2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Gradient-based Fuzzy Fault Isolation in Residual-based Fault Detection Systems

Francisco Serdio, Edwin Lughofer, Kurt Pichler, Thomas Buchegger, Markus Pichler, Hajrudin Efendic

***Abstract*— We introduce a fault isolation technique based on the analysis of the deformation of data-driven models produced by an incoming fault. Combining the gradients within a model, with the confidence of the model in terms of its quality influenced by the degree of violation of the uncertainty measure used in the fault detection phase allows us to successfully identify faults from the fault alarms produced by a residual-based fault-detection system relying on data-driven models. These models are built from scratch fully automatically on the basis of measurements recorded online and collected off-line in a preliminary batch phase (no physical or expert knowledge required). We used Partial Least Squares (PLS) regression and fuzzy modeling techniques with the inclusion of time lags in the input variables to establish time-varying prediction models. The deformation analysis is performed throughout the warning-models (those signaling the presence of a fault), and combines the contributions of all channels to the model prediction and then proposes a candidate faulty channel. We also introduce the concept of a Fault Isolation Likelihood Curve (FILC), inspired by the well-known Receiver Operating Characteristic (ROC) curves, in order to (i) show the isolation rates in a convenient and interpretable way and (ii) allow comparison between the detection and isolation capabilities of a fault detection system. In tandem with the FILC, we introduce the concept of the Fault Isolation Gap (FIG) as a tool for measuring the isolation capabilities of an algorithm with regards to the (fault) detection capabilities achieved by a fault detection method.**

I. INTRODUCTION

Fault Detection (FD) is of great importance as proper and accurate condition monitoring (i.e., early detection of faults) in industrial systems minimizes costs for repairs and increases production efficiency. Fault Isolation (FI) in particular is also attracting attention since once a fault is detected, it allows system operators to identify the channel or combination of channels responsible for causing the abnormality in the process and thus to determine which part should be maintained or replaced (localization).

Previous works relied on, once a fault was detected, examining the contribution of the original channels to the observable characteristic of the system that exceeded a control limit. This means that FI can be carried out by reverting to the original process variables. This was the approach followed in [1], where the authors implemented a process variable contribution plot for linear Principal Component Analysis (PCA) describing the change in the new observation variables relative to average values calculated from the nominal model. This work was successfully extended to the non-linear PCA case in [2]. However, PCA-based fault detection performed poorly for particular measurement signals, for instance untypical occurrences such as abrupt pattern changes; for details refer to [3]. Other FI approaches rely on analytical fault and isolation models [4] [5] [6], which are usually very system-specific (applicable only to one system type) and require significant manpower (physicians, experts) during setup.

In this paper, we investigate a new kind of FI algorithm which is based on on-line residuals extracted from System Identification (SysID) models within a FD framework as demonstrated in [3]. It can deal with arbitrary types of measurement signal, avoids use of fault patterns and types, acts in a fully automatic, unsupervised manner, and reduces operator effort significantly. The FI component serves as an add-on in this framework and relies on time-lagged prediction models obtained by data-driven regression techniques (i.e. PLS and fuzzy modeling, the latter for modeling non-linear relations). These may become violated in on-line mode, which means they may show untypical residuals and point to potential fault candidates. All violated models are used in the FI process by examining the degrees of contributions of all the measurement variables/channels included in these models. The contributions are calculated for each model separately by a weighted combination of the various models' gradients (along each input variable) for the current sample, model quality (in terms of expected prediction error) and the degree of model violation (distance to tolerance band), and are accumulated over all the violated models. The variables with the highest contributions are expected to be those most affected by system failures.

The following Section II explains how the FD is built from scratch and describes the models involved. Section III explains the FI approach, including how the contribution of variables is computed and introducing how these contributions are aggregated to make a decision concerning FI with two variants: crisp (isolated channel) and fuzzy (relative contributions over all channels). Section IV shows FI results obtained for a real-world data set from engine test benches. To this end, the concepts of the FILC and the FIG are introduced as evaluation measures. Finally, Section V concludes the paper and presents future work.

Francisco Serdio and Edwin Lughofer are with the Department of Knowledge-Based Mathematical Systems, Johannes Kepler University Linz, Austria (email: {francisco.serdio, edwin.lughofer}@jku.at).

Kurt Pichler, Thomas Buchegger and Markus Pichler are with the Linz Center of Mechatronics (LCM), Linz, Austria (email: {kurt.pichler, thomas.buchegger}@jku.at, {markus.pichler}@accm.co.at).

Hajrudin Efendic is with the Institute for Design and Control of Mechatronical Systems, Johannes Kepler University Linz, Austria (email: {hajrudin.efendic}@jku.at).

II. Fault Detection

A. The system

We used a residual-based FD system as developed in [3] using standard soft computing models, extended in [7] with variable projections (transformations) and time-lagged variables, inducing non-linear Vector Auto-Regressive Moving Average (VARMA) and non-linear Box-Jenkins-type models [8] (observing the dynamics required in the SysID step). Further details on PLS and fuzzy modeling using Sparse Fuzzy Inference System (SparseFIS) can be found in [9] and [10] respectively. Our FD system is based on the online tracking of residuals extracted from the models for new instances. There are several models for different monitoring channels (i.e. for those that can be approximated by a subset of other channels with sufficient accuracy), which are suitable for providing a measure of the model's uncertainty in the prediction of the target given a concrete input sample. We show this uncertainty measure as an error bar and use it later to normalize the residuals, that is, the differences between predicted and observed targets, which are tracked over time. A tolerance band is created on the basis of the residuals' online tracking to handle the mean and standard deviation for the normalized residuals, thus producing a dynamic threshold which, when exceeded, raises a fault alarm. "Exceeding" means in this context that they show an untypical behavior in the time signal. Figure 1 presents an overall-overview of the workflow with the FD system in the upper part. Further explanations and more detail about the FD system can be found in [3]. Below we provide a brief summary of the uncertainty measure and the online tracking of the residuals used, as these serve as one input to our FI approach.

B. Uncertainty measure

A local error bar serves as uncertainty measure and takes into account the local data distributions, which may change over the whole input feature space. Combining the inverse covariance matrix of the model parameters $X^T X$, which is a reliable representation of dense and sparse data regions [11], with the noise level σ^2 to integrate the noise intensity yields a local error bar expressed by the formula:

$$cov\{\hat{y}\} = X_{test}\hat{\sigma}^2(X^T X)^{-1}X_{test}^T \qquad \hat{y} \pm \sqrt{diag(cov\{\hat{y}\})} \quad (1)$$

where X_{test} is the current on-line test sample.

C. Online-tracking of residuals

Before residuals are tracked, they are normalized using the uncertainty of their corresponding model in the current sample region:

$$res_i(k) = \frac{|f_i(\vec{x}(k)) - y_i(k)|}{uncert_i}, \quad (2)$$

where $\vec{x}(k)$ is the current sample, f_i is the evaluation of the model of channel i, y_i is the observation of the same channel, and $uncert_i = cov\{\hat{y}_i(.)\}$ as given in Equation 1. Recall that low confidence levels (high values of $uncert_i$) reduce the residual even when there are clear deviations from the model. The tracking is performed samplewise and handles the mean and standard deviation of the normalized residual in an incremental fashion:

$$tolband_i(k) = \mu_i(k) + n * \sigma_i(k), \quad (3)$$

where μ is the mean, σ is the standard deviation, and n is a multiplier and an essential parameter in our FD method. Note the incremental/decremental σ-update includes rank-1 modification $\Delta\mu_i(k)$ for better stability [12] and that values outside this band are, of course, those seen as fault candidates.

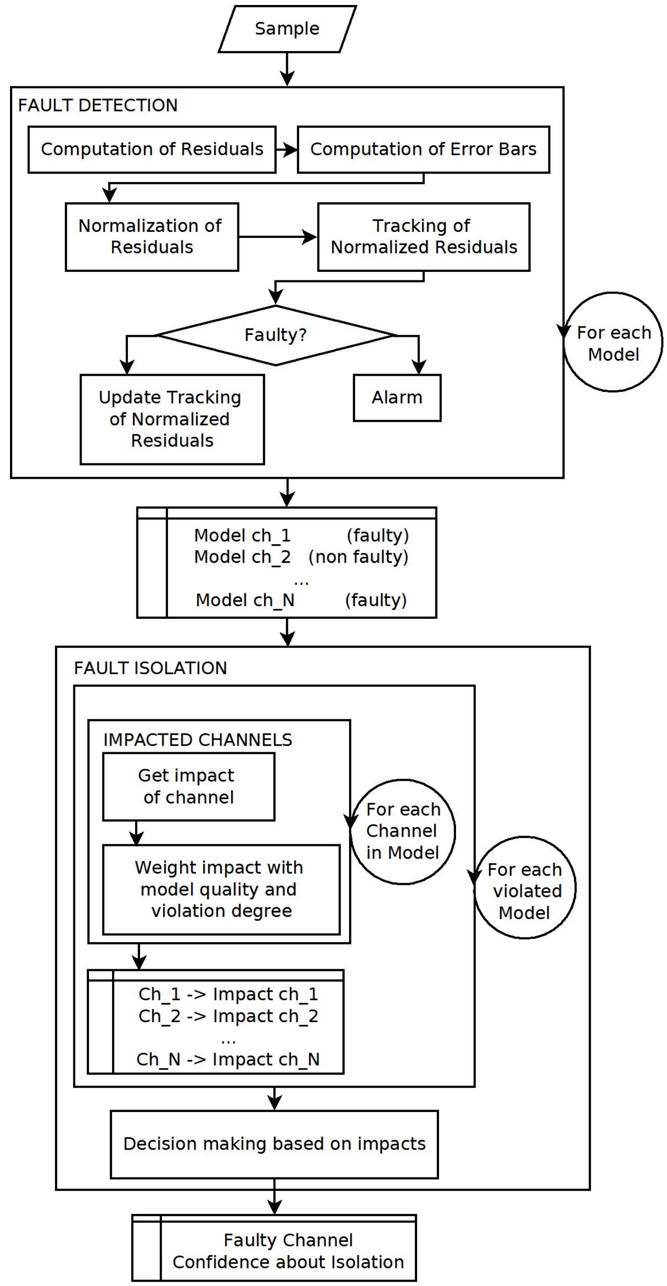

Fig. 1. From detection to isolation: reverting to the original process variables to determine the influence of each

D. Fault detection rates

As mentioned above, previous research focused on FD. Table I summarizes the most meaningful methods, based on a statistical comparison of their performance. Table II shows the detection rates of the best methods for one of the data sets studied (excluding other methods such as Auto-Regressive Moving Average (ARMA) [13], the PCA-based approach [14], and One-Class Support Vector Machines (SVMs) [15], which are non-residual-based techniques and perform much more poorly). Note that only one column has no winning method (5% fault intensity, with false positives below 3%), which confirms that the 4 methods shown were the most significant of the 27 different approaches investigated. Additionally, Table II shows that the PCA-based approach would not have been a reasonable choice for FI, as its FD performance was already very poor.

For further information about PLS and its fuzzy variant, Fuzzy PLS (FPLS), refer to [16] and [17] respectively. For further details on VARMA models (denoted by the keyword "lags" in all the tables), see [7] and [13].

TABLE I
STATISTICAL EVALUATION OF METHODS USING METHOD RANKING, WHERE (+/-/0) MEANS (WINNER/LOSSER/TIE).

Set / Level	PLS	PLS + lags	PLS + SparseFIS	PLS + SparseFIS + lags
PLS	0	-	0	0
PLS + lags	+	0	+	0
PLS + SparseFIS	0	-	0	0
PLS + SparseFIS + lags	0	0	0	0

III. FAULT ISOLATION

Our FI strategy is based on partial derivatives of the SysID models. As previously mentioned, the partial derivatives of a function with respect to a specific dimension can indicate the relative importance of the corresponding variable (channel) for that function. The lower part of Figure 1 schematically shows how all violated models are analyzed at a particular time instant to extract the contribution of each channel involved and how the quality of the model and the degree of violation of the model (deviation of the residual with regards to the threshold) are used to normalize and weight the impact of the channel.

A. Contribution of the variables

When a fault is detected, the FI system comes into play. The FI system is fed with all the violated models (i.e., those whose tolerance band in (3) is exceeded) and the channels involved in them and then analyzes the contribution of each channel to the prediction of the model it belongs to. This process is detailed in Algorithm 1. A unit contribution is assigned to the target channel (the channel with the model), while the contribution of the model's input channels is the result of the partial derivative of the channel multiplied by model quality and degree of violation of the model at instant t. The degree of violation is expressed by the factor by which the residual is greater than the threshold. The partial derivatives are directly obtained from the regression coefficients multiplied by the weights in the principal components in the case of PLS regression and measured by numerical differential quotients in the case of fuzzy models (using the difference between $f(\vec{x})$ and $f(\vec{x})+\Delta$ with Δ a small positive number). The quality of each model is determined during the batch off-line system identification steps in a stratified cross-validation procedure [18] that estimates the expected prediction error on new samples, see [3]. We use as quality the *coefficient of determination* (R2) of the model (function Get_Quality in Algorithm 1). Note that when time lags are used in the models (VARMA case), the contribution of a lagged channel must be added to the contributions of its original (non-lagged) channel, which is the candidate channel for isolation.

B. Decision based on the accumulated contributions

We consider two ways of isolating the fault (i.e., of deciding which are the faulty channels where the fault is mostly likely to happen).

a) Crisp decision: follows a winner-takes-all approach, that considers the faulty channel to be that with the highest accumulated contribution amongst the channels present in the violated models at time t. The decision relies on the idea that the faulty channel usually contributes most to untypical residuals obtained from the models and causes them to exceed the corresponding FD model's dynamic threshold in (3). However if the channels affected by a fault contribute little to the violated models, the residual is unlikely to show any untypical behavior at all and does not violate (3) (thus, this fault cannot be detected at this stage, and FI is not triggered). Since the FI performance therefore depends on the FD performance, it can never be equally high (see the results section below). In fact, one channel may contribute more to

Algorithm 1 Extracts the contributions of the channels in a violated model at a time instant t

$Input\ model:\ the\ violated\ model$
$Input\ inputs\_t:\ the\ inputs\ at\ instant\ t$
$Input\ violation:\ the\ violation\ degree\ of\ the\ model$
$Return:\ contributions\ of\ the\ channels$
function CONTRIBUTIONS($model$, $inputs\_t$, $violation$)
    $contribs \leftarrow empty\_list$
    $contrib\_target \leftarrow 1$
    $\text{ADD\_TO\_LIST}(contribs, contrib\_target)$
    $quality \leftarrow \text{GET\_QUALITY}(model)$
    $input\_channels \leftarrow \text{GET\_INPUT\_CHANNELS}(model)$
    for all channels ch_i in $input\_channels$ **do**
        $partial\_ch_i \leftarrow \frac{\partial i}{\partial t}(model, ch_i, inputs\_t)$
        $contrib\_ch_i \leftarrow partial\_ch_i * quality * violation$
        $\text{ADD\_TO\_LIST}(contribs, contrib\_ch_i)$
    end for
    return $contribs$
end function

TABLE II

ENGINE DATASET. SUMMARY OF DETECTION RATES WITH DIFFERENT FAULT INTENSITIES AND FALSE ALARM LEVELS.

Method	5% Fault			20% Fault			100% Fault		
	Max. Overdetections			Max. Overdetections			Max. Overdetections		
	$< 3\%$	$< 5\%$	$< 10\%$	$< 3\%$	$< 5\%$	$< 10\%$	$< 3\%$	$< 5\%$	$< 10\%$
PLS	**73.00**	73.00	**89.00**	69.00	69.00	85.00	62.00	62.00	78.00
PLS + lags	70.00	78.00	85.00	**83.00**	89.00	**94.00**	**93.00**	**96.00**	**97.00**
PLS + SparseFIS	72.00	72.00	85.00	79.00	79.00	87.00	75.00	75.00	84.00
PLS + SparseFIS + lags	58.00	80.00	80.00	81.00	**90.00**	90.00	**93.00**	95.00	95.00
PCA - CPV 0.90	n/a	n/a	79.00	n/a	n/a	78.00	n/a	n/a	87.00

a particular model than the channel which is really affected by the fault, but this likelihood decreases with the number of models violated because the faulty channel appears as input (exploration through model diversity). Since this strategy is based on a crisp decision, a faulty channel is either (properly) isolated or not, but never partially isolated. Considering degrees of isolation in order to obtain a relative isolation level and to offer more options to the user naturally leads to the second approach.

b) Fuzzy decision: allows a set of channels -each with its corresponding contribution- to be provided to the FD system operator as valuable extra information. This also compensates for cases in the decision-making process where the true faulty channel is not that with the highest contribution amongst the violated models — in this case, the crisp variant will always deliver a false information, whereas the fuzzy variant postpones the decision, shows possible alternatives, and potentially provides a degree of isolation for that faulty channel. In cases where the faulty channel has the highest impact, this fuzzy decision approach will, of course, produce the same results as the crisp decision approach. A remarkable point directly arising by definition is that this fuzzy version will always produce better isolation results since every time a faulty channel is not that with the greatest contribution to the target's prediction of a model, it will contribute some value to the isolation rate due to a partial isolation degree. In particular, in the fuzzy case, we have

$$fuzzycont\_ch_m = \frac{acc(contrib\_ch_m)}{max_{i=1,\ldots,M} acc(contrib\_ch_i)} \qquad (4)$$

for each channel m included in the measurement system. Thus, if the faulty channel has the maximum accumulated contribution over the set of M channels in all violated models, then it will be assigned isolation degree 1; otherwise, if will receive an isolation degree (> 0) proportional to that of the channel with the maximum accumulated contribution. These degrees can be reported to operators (in the form of a list of channels most likely affected by the fault) and are also used when the FI rates are calculated (see below).

Algorithm 2 Extracts the contributions of the channels all along the violated models, at a time instant t

```
Input models : the models
Input inputs_t : the inputs
Return : contributions of channels
function JOIN_CONTRIBUTIONS(models, inputs_t)
    contribs ← empty_list
    for all models m_i in models do
        if m_i is violated then
            v_m_i ← (residual/threshold)
            ctr_i ← CONTRIBUTIONS(m_i, inputs_t, v_m_i)
            ADD_TO_LIST(contribs, ctr_i)
        else
            Skip model
        end if
    end for
    return contribs
end function
```

IV. RESULTS

A. Fault Isolation Likelihood Curves (FILC)

ROC curves were used in previous research (e.g., [3]) to visualize detection rates -we prefer the name Fault Detection Curve (FDC), for use in this domain. These curves, plotting true positives (y-axis) against false positives (x-axis) can also be adapted to display isolation rates in so-called FILC curves: the x-axis again shows the false positive detections, whereas the y-axis will show the true positive isolations. This leads to two direct observations: an FDC will always be above its corresponding FILC, and the x-values of both curves will contain the same information (i.e., the false positives values are the same). This means that a direct measure of the isolation capabilities of our method can be provided in term of what we call FIG, as explained below.

B. Fault Isolation Gap (FIG)

We define the FIG as the difference between an FDC and an FILC for a particular value of x in the combined graph. The concept can, of course, be easily extended to the difference between Areas Under the Curve (AUCs) when whole curves instead of concrete points are considered. However, in both cases this is a measure of what proportion of detected faults could not be isolated. This gap will produce a direct graphical and numerical impression of the ability of a method to isolate faults. This measure should be presented in combination with the number of candidate channels between which to distinguish in an isolation-decision-making process. Tendentially, the higher this number becomes, the more difficult it is to extract the affected faulty channel correctly. This is somewhat related to a multi-class classification problem, where the number of classes represents the number of channels (possible isolation feedbacks) — and it is well-known that the performance usually decreases with an increasing number of classes [18]. Thus, we also report on the average number of channels involved in the violated models to assess the ability to isolate faults: for instance, when 10 different channels are involved in all the violated models for a particular sample, an isolation rate of 50% constitutes acceptable performance, whereas when only 2 channels are involved (one-input-one-output model), this is no better than random guessing.

C. Fault Isolation Ratio (FI Ratio)

Isolation deals with the question of *what can be isolated from what can be detected*, i.e., how well our fault isolation strategy performs once a fault is detected. We therefore define a Fault Isolation Ratio (FI Ratio) as the proportion between isolation and detection

$$FI\ Ratio = FIR \ / \ FDR, \tag{5}$$

where FIR is the Fault Isolation Rate defined as

$$FIR = n^{\underline{o}}\ faults\ isolated \ / \ n^{\underline{o}}\ total\ faults, \tag{6}$$

and FDR is the Fault Detection Rate defined as

$$FDR = n^{\underline{o}}\ faults\ detected \ / \ n^{\underline{o}}\ total\ faults. \tag{7}$$

Note that FI Ratio, FIR and FDR are bounded by 1 (≤ 1) and that $FIR \leq FDR$. Further, note how for fuzzy isolation approaches the FIR becomes a likelihood instead of a proper rate since partial isolation results are taken into account.

D. The results

The results of PLS and its VARMA variant for the engine data set investigated are shown in Figures 2 and 4, respectively. The graphical version compares the FDC and FILC curves (cf. Section IV-A). The FILC is shown in its crisp and fuzzy versions, and -as expected- the latter outperforms the former because the partial identification rates increase the FI rates in a stepwise manner, whereas the crisp version does not produce meaningful results.

The role of the fault intensity. From the figures, it is clear that our fault isolation method is able to isolate more faults when the fault intensity is sufficiently high (20% to 100% intensity) than when it is low (5% to 10% intensity). This confirms the expectation that the faulty channel must influence the models such that it is *visible* among all other channels when reverting from the prediction of the model to the contributions of its channels.

Autoregressive gain. The figures also clearly show that the autoregressive models help the fault isolation process. This was also the case in the fault detection stage, but from the isolation results the gain appears to be even clearer than before. As a good example, high-intensity faults (50% and 100% intensity) produce isolation rates that are around 40% greater than those produced by their non-autoregressive counterparts. However, for low-intensity faults (5% and 10% intensity), the isolation results with time lags included (VARMA models) are slightly worse than those without.

The impact of over-detections. From Table III it can be seen that when the over-detection rates rise, the FI Ratio decreases, which is in line with our expectations of falsely violated models corrupting the isolation process (see Section III). We investigated only small over-detection rates (less than 3%) as required by the system experts to avoid too many distractions. However, we were able to observe a similar trend in cases of 5-20% over-detection rates.

The classification problem. Figures 3 and 5 show how the number of channels involved in the violated models varies for the different values of n multiplying the tracking tolerance band (cf. Section II-C). As expected, when the tolerance band for the residuals is too narrow ($n = 1, n = 2$), the average and standard deviation of the number of channels involved in the violated models is high ($\approx 16.5 \pm 2.5$ channels for non-VARMA models, $\approx 20 \pm 5$ channels for VARMA models), thus making the classification problem harder. For values of $n \geq 3$, the average and standard deviation over the number of channels involved stabilizes at about $[9.75, 11.75] \pm [2.75, 4.25]$ channels, reducing the complexity of the isolation task to (merely) 10 to 15 channels. These results are detailed in Table IV: clearly, for a tolerance band value $n \geq 4$ (the important band range, as with values below 4 the over-detection rates grows drastically [7]), the complexity of the isolation problem remains fairly constant.

V. Conclusions and Future work

We have introduced a residual-based FI system which relies on partial derivatives and comes in two variants (fuzzy and crisp) in order to isolate faults in a purely data-driven black-box Fault Detection and Isolation (FDI) system that uses neither information about the process producing the data nor about the faults to be isolated. We have successfully tested the algorithm with two modeling techniques (selected from previous research where good detection rates were obtained) on a data set from a real-world process. Our results confirm that the technique can be used effectively in real-world applications and that the influence of a fault in the residuals of one or several models can be backtracked simply by investigating the gradients within the models at the time the fault is signaled by the fault detection system. We have also introduced the Fault Isolation Likelihood Curve, the Fault Isolation Gap and the Fault Isolation Ratio as tools for measuring the (fault) isolation capabilities of a concrete method/technique and for comparing the detection and the isolation capabilities within the method/technique itself. Considering the complexity of the isolation problem (equivalent to a multi-class classification problem with around 10 classes), our approach shows solid isolation ratios. Thus, it can also be used in other (residual-based) fault detection frameworks showing a different performance, as the isolation ratios indicate the FI performance independently of the FD performance. Future work will focus on how to improve the technique's isolation capabilities, mainly by modifying

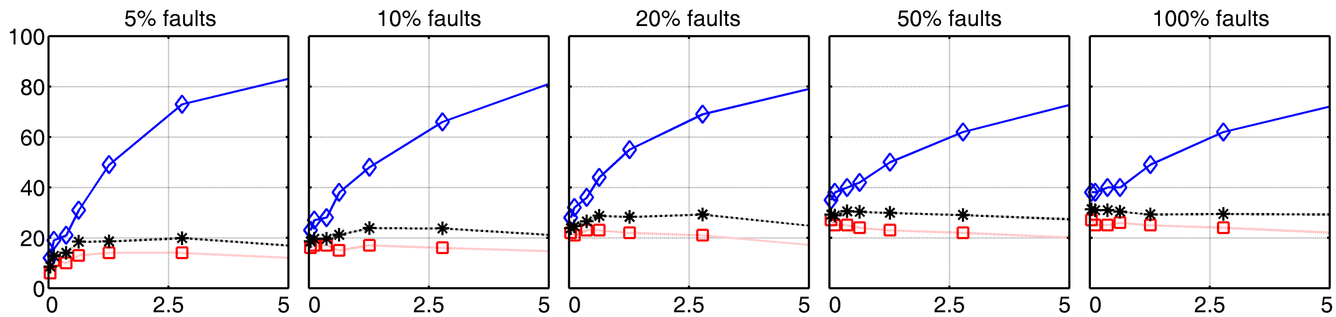

Fig. 2. PLS: FDC vs. Fuzzy FIDL vs. Crisp FIDL. Top curve (blue) is FDC, middle curve (black) is Fuzzy-FILC and lower curve (blue) is Crisp-FILC. The curves plot True Positives (y) against False Positives (x)

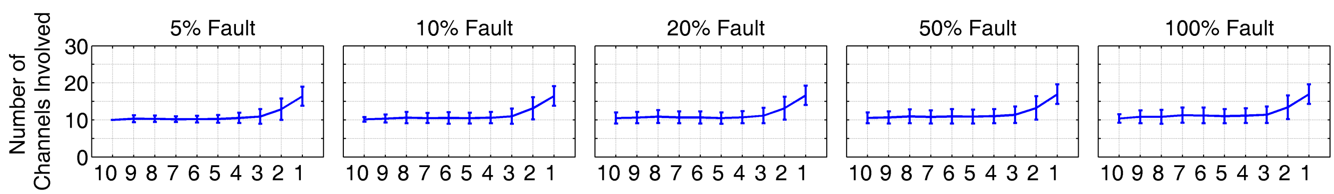

Fig. 3. PLS: Error bar plot ($\mu \pm \sigma$) of the number of channels against the σ values of the tolerance band. Low values of σ imply a greater number of channels to be isolated because of a greater number of violated models, which makes the isolation harder.

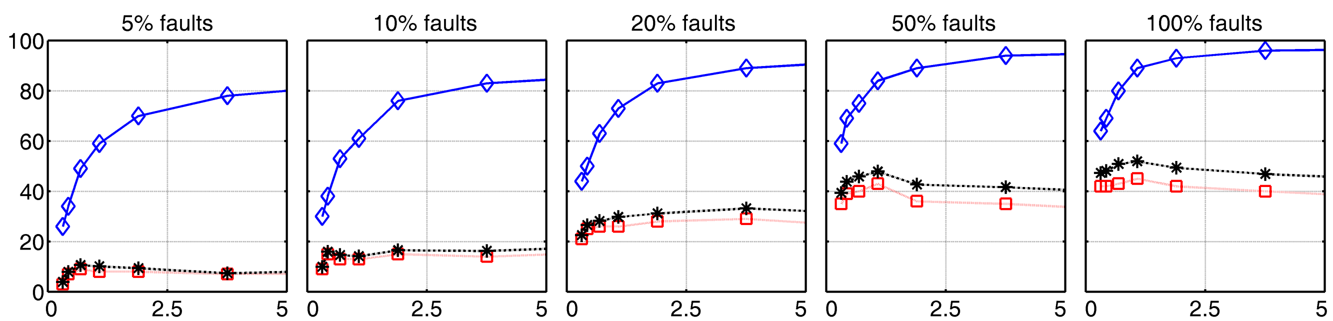

Fig. 4. PLS + Lags: FDC vs. Fuzzy FIDL vs. Crisp FIDL. Top curve (blue) is FDC, middle curve (black) is Fuzzy-FILC and lower curve (blue) is Crisp-FILC. The curves plot True Positives (y) against False Positives (x)

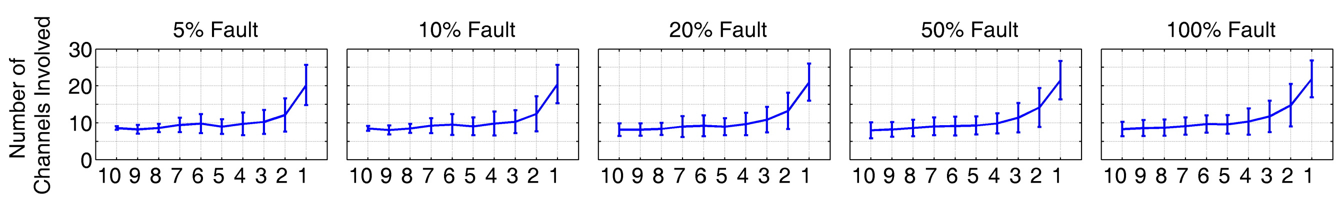

Fig. 5. PLS + Lags: Error bar plot ($\mu \pm \sigma$) of the number of channels against the σ values of the tolerance band. Low values of σ imply a greater number of channels to be isolated because of a greater number of violated models, which makes the isolation harder.

the way in which channel contributions are aggregated (for instance, using time frames) to smooth out the disturbances produced by false positives.

Acknowledgment

This work has been [partially] supported by the Austrian COMET-K2 programme of the Linz Center of Mechatronics (LCM), and was funded by the Austrian federal government and the federal state of Upper Austria. The work-related projects are kindly supported by the Austrian Government and Johannes Kepler University Linz. This publication reflects only the authors' views.

References

[1] P. Miller, R. Swanson, and C. Heckler, "Contribution plots: A missing link in multivariate quality control," *Applied Mathematics and Computer Science*, vol. 8, p. 775792, 1998.

[2] F. Jia, E. Martin, and A. Morris, "Nonlinear principal components analysis with application to process fault detection," *International Journal of Systems Science*, vol. 31, p. 14731487, 2001.

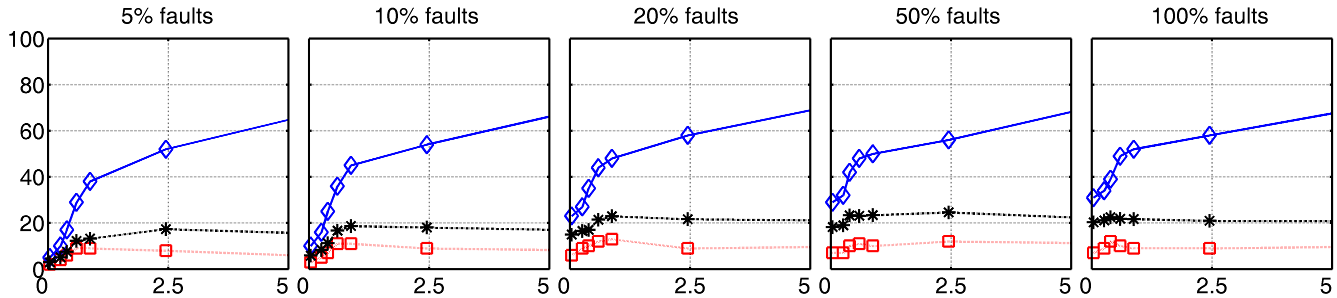

Fig. 6. SparseFIS: FDC vs. Fuzzy FIDL vs. Crisp FIDL. Upper curve -blue- is FDC, where middle curve -red- is Fuzzy-FILC and lower curve -black- is Crisp-FILC. The curve faces True Positives (Y) against False Positives (X)

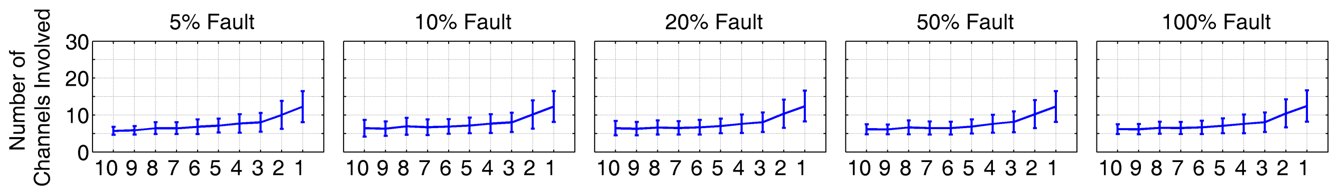

Fig. 7. SparseFIS: Error bar plot ($\mu \pm \sigma$) of the number of channels along the different σ values of the tolerance band. Small values of σ imply greater number of channels to isolate due to more violated models, thus making the isolation harder.

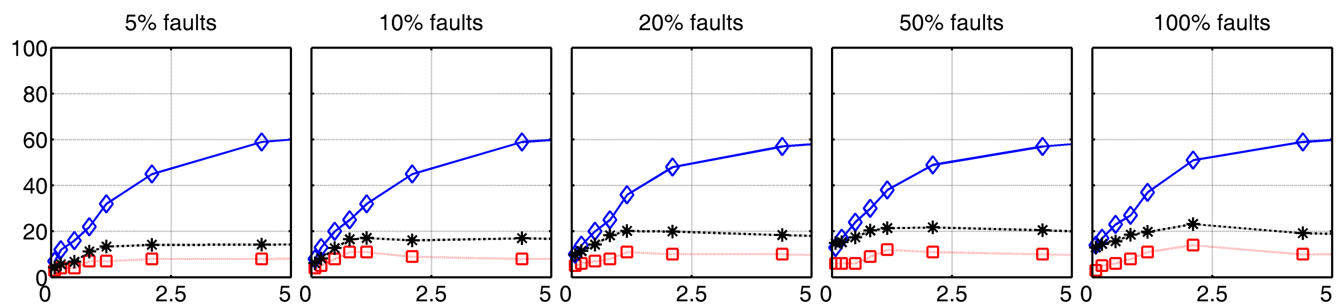

Fig. 8. SparseFIS + Lags: FDC vs. Fuzzy FIDL vs. Crisp FIDL. Upper curve -blue- is FDC, where middle curve -red- is Fuzzy-FILC and lower curve -black- is Crisp-FILC. The curve faces True Positives (Y) against False Positives (X)

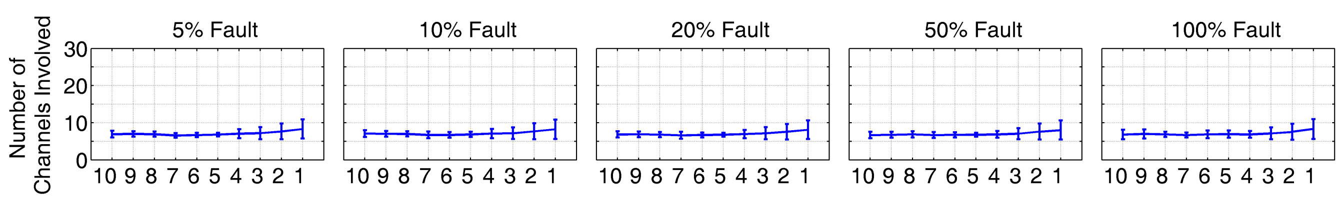

Fig. 9. SparseFIS + Lags: Error bar plot ($\mu \pm \sigma$) of the number of channels along the different σ values of the tolerance band. Small values of σ imply greater number of channels to isolate due to more violated models, thus making the isolation harder.

[3] F. Serdio, E. Lughofer, K. Pichler, T. Buchegger, and H. Efendic, "Residual-based fault detection using soft computing techniques for condition monitoring at rolling mills," *Information Sciences*, vol. 259, pp. 304–330, 2013.

[4] S. Ding, *Model-based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools*. Berlin Heidelberg: Springer, 2008.

[5] R. Isermann, *Fault Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Berlin Heidelberg: Springer, 2009.

[6] J. Korbicz, J. Koscielny, Z. Kowalczuk, and W. Cholewa, *Fault Diagnosis - Models, Artificial Intelligence and Applications*. Berlin Heidelberg: Springer Verlag, 2004.

[7] F. Serdio, E. Lughofer, K. Pichler, T. Buchegger, and H. Efendic,

TABLE III

ENGINE DATASET. FI RATIO AND AVERAGE ± STANDARD DEVIATION NUMBER OF INVOLVED CHANNELS DURING FI, BASED ON FUZZY FAULT ISOLATION RATES.

Method	5% Fault			20% Fault			100% Fault		
	Max. Overdetections			Max. Overdetections			Max. Overdetections		
	< 0.8%	< 1.6%	< 2.4%	< 0.8%	< 1.6%	< 2.4%	< 0.8%	< 1.6%	< 2.4%
PLS	59.42 10.19±0.74	37.90 10.22±0.91	37.90 10.22±0.91	65.20 10.66±1.59	51.45 10.65±1.66	51.45 10.65±1.66	75.83 11.28±2.09	59.80 11.20±2.12	59.80 11.20±2.12
PLS+ lags	21.65 8.61±1.12	21.65 8.61±1.12	21.65 8.61±1.12	44.70 8.37±1.65	44.70 8.37±1.65	44.70 8.37±1.65	63.53 8.70±2.16	63.53 8.70±2.16	63.53 8.70±2.16
SparseFIS	42.10 6.42±1.61	34.74 6.84±1.98	34.74 6.84±1.98	48.59 6.49±1.81	47.77 6.67±1.98	47.77 6.67±1.98	44.47 6.48±1.62	41.69 6.65±1.80	41.69 6.65±1.80
SparseFIS+ lags	50.95 6.57±0.62	41.75 6.68±0.57	31.47 6.81±0.48	72.96 6.60±0.95	56.06 6.70±0.67	41.54 6.77±0.53	68.48 6.67±0.65	53.51 6.86±1.03	45.51 6.93±0.96

TABLE IV

AVERAGE AND STANDARD DEVIATION OF THE NUMBER OF CHANNELS INVOLVED DURING THE ISOLATION PROCESS.

Method	PLS										PLS+Lags									
Fault	5%		10%		20%		50%		100%		5%		10%		20%		50%		100%	
$n*\sigma$	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
1	16.38	2.58	16.45	2.65	16.63	2.62	16.94	2.66	16.94	2.63	20.20	5.42	20.47	5.21	20.99	5.02	21.49	5.17	21.86	4.98
2	12.88	2.90	13.13	2.99	13.16	3.09	13.22	3.16	13.41	3.19	12.13	4.50	12.42	4.70	13.21	4.90	14.15	5.25	14.74	5.75
3	10.90	1.98	11.00	2.08	11.17	2.08	11.38	2.25	11.38	2.24	10.25	3.25	10.28	3.11	10.85	3.43	11.39	3.97	11.74	4.26
4	10.53	1.41	10.60	1.55	10.71	1.64	11.03	1.92	11.14	1.99	9.73	3.05	9.80	3.23	9.65	3.03	9.84	2.74	10.35	3.55
5	10.25	1.06	10.48	1.48	10.49	1.54	10.90	1.88	11.02	1.90	8.97	2.00	9.04	2.41	8.97	2.29	9.29	2.44	9.58	2.52
6	10.22	0.91	10.52	1.55	10.65	1.66	10.96	1.92	11.20	2.12	9.79	2.56	9.51	2.83	9.19	2.82	9.12	2.56	9.66	2.33
7	10.19	0.74	10.50	1.35	10.66	1.59	10.81	1.78	11.28	2.09	9.41	1.95	9.25	2.03	8.99	2.80	9.05	2.40	9.09	2.34
8	10.29	0.88	10.61	1.54	10.89	1.76	10.95	1.87	10.83	1.90	8.61	1.12	8.49	1.22	8.37	1.65	8.61	2.23	8.70	2.16
9	10.32	0.92	10.37	1.06	10.63	1.52	10.66	1.61	10.84	1.72	8.24	1.19	8.08	1.24	8.16	1.68	8.23	1.99	8.58	2.14
10	10.00	0.00	10.13	0.61	10.50	1.48	10.57	1.46	10.39	1.16	8.58	0.49	8.47	0.67	8.16	1.70	7.98	2.15	8.31	1.94

Method	SparseFIS										SparseFIS+Lags									
Fault	5%		10%		20%		50%		100%		5%		10%		20%		50%		100%	
$n*\sigma$	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
1	12.27	4.20	12.29	4.20	12.40	4.17	12.31	4.18	12.43	4.24	8.31	2.58	8.23	2.60	8.10	2.53	8.01	2.57	8.29	2.70
2	10.03	3.79	10.15	3.82	10.34	3.80	10.24	3.80	10.43	3.80	7.64	2.12	7.67	2.16	7.57	2.10	7.61	2.16	7.55	2.15
3	8.03	2.54	8.01	2.60	8.04	2.64	8.15	2.79	8.05	2.65	7.18	1.63	7.16	1.58	7.17	1.66	7.04	1.49	7.13	1.62
4	7.70	2.49	7.68	2.55	7.60	2.50	7.60	2.45	7.60	2.55	7.04	1.19	7.07	1.28	6.93	1.11	6.85	0.90	6.85	0.92
5	7.11	1.88	7.17	2.11	7.01	2.00	6.89	1.89	7.08	2.02	6.81	0.48	6.85	0.71	6.77	0.53	6.77	0.53	6.93	0.96
6	6.84	1.98	6.88	1.98	6.67	1.98	6.44	1.74	6.65	1.80	6.68	0.57	6.71	0.79	6.70	0.67	6.73	0.68	6.86	1.03
7	6.42	1.61	6.69	2.12	6.49	1.81	6.47	1.80	6.48	1.62	6.57	0.62	6.70	0.90	6.60	0.95	6.68	0.78	6.67	0.65
8	6.43	1.62	6.93	2.29	6.61	1.91	6.64	1.88	6.53	1.65	6.91	0.71	6.97	0.71	6.80	0.71	6.87	0.86	6.86	0.71
9	5.87	1.15	6.31	1.93	6.29	1.83	6.09	1.35	6.15	1.38	6.96	0.76	7.00	0.76	6.88	0.77	6.74	0.81	6.97	1.19
10	5.69	1.07	6.41	2.25	6.43	1.94	6.14	1.36	6.16	1.33	6.92	0.92	7.07	0.88	6.85	0.85	6.68	0.88	6.81	1.30

"Fault detection in multi-sensor networks based on multivariate time-series models and orthogonal transformations," *Information Fusion*, vol. In Press, Accepted Manuscript, 2014.

[8] J. Abonyi, *Fuzzy Model Identification for Control*. Boston, U.S.A.: Birkhäuser, 2003.

[9] G. Li, S. Qin, and D. Zhou, "Geometric properties of partial least squares for process monitoring," *Automatica*, pp. 204–210, 2010.

[10] E. Lughofer and S. Kindermann, "SparseFIS: Data-driven learning of fuzzy systems with sparsity constraints," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 2, pp. 396–411, 2010.

[11] O. Nelles, *Nonlinear System Identification. From Classical Approaches to Neural Networks and Fuzzy Models*. Berlin Heidelberg, Germany: Springer, 2001.

[12] W. Li, H. Yue, S. Valle-Cervantes, and S. Qin, "Recursive PCA for adaptive process monitoring," *Journal of Process Control*, vol. 10, no. 5, pp. 471–486, 2000.

[13] H. Schöener, B. Moser, and E. Lughofer, "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)*, Vienna, Austria, 2008, pp. 414–419.

[14] P. Odgaard, B. Lin, and S. Jorgensen, "Observer and data-driven-model-based fault detection in power plant coal mills," *IEEE Transactions on Energy Conversion*, vol. 23, no. 2, pp. 659–668, 2008.

[15] H. Chen, P. Tino, X. Yao, and A. Rodan, "Learning in the model space for fault diagnosis," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, pp. 124–136, 2014.

[16] M. Haenlein and A. Kaplan, "A beginner's guide to partial least squares (PLS) analysis," *Understanding Statistics*, vol. 3, no. 4, pp. 283–297, 2004.

[17] Y. Bang, C. Yoo, and I. Lee, "Nonlinear pls modeling with fuzzy inference system," *Chemometrics and Intelligent Laboratory Systems*, vol. 64, pp. 137–155, 2002.

[18] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction - Second Edition*. New York Berlin Heidelberg: Springer, 2009.