matrix of the model outputs $\widehat{y}_{test}$ becomes

$$cov\{\widehat{y}_{test}\} = \vec{x}_{test}\widehat{\sigma}^2(X^TX)^{-1}\vec{x}_{test}^T = conf_m\,, \tag{10}$$

and is equivalent to the confidence. Thus, Equation (10) express the local error bars in the case of Linear Regression.

As long as the linear parameters of the consequent functions in the Takagi-Sugeno fuzzy systems are estimated by a local learning approach, the formula can also be exploited for its partial local linear models. Then, a nestling of the consequent hyper-planes to the models surface can be observed as analyzed in Angelov, Lughofer, and Zhou (2008). This yields a good approximation of the global model consisting of local linear pieces, which can be used to calculate error bars for each rule consequent function separately and to connect them with weights, thus forming an overall error bar for the whole fuzzy model. The error bar of an evolving (TS) fuzzy model with C rules, for a specific sample $\vec{x}_{act}$ can be calculated by

$$\widehat{y}_m \pm \sqrt{cov\{\widehat{y}_m\}} = \widehat{y}_m \pm \frac{\sum_{i=1}^{C}\varphi_i(\vec{x}_{act})\sqrt{cov\{\widehat{y}_{m,i}\}}}{\sum_{i=1}^{C}\varphi_i(\vec{x}_{act})}, \tag{11}$$

where $\widehat{y}_i$ is the estimated value of the i^{th} rule consequent function, for which $cov\{\widehat{y}_{m,i}\}$ is calculated as in (10) by using the inverse weighted matrix $(X_i^TQ_iX_i)^{-1}$ corresponding to the i^{th} rule and the noise variance as in (9). Hence,

$$cov\{par_{m,i}\} = \widehat{\sigma}^2(X_i^TQ_iX_i)^{-1}\,. \tag{12}$$

The symbol $\varphi_i(\vec{x}_{act})$ denotes the membership degree of the current point to the i^{th} rule and $\widehat{y}_m$ the output value from the TS fuzzy model for the current input point $\vec{x}_{act}$.

The tolerance band equation is completed with the equations of the incremental/decremental tracking along the timeline, where the mean (13) and standard deviation (14) of the residuals are computed.

$$\mu_i(k) = \frac{N_1\mu_i(k-1) + res_i(k) - res_i(k-T)}{N_2} \tag{13}$$

$$\begin{aligned}\sigma_i(k) =& \frac{1}{N_2}(N_1\sigma_i(N-1) + N_2\Delta\mu_i(k)^2 + \\ &(\mu_i(k) - res_i(k))^2 - N_2\Delta\mu_i(k-T)^2 - \\ &(\mu_i(k-T) - res_i(k-T))^2)\end{aligned} \tag{14}$$

where $res_i(k-T) = 0, \mu_i(k-T) = 0, N_1 = k-1, N_2 = k$ for all $k < T$ and $N_1 = N_2 = T$ for $k \geq T$, and $\Delta\mu_i(k) = \mu_i(k) - \mu_i(k-1)$. Combining (13) and (14) yields the tolerance band around the residuals of a given model

$$tolband_i(k) = \mu_i(k) + n * \sigma_i(k)\,, \tag{15}$$

where n is the parameter in the fault detection method and produces the ROC curve when varied.

Finally, our FD system is composed of M identified models, thus producing M residuals signals that are evaluated in parallel. The OR condition is used to identify faults: if a tolerance band in (15) is exceeded by at least one model, a fault alarm is triggered.

3. EXPERIMENTAL SETUP

We have not real faults in our datasets, so we were required to introduce them artificially. This was done by an automatic process, selecting 10 channels at random and introducing a fault in each one. The fault parameters, i.e. type (abrupt of incipient), shape (only for incipient faults) and position were also chosen randomly. The process intentionally skips introducing faults in the warm-up of the detection system, which is an initial stage (20 samples) where faults are neither assumed nor signaled. It also skips faults in regions where the nominal signal is constantly zero. Keeping the random-chosen-parameters of a fault, it is replicated with 5 different intensities (5%, 10%, 20%, 50%, 100%). Since each dataset is composed by 5 different files to test, this led to 50 faults to be detected in each tested dataset at each fault intensity.

This automatic process produces the test files of a run. A run therefore is the execution of a method over these test files, one time per fault intensity.

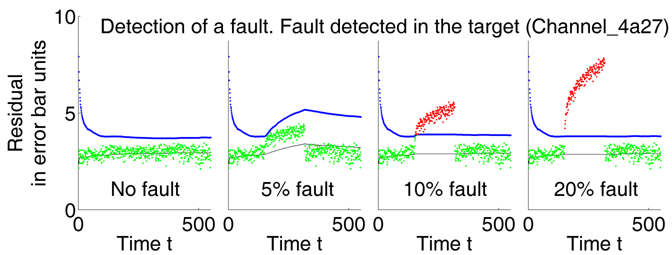


Figure 2. Detection of a fault (drift with exponential shape) in the residual space using a tolerance band. Different fault intensities are shown. The upper line (blue) is a dynamic threshold, whereas the lower line (green when below the threshold, red otherwise) are the residuals. The thin line (black) belongs to the trend of the residuals, i.e. its mean.

The datasets were composed by 240 channels, that after the cleaning process were cut down to 51 and 45 for Dataset 1 and Dataset 2 respectively. There were 9676 samples to train and 8959 samples to test in Dataset 1, whereas Dataset 2 had 11462 samples to train and 12716 samples to test.

4. RESULTS

We base our results on the average of 10 runs, leading to 500 candidate faults to be detected in each tested dataset at each fault intensity. The different fault intensities determine the minimum deviation from the nominal state that a method can detect. Since the results are expressed in terms of ROC curves, the outputs of the experiments are comparable. The sensitivity parameter is the width n in the $n*\sigma$ tolerance band (15); it was increased by steps of 1, from 1 to 10. As we were especially interested in low false alarm rates ($< 10\%$), the X axis of the ROC curves (Figure 3) ranges from 0 to 10 (%). The Y axis denotes the detection rate, and the larger the Area Under the Curve (AUC), the better the method.

Figure 3, up, left, shows how Linear Regression using the spanned datasets (thus obtaining VARMA models) and transforming the dataset using PCA improves ROC curves compared to those of purely linear methods. When the dataset is not spanned but only transformed, the ROC curve produced by the test is even better (recall that *better* means that the Area Under the Curve (AUC) is greater). In this case, both the expansion of the dataset using lags and the orthogonal transformation help to improve the fault detection capabilities, but when combined, the result is not as good as when training the method only in the transformed (i.e., PCA) space. This result seems to be consistent with the analogous case using SparseFIS instead of Linear Regression. Figure 3, up, right, illustrates a behaviour similar to that described previously. SparseFIS produces an improved ROC curve when the method is trained (and tested) on the spanned and transformed dataset, that is, using VARMA models and PCA. When the method is trained with the transformed dataset without lags (i.e. without VARMA models), the ROC is better for one dataset but worse for the other. Therefore, PCA as a preprocessing stage helps to improve the fault detection capabilities of our approach either with Linear Regression or with SparseFIS, but to spanning the dataset with lags seems to yield no further improvement.

The results are different when PLS is used. Figure 3, down, left, shows how Linear Regression trained in a transformed space using PLS without spanning the datasets (i.e. without VARMA models) produces better ROC curves than the linear method itself. With the PLS transformation, the lags also play a role. When the Linear Regression is trained in both transformed and spanned space, the VARMA models yield a greater AUC, as seen in the ROC curves. In Figure 3, down, right, where Linear Regression is substituted by SparseFIS, the result is consistent: the method trained (and tested) with VARMA models in the transformed space using PLS produces better ROC curves than the model alone. Then, the use of VARMA models on the transformed space still produce a gain in the ROC curves. This is particularly noticeable for 'Dataset 1', where the use of lags clearly increases the AUC.

Tables **??** and **??** show the detection rates for each method when establishing thresholds for the false alarm rates in 3%, 5% and 10%, The highest rate per column is highlighted in bold font and 'n/a' denotes that the method is not able to achieve a false alarm rate of $X\%$ or lower.

As can be seen, the methods trained in the transformed space yield better results than methods trained in the original space. As for transformation, PLS is clearly preferable to PCA; although PCA produces better results than methods using the original space, it is outperformed by PLS. VARMA models and PLS appears to be a good combination, since the winning approaches use a transformed space and a VARMA model. The results do not indicate which model is preferable, since Linear Regression and SparseFIS are winners the same number of times, i.e. 9 out of 18. Apparently, SparseFIS better detects low-intensity faults, whereas Linear Regression is best suited for high-intensity ones.

5. Conclusions and Future Works

We have introduced vector autoregressive moving-average models in combination with multivariate orthogonal space transformations to the fault detection domain. Using ROC curves, we have demonstrated that, in terms of detection rates, our approach outperforms approaches in which either the datasets are not spanned with lags (time delays) or orthogonal transformations are not performed before the modeling stage. When PCA transformation is used, the expansion of the datasets with the lags produces no clear improvement in fault detection capability, and the VARMA models can be ignored in this case. In contrast, when the datasets are transformed using PLS, VARMA models help to improve the ROC curves, and therefore their fault detection capabilities.

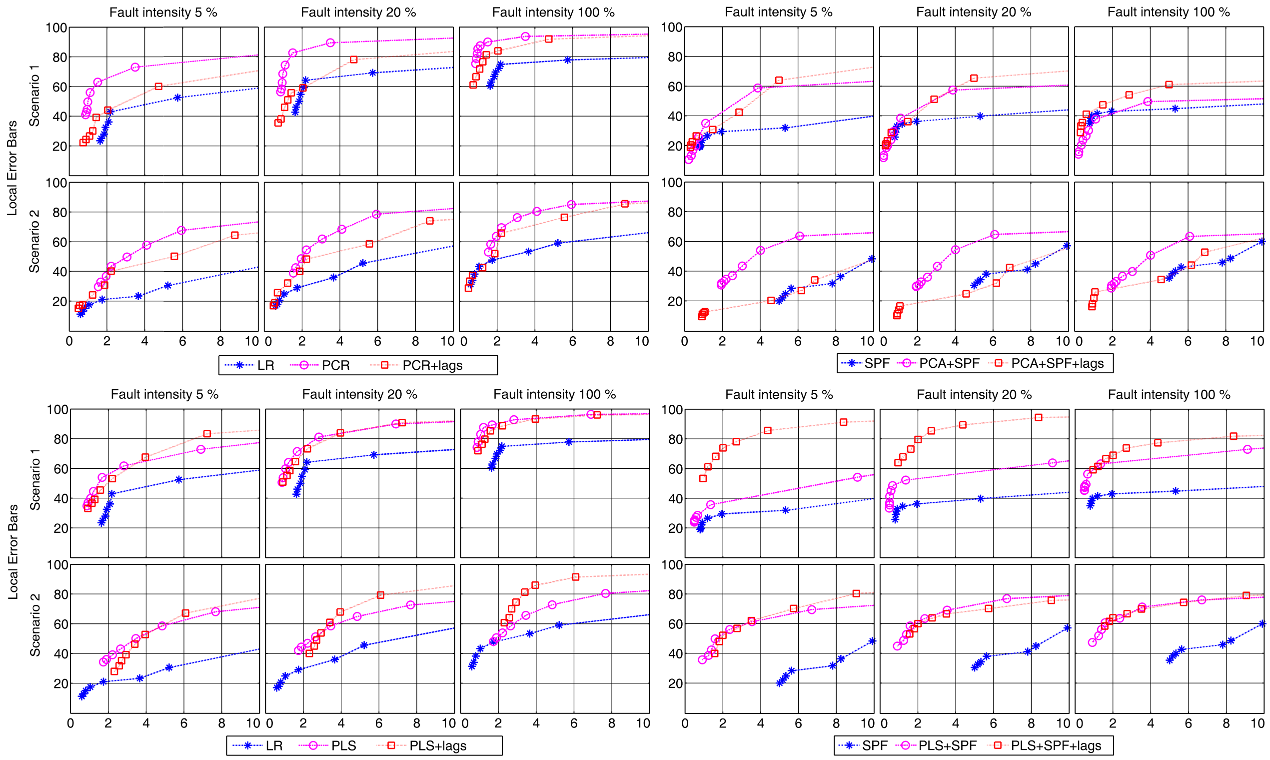


Figure 3. ROC curve (% detection (Y) vs % false alarm (X)) improved. Initial (worst) curves belong to Linear Regression (LR); curves are improved when using PCR with lags; best curves use PCR without lags.

Table 1. Dataset 1. Highest detection rates with different fault and false alarm levels

Method	5% Fault			20% Fault			100% Fault		
	Max. False Alarms			Max. False Alarms			Max. False Alarms		
	< 3%	< 5%	< 10%	< 3%	< 5%	< 10%	< 3%	< 5%	< 10%
Linear Reg.	42.95	42.95	52.53	64.24	64.24	69.24	74.89	74.89	77.86
PCR	63.05	73.15	73.15	82.70	89.40	89.40	89.95	93.72	93.72
PCR + lags	47.10	64.10	64.10	63.39	79.95	79.95	86.41	94.40	94.40
PLS	61.77	61.77	72.91	81.22	81.22	89.98	92.84	92.84	96.43
PLS + lags	73.54	81.67	88.79	85.36	89.16	93.15	**93.09**	**94.99**	**96.86**
SparseFIS	34.45	34.45	39.87	45.08	45.08	49.65	51.59	51.59	53.67
PCA + SparseFIS	35.10	58.78	58.78	38.56	57.38	57.38	37.91	49.65	49.65
PCA + SparseFIS + lags	42.50	64.04	64.04	51.16	65.39	65.39	54.17	61.06	61.06
PLS + SparseFIS	35.78	35.78	54.26	52.37	52.37	63.92	63.24	63.24	73.05
PLS + SparseFIS + lags	**78.11**	**85.65**	**91.30**	**85.44**	**89.42**	**94.41**	73.81	77.33	81.86

Table 2. Dataset 2. Highest detection rates with different fault and false alarm levels

Method	5% Fault			20% Fault			100% Fault		
	Max. False Alarms			Max. False Alarms			Max. False Alarms		
	< 3%	< 5%	< 10%	< 3%	< 5%	< 10%	< 3%	< 5%	< 10%
Linear Reg.	21.08	23.36	30.57	29.06	35.97	45.63	47.72	53.40	59.07
PCR	43.49	57.78	67.69	54.57	68.43	78.58	69.49	80.46	85.11
PCR + lags	40.13	40.13	50.21	48.31	48.31	58.59	65.76	65.76	76.46
PLS	43.15	58.62	68.09	51.28	64.90	72.69	58.54	72.84	80.44
PLS + lags	39.13	52.79	67.20	53.78	67.87	**79.24**	**74.45**	**85.93**	**91.42**
SparseFIS	n/a	14.51	29.68	n/a	20.43	36.55	n/a	32.79	47.88
PCA + SparseFIS	36.96	54.00	63.72	35.96	54.48	64.75	36.66	50.76	63.42
PCA + SparseFIS + lags	12.63	20.30	34.04	16.60	24.77	42.49	26.04	34.40	52.77
PLS + SparseFIS	56.08	61.34	69.55	63.49	**69.15**	76.96	63.65	71.26	76.13
PLS + SparseFIS + lags	**56.76**	**62.05**	**80.37**	**63.92**	66.68	75.75	66.76	69.92	79.05

Testing our approach with more datasets and performing statistical tests on the results will be part of future work. Future work will also concentrate on Fault Identification (FI). In FI, the Fault Detection system goes further, providing additional indications, with confidence measurements, of signaled faults and their locations; this should equip the operator with more information for the decision making process. How a fault deforms a model is a potential starting point for future research.

ACKNOWLEDGMENT

This work formed part of the research program at the Austrian Center of Competence in Mechatronics (ACCM), a part of the COMET K2 program of the Austrian Government. The work-related projects are kindly supported by the Austrian Government and Johannes Kepler University Linz. This publication reflects only the authors' views.

REFERENCES

Angelov, P., Giglio, V., Guardiola, C., Lughofer, E., & Luján, J. (2006). An approach to model-based fault detection in industrial measurement systems with application to engine test benches. *Measurement Science and Technology*, *17*(7), 1809–1818.

Angelov, P., Lughofer, E., & Zhou, X. (2008). Evolving fuzzy classifiers using different model architectures. *Fuzzy Sets and Systems*, *159*(23), 3160–3182.

Bang, Y., Yoo, C., & Lee, I. (2002). Nonlinear pls modeling with fuzzy inference system. *Chemometrics and Intelligent Laboratory Systems*, *64*, 137–155.

Bishop, C. (2006). *Pattern recognition and machine learning*. New York, NY, USA: Springer.

Bolt, P., Batazzi, D., Belfiore, N., Gaspard, C., Goiset, L., Laugier, M., et al. (2010). Damage resistance and roughness retention of work rolls in cold rolling mills. *Revue de Métallurgie*, *107*(6), 245–255.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, *41*(3).

Daubechies, I., Defrise, M., & Mol, C. D. (2004). An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, *57*(11), 1413–1457.

Dong, M., Liu, C., & Li, G. (2010). Robust fault diagnosis based on nonlinear model of hydraulic gauge control system on rolling mill. *IEEE Transactions on Control Systems Technology*, *18*(2), 510–515.

Frank, P., Alcorta, E., & Köppen-Seliger, B. (2000). Modelling for fault detection and isolation versus modelling for control. *Mathematics and Computers in Simulation*, *53*(33), 259 - 271.

Holan, S., Lund, R., & Davis, G. (2010). The arma alphabet soup: A tour of arma model variants. *Statistical Surveys*, *4*, 232–274.

Isermann, R., & Ballé, P. (1997). Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, *5*(5), 709–719.

Jackson, J., & Mudholkar, G. (1979). Control procedures for residuals associated with principal component analysis. *Technometrics*, *21*(3), 341–349.

Jolliffe, I. (2002). *Principal component analysis* (2nd ed.). New York, NY, USA: Springer.

Li, G., Qin, S., & Zhou, D. (2010). Geometric properties of partial least squares for process monitoring. *Automatica*, 204–210.

Lughofer, E., & Kindermann, S. (2010). SparseFIS: Data-driven learning of fuzzy systems with sparsity constraints. *IEEE Transactions on Fuzzy Systems*, *18*(2), 396–411.

Miller, A. (2002). *Subset selection in regression second edition*. Boca Raton, FL, USA: Chapman and Hall/CRC.

Muradore, R., & Fiorini, P. (2012). A pls-based statistical approach for fault detection and isolation of robotic manipulators. *Industrial Electronics, IEEE Transactions on*, *59*, 3167–3175.

Nelles, O. (2001). *Nonlinear system identification*. Berlin: Springer.

Odgaard, P., Lin, B., & Jørgensen, S. (2008). Observer and data-driven-model-based fault detection in power plant coal mills. *IEEE Transactions on Energy Conversion*, *23*(2), 659–668.

Pichler, K., Lughofer, E., Buchegger, T., Klement, E., & Huschenbett, M. (2012). A visual method to detect broken reciprocating compressor valves under varying load conditions. In *Proceedings of the asme 2012 international mechanical engineering congress & exposition* (p. to appear). Houston, TX, USA: ASME.

Schöener, H., Moser, B., & Lughofer, E. (2008). On preprocessing multi-channel sensor data for online process monitoring. In *Proceedings of the international conference on computational intelligence for modelling, control and automation (cimca)* (pp. 414–419). Vienna, Austria.

Söderström, T., & Stoica, P. (1988). *System identification*. Upper Saddle River, NJ, USA: Prentice Hall.

Tamura, M., & Tsujita, S. (2007). A study on the number of principal components and sensitivity of fault detection using pca. *Computers & Chemical Engineering*, *31*(9), 1035–1046.

Theilliol, D., Mahfouf, M., Ponsart, J., Sauter, D., & Gama, M. (2010). Design of a fault diagnosis system based on a bank of filter-observers with application to a hot rolling mill. *Transactions of the Institute of Measurement and Control*, *32*(3), 265–285.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S., & Yin, K. (2003). A review of process fault detection and diagnosis: Part iii: Process history based methods. *Com-*

puters & Chemical Engineering*, *27*(3), 327–346.

Wang, X., Kruger, U., & Lennox, B. (2003). Recursive partial least squares algorithms for monitoring complex industrial processes. *Control Engineering Practice*, *6*(11), 613–632.

Yang, M., & Makis, V. (2010). ARX model-based gearbox fault detection and localization under varying load conditions. *Journal of Sound and Vibration*, *329*(24), 5209–5221.

Yang, T. (2006). A method of fast fault detection based on ARMA and neural network. In *Proceedings of the sixth world congress on intelligent control and automation (wcica)* (pp. 5438–5441). Dalian, China.

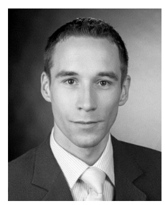
BIOGRAPHIES

Francisco Serdio was born in Asturias, Spain, in 1979. He received the Dipl.-Ing. degree in Computer Science and the MSc. degree in Soft Computing and Intelligent Data Analysis from the University of Oviedo, Asturias, Spain, in 2002 and 2011, respectively. He was involved in commercial software development companies from 2003 to 2010, including banking, stock market, CRM, and others. He is currently pursuing the Ph.D. degree with the Department of Knowledge-Based Mathematical Systems at the Johannes Kepler University, Linz. He received best paper awards at the IFAC Conference on Manufacturing Modelling, Management and Control, 2013. His research interests mainly include data-driven modeling and fault detection.

Edwin Lughofer was born in Ried im Innkreis, Austria, in 1972. He received his PhD. degree from the Department of Knowledge-Based Mathematical Systems at the Johannes Kepler University, Linz, where he is now employed as post-doctoral fellow. He has published more than 90 journal and conference papers in the fields of evolving fuzzy systems, machine learning and vision, data-driven modeling, clustering, fault detection, quality control and human-machine interaction, including a monograph on 'Evolving Fuzzy Systems' (Springer, 2011) and an edited book on 'Learning in Non-stationary Environments' (Springer, 2012). He received best paper awards at the 2006 International Symposium on Evolving Fuzzy Systems, at the 2008 International Workshop on Genetic and Evolving Fuzzy Systems and at the IFAC Conference on Manufacturing Modeling, Management and Control 2013, as well as the Royal Society Grant for knowledge exchange with Lancaster University in 2007.

Kurt Pichler was born in Grieskirchen, Austria, in 1981. He received the Dipl.-Ing. degree in technical mathematics from the Johannes Kepler University Linz (JKU), Austria, in 2005. He was with the Institute for Design and Control of Mechatronical Systems at the JKU and joined the Linz Center of Mechatronics (LCM) in 2007. He is currently pursuing the Ph.D. degree with the Department of Knowledge-Based Mathematical Systems at the JKU. His research interests include fault detection, parameter estimation, data-driven modeling and pattern recognition.

Thomas Buchegger was born in Linz, Austria, in 1974. He received the Dipl.-Ing. degree in mechatronical engineering in 2002 and his Dr. techn. (Ph.D.) degree in 2005 from the Johannes Kepler University, Linz, Austria. He is employed at the Linz Center of Mechatronics GmbH as Business Unit Manager for Sensors and Communication and Information Analysis. In the Comet K2 Center Austrian Center of Competence in Mechatronics GmbH he works as Vice Area Coordinator for Wireless Systems.



Markus Pichler was born in Linz, Austria, in 1976. He received the Dipl.-Ing. degree in mechatronics and the Dr.techn. degree from the Johannes Kepler University, Linz, Austria, in 2002 and 2007, respectively. He was with the Institute for Communications and Information Engineering at the University of Linz, and joined the Linz Center of Mechatronics (LCM) in 2002. His research interests include systems design, signal generation, digital signal processing and parameter estimation for radar and positioning systems as well as fault detection and error analysis. He has authored or co-authored over 25 publications, was the recipient of the 2004 European Microwave Association (EuMA) Radar Prize, and is a member of IEEE.

Hajrudin Efendic was born in Gorazde, Bosnia and Herzegovina, in 1970. He received the Dipl.-Ing. degree in automatic control and electronics and the MSc. degree in telecommunications from the University of Sarajevo, Bosnia and Herzegovina, in 1997 and 2001 respectively, and the Ph.D. degree in technical sciences from Johannes Kepler University (JKU), Linz, Austria, in 2012. He also received the MBA degree (summa cum laude) with specializations in management, international management and quantitative analysis from the University of Cincinnati, OH, U.S., in 2011. He is with the Institute for Design and Control of Mechatronical Systems at the JKU. His research is focused on data-based structural analysis, modeling and data-based modeling, monitoring and fault diagnosis in complex technical systems and in physiological and medical systems.