

Model-based Fault Detection in Multi-Sensor Measurement Systems

Edwin Lughofer
Fuzzy Logic Laboratorium Linz-Hagenberg
e-mail edwin.lughofer@jku.at

Erich Peter Klement
Fuzzy Logic Laboratorium Linz-Hagenberg
e-mail ep.klement@jku.at

Abstract — In the process and manufacturing industries, there has been a large push to produce higher quality products, to reduce product rejection rates, and to satisfy increasingly forceful safety and environmental regulations. Hence, the increasing complexity of measurement systems inside modern industrial processes with a rising amount of actuators and sensors demands automatic fault detection algorithms which can cope with a huge amount of variables and high-frequented dynamic data. Indeed, humans are being able to classify sensor signals by inspecting by-passing data, but this classifications are very time-consuming then and also have deficiencies because of underlying vague expert knowledge consisting of low-dimensional mostly linguistic relationships. In this paper we propose a model-based fault detection algorithm which is generic in the sense, that any model correctly describing a functional dependency inside a system can be enclosed easily almost without adjusting any thresholds or other essential parameters. This advanced 'residual view' fault detection includes aspects for incorporating sensor inaccuracies and model qualities as well as processing further normalized residuals for obtaining fault probabilities. Validation results with respect to data coming from engine test benches are included at the end of the paper.

Key words — *measurement systems, automatic fault detection, classification, model-based fault detection, sensor inaccuracies, model qualities, normalized residuals, fault probabilities*



Johannes Kepler Universität Linz



Institut für Algebra, Stochastik und
wissensbasierte mathematische Systeme
Johannes Kepler Universität Linz
A-4040 Linz
Austria

Fuzzy Logic Laboratorium Linz-Hagenberg

FLLL
Softwarepark Hagenberg
Hauptstrasse 99
A-4232 Hagenberg
Austria



1 Introduction

The IFAC Technical Committee SAFEPROCESS has defined a fault formally in the following way (also referenced in [IB96] and [CRB01]):

‘Unpermitted deviation of at least one characteristic property or variable of the system from acceptable/usual/standard behaviour.’

Moreover, fault detection is formulated as:

‘Determination of faults present in a system and time of detection.’

In the literature various approaches exist for performing data-based fault detection, which can be divided basically into 3 main groups:

1. Classification-based fault detection: different system conditions representing faulty cases define different fault classes due to their appearance in the corresponding recorded data (= fault patterns); also the faulty-free case represents one class; these classes are learned by training data and are applied whenever a new data point needs to be assigned to a class. The big disadvantage of this approach is, that all kind of faults and their corresponding appearance in the data need to be known a-priori and therefore new faults cannot be detected very often (only in the case that its pattern is luckily similar to an already known pattern of another fault)
2. Signal-based fault detection in intelligent sensors: can be seen as a single channel check approach where dynamic sensor data is analyzed with respect to the occurrence of peaks, drifts or other unnatural behaviors in their corresponding signal curves. As commonly no interactions in form of redundancy or correlation analysis between other sensors are taken into account, no wide-spread system failures can be detected within this approach.
3. Model-based fault detection: multi-dimensional models or some of their parameters are trained from simulated, historic or online measured data and used as reference situation characterizing functional dependencies between measurement variables for the faulty-free case. The drawback for this approach is, that if systematic failures occur in the train data (when no simulated data is available, of course), wrong models are trained, which get useless for fault detection. Besides, a fault isolation strategy as for example proposed in [ESR03] has to be appended in order to identify the faulty variables amongst faulty-free ones, all integrated in complex high-dimensional models.

While in [CRB01] a classification-based approach by using discriminant analysis functions which apply *PCA = Principal Component Analysis* and some of its variations or *FDA = Fisher Discriminant Analysis* are suggested and also successfully verified on data collected through process simulation for the *TEP = Tennessee Eastman process*, a statistical-based fault detection with a new general approach, called *structural hypothesis tests* is introduced in [Nyb99], which is theoretically grounded in classical hypothesis testing and propositional logic. Other approaches use intelligent sensors for performing signal analysis methods and filtering ([FAB92]) on high

dynamic data inside sensors for detecting sensor faults like peaks, (mean) drifts or other anomalies ([AA96]) in the signals' time or frequency series. In [HLS95], [CP99] model-based fault detection, also called 'residual view'-fault detection, as residuals are calculated representing the deviation between measured process values and estimated output values from corresponding models representing the nominal faulty-free case, is based on residual observer functions which entail a funded analytical theory, whereas more or less heuristic model-based fault detection approaches can be found in [SRG⁺01] and [LPK⁺95]. In [LPK⁺95] as well as in [KI] the 'residual view' approach is adjusted to special systems including inspections of some variables together with particular possibly occurring faults inside a special measurement system. This means also that manual threshold tuning and very specific residual calculations are carried out therein.

In this paper these drawbacks should be overcome and a generic approach for model-based fault detection in a multi-sensor system is demonstrated, which takes into account arbitrary many and any kind of models describing physical dependencies inside the system together with their qualities, incorporates sensor inaccuracies for automatic thresholding and calculates not only pure deviation from the model-based nominal case represented as a normalized residual but also real probabilistic funded fault probabilities based on rated historic data. Hence, it can be seen as an advanced 'residual view' fault detection approach which is feasible to be applied in various industrial processes wherever continuous data is measured and recorded. Additionally, this paper can be seen as an extension of the paper [SRG⁺01] with the main focus on fault detection instead of data-driven modelling, entailing a better performance for the same data set as applied in [SRG⁺01].

First in the proximating chapter the overall goal of an automated fault detection system is formulated, followed by theoretical aspects of model-based fault detection together with a generic algorithm for tracing faults no matter which kind of models are used as reference situation; validation results with respect to miscellaneous kind of models applied onto 2 different data sets obtained from measurement recordings at engine test benches in chapter 4 conclude the paper.

2 Problem Statement

Let be (x_1, x_2, \dots, x_n) n independent measurement variables in an arbitrary industrial process recorded dynamically with a certain frequency (=time steps), then a general static model for a specific variable x_i at time instant k can be defined as

$$x_i = y = f_k(x_{j_1}, \dots, x_{j_n}), j_m \in \{1, \dots, n\} \setminus i \quad (1)$$

whereby in the case of a time-variant dynamic model a time shift of the variables has to be taken into account, too, hence

$$x_i = y = f_k(x_{j_1}(k), x_{j_1}(k-1), \dots, x_{j_1}(k-l), \dots, x_{j_n}(k), x_{j_n}(k-1), \dots, x_{j_n}(k-l)), j_m \in \{1, \dots, n\} \setminus i \quad (2)$$

i.e. a dependency between x_i and a subset of the other measurement variables in the system is described by a model f_k at time instant k . If f_k stands for an analytical formula where all the parameters are known and set a-priori then $f_k = f_{k+m} \forall m \in \mathbb{N}$, which means that no parameters of the model and hence the model itself is never changed over time. If f_k stands for a data-driven model, i.e. its parameters are estimated from simulated, historic or online data corresponding to

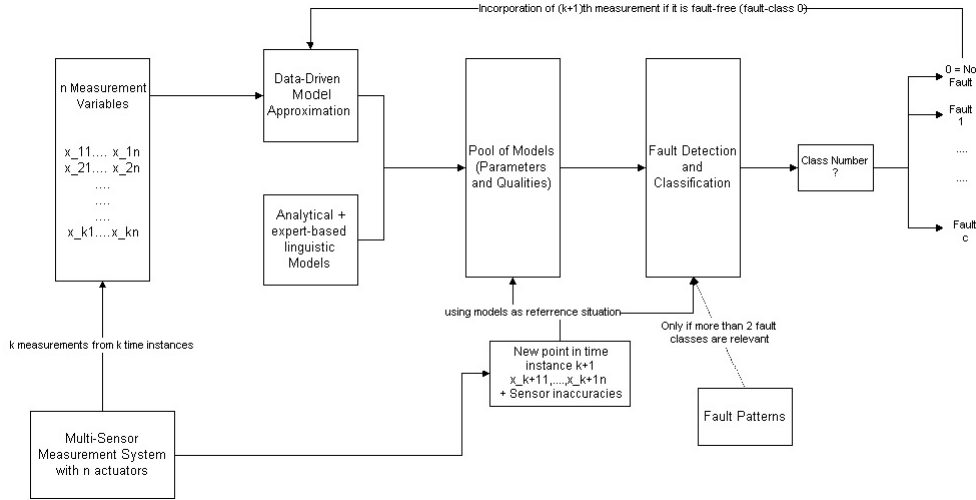


Figure 1: Fault Detection Scheme in an Online Measurement and Plausibility Check System

the same process, or some parameters in an analytical formula need to be adjusted (e.g. with respect to the ranges of some variables), $f_k \neq f_{k+m} \forall m \in N$, whenever new recorded data points \vec{x}_{k+m} are incorporated in the model, which is usually the case, above all in dynamic systems in order to be up-to-date and to obtain better approximations. Moreover, as data-driven models are applied whenever there are no well-known formal descriptions for some variables, their input structure, i.e. the subset of measurement variables, most significant for approximating a certain variable x_i , is not parametrized a-priori and has therefore also be calculated out of the train data. Several methods exist for performing this task ([Mil02]).

With these notations the goal of a fault detection framework as shown in figure 1 can be formulated in the following way:

Goal 1. *Let be $f_{1,k}, \dots, f_{m,k}$ various models as defined in equations (1) or (2) describing some relationships inside an industrial process at an arbitrary point of time k , then newly recorded points $\vec{x}_{k+1}, \dots, \vec{x}_{k+m}$ should be classified in a way by using these models as reference situation, such that the number of correct classifications should be as high as possible.*

If only 2 classes of fault appearances occur, i.e one class representing the fault-free case and the other representing all possible faulty cases, the correct classifications can be split into correct detections of faults and into correct detection of no faults, both influencing the *detection rate* and *overdetection rate* as defined in chapter 3.5. This splitting up is done, because mostly detection and overdetection rates play different roles with different priorities (see chapter 4).

Figure 1 presents a fault detection framework in an online measurement and plausibility check system; the figure should be more or less self-explainable, fault patterns have only be trained from data with known faults together with their appearance and fed into the system, if more than 2 fault classes should be distinguished. In the case of offline fault detection the framework would look similar, except that data-driven models would be fully trained before checking a whole data set and therefore no incorporation of checked and classified fault-free points into model re-estimation or adaptation would take place.

In the following chapter key aspects for achieving this goal are demonstrated: in chapter 3.2 aspects for fault detection in the case of 2 fault classes are stated, which are verified and validated in chapter 4 and extended to a multi-class approach in chapter 3.3.

3 Theoretical Aspects

3.1 Models as Reference Situation

Basically, 3 different kinds of models can be distinguished:

- Analytical models
- Knowledge-based models and
- Data-driven models

While analytical models originate from research results and therefore are founded on a solid theoretical basis and knowledge-based models consist mainly of linguistic expert knowledge, e.g. coded in fuzzy or other expert systems, data-driven models have the purpose to approximate unknown relationships inside a system in a way, such that convergence to the real function is guaranteed with increasing amount of data points, i.e famous Stone-Weierstrass theorem

$$\lim_{k \rightarrow \infty} f_k(x_{j_1}, \dots, x_{j_n}) = f \quad (3)$$

$$\lim_{k \rightarrow \infty} e(k) = 0 \quad (4)$$

is fulfilled, where f is the real underlying dependency and e the approximation error calculated as L_p -norm between measured and from the model estimated output values. Moreover, the data-driven model architecture should support universal approximation, which means that every real-occurring function can be approximated to a given degree of accuracy, so the following theorem is fulfilled:

Theorem 1. *For any given real continuous function $f(x) : X \subset R^n \rightarrow Y \subset R$ where X and Y are compact, and an arbitrary $\epsilon > 0$, there exists a model inside the group of models with a particular model architecture $g(x; \Phi) : X \rightarrow Y$, Φ the set of parameters such that*

$$\sup_{x \in X} |g(x; \Phi) - f(x)| < \epsilon \quad (5)$$

As fuzzy systems fulfill this property ([Wan92]) and have the additional advantage that they represent linguistic interpretable expert systems describing real-life relationships, they are widely applied for data-driven approximations in industrial systems. Opposed to that, neural networks approaches also fulfill the universal approximation property, but they lack of being interpretable as they result in pure black box models. Different algorithms exist for generating and adapting data-driven fuzzy models, whose explanations are not the topic of this paper, some approaches can be found in [BHB01], [YF94] or [LK03].

3.2 Fault Condition and Residual Generation

Let $(\vec{x}, y)_k$ a new incoming measurement point at time instance k , $y_k = x_{ik}$ be the measured output process variable and \hat{y}_{k-1} the estimated value due to an arbitrary model describing a relationship as stated in equation 1 or 2 and in the case of data-driven models fulfilling 3 and 5, then the residual describing the degree of deviation can be calculated through

$$res_k = \frac{\|\hat{y}_{k-1} - y_k\|_1}{bias_error} \quad (6)$$

where the *bias error* denotes approximation quality of the model due to its flexibility, hence the error on the training data for data-driven models. This approach for residual calculation is more or less the same as described in [LPK⁺95], but extended with the division through the *bias error* which can be justified in the following way: models with a high *bias error* are unconfident for delivering the correct estimated value, therefore a large deviation between estimated and measured output value is weighted lighter than a large deviation obtained by a more trustful model with a low *bias error*. For analytical and knowledge-based models $\hat{y}_{k-1} = \hat{y}$, if there is no refinement of some parameters in these models with new online data, and the *bias error* need to be estimated by experience or by simulated data applied to the models. Moreover, this division in equation 6 can be seen as a normalization of residuals in order to make them comparable amongst various models, as different models include different variables with different ranges, hence a common threshold can be applied and the fault condition, i.e. the condition that there is a fault in the actual k th measurement becomes

$$\exists m \frac{\|f_{m,k-1} - y_k\|_1}{bias_error_m} > thresh \quad (7)$$

where $f_{m,k-1}$ denotes the estimated output value from the m th model representing a specific relationship at time instance $k-1$, i.e. a significant violation of one model (amongst say r models) is enough in order to classify the measurement as faulty. This fact is justified, as different models possess a different structure containing different input/output variables and a faulty value in one variable should be always detected.

Indeed, this fault condition approach incorporates the confidence of the models but does not take into account inaccuracies in the sensors which usually exist in measurement systems — see figure 2, where the dynamic data signal of a certain channel at an engine test bench is shown: the range of inaccuracy varies from -3 till +3 and also can effect the corresponding stationary point which is evaluated by averaging over the dynamic points.

Different levels of sensor inaccuracies can appear for different variables and also in different measurement systems, hence in order to guarantee automatic thresholding and improve the correctness of fault detection statements obtained through condition 7 amongst models, the so-called *variance error* for including uncertainties due to the noise levels is incorporated into the estimated values of models:

$$\hat{y}_{m,k-1} = f_{m,k-1} \pm model_error_m = f_{m,k-1} \pm \sqrt{bias_error_m^2 + var_error_m} \quad (8)$$

Obviously, in the case of a high *bias error* the *variance error* can be neglected, while in the case of a low *bias error* the *variance error* gives a significant contribution to the whole model error which is essentially important to avoid the famous overfitting effect ([Sch93]). For data-driven as

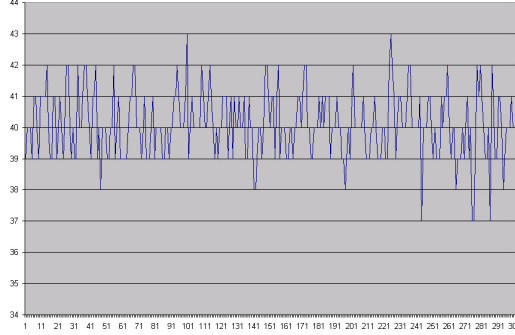


Figure 2: Dynamic sensor signal of a measurement variable at an engine test bench: the average and correct value is 40 while the inaccuracy level is ± 3 ; x-axis: time in tenth parts of second, y-axis: values of measurement variable

well as analytical and knowledge-based models the *variance error* can be computed by applying error back-propagation law (+++ reference +++) (neglecting the time component)

$$\epsilon_{y_m} = \left| \frac{\partial f_m}{\partial x_1} \right| \epsilon_{x_1} + \left| \frac{\partial f_m}{\partial x_2} \right| \epsilon_{x_2} + \dots + \left| \frac{\partial f_m}{\partial x_n} \right| \epsilon_{x_n} \quad (9)$$

which incorporates sensor inaccuracies of all input variables x_1, \dots, x_n contained in the model f_m . The drawback of this approach lies in the indispensable a-priori knowledge about sensor inaccuracies for all variables in a system which is not always available. For data-driven models another possibility arise from applying the test error, i.e. the error of new fresh test data to a model, by following formula

$$var_error = e_{test} - e_{train} \quad (10)$$

where e_{test} denotes the error on the test data and e_{train} the error on the train data. As a complete data set needs to be split up into a train data set and a test data set, this approach has a drawback when only few data points are available for training which it is usually the case for online identification and fault detection tasks.

With these notations and $\tilde{y}_{m,k} = y_{m,k} \pm \epsilon_y$, where ϵ_y denotes the inaccuracy level of those sensor which samples model's output variable y

$$\hat{y}_{m,k-1} - \tilde{y}_{m,k} = f_{m,k-1} \pm \sqrt{bias_error_m^2 + var_error_m} - y_{m,k} \mp \epsilon_y \quad (11)$$

leading to the fault condition

$$\begin{aligned} \exists m \ f_{m,k-1} - y_{m,k} \mp \epsilon_y - \sqrt{bias_error_m^2 + var_error_m} &> t \quad \vee \\ f_{m,k-1} - y_{m,k} \mp \epsilon_y + \sqrt{bias_error_m^2 + var_error_m} &< -t \quad t \in N_0 \end{aligned} \quad (12)$$

or by choosing t without the loss of generality as a positive integer multiplicator, of $\sqrt{bias_error_m^2 + var_error_m}$, lets call it *thresh*, leading to

$$\begin{aligned} \exists m \ \frac{f_{m,k-1} - y_{m,k} \mp \epsilon_y}{\sqrt{bias_error_m^2 + var_error_m}} &> thresh \quad \vee \\ \frac{f_{m,k-1} - y_{m,k} \mp \epsilon_y}{\sqrt{bias_error_m^2 + var_error_m}} &< -thresh \quad thresh \in N_0 \end{aligned} \quad (13)$$

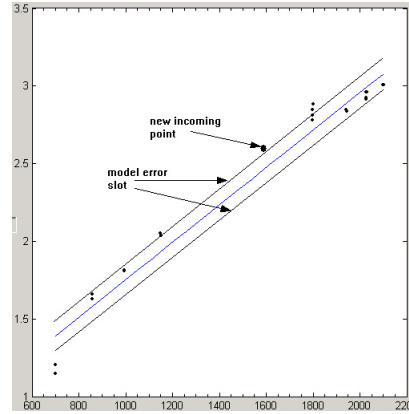


Figure 3: New incoming point (marked as big dot) is incorrectly classified as faulty because threshold is set due to the model error band which is too tight

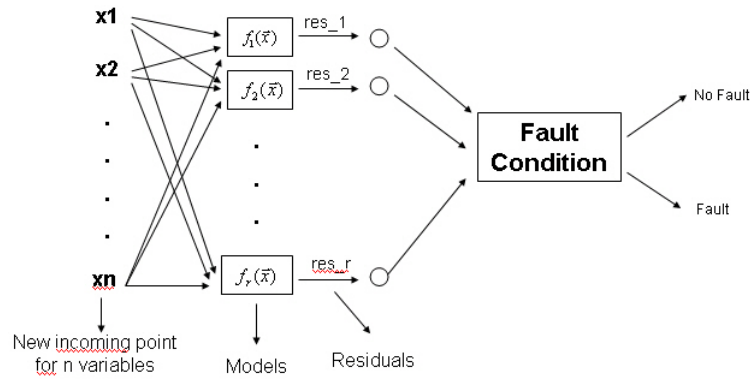


Figure 4: Residual-based fault detection scheme

As ϵ_y is always positive the fault condition becomes

$$\begin{aligned} \exists m \frac{f_{m,k-1} - y_{m,k} - \epsilon_y}{\sqrt{\text{bias_error}_m^2 + \text{var_error}_m}} &> \text{thresh} \quad \vee \\ \frac{f_{m,k-1} - y_{m,k} + \epsilon_y}{\sqrt{\text{bias_error}_m^2 + \text{var_error}_m}} &< -\text{thresh} \quad \text{thresh} \in N_0 \end{aligned} \quad (14)$$

If $t = 0$, i.e. a kind of one- σ area, is used in order to allow only values between the 'model error band', this will lead to a too tight threshold as demonstrated in figure 3 which will probably result in too much overdetections.

By applying this approach the internal flowchart of the 'Fault Detection and Classification' box stated in figure 1 have the appearance as it is demonstrated in figure 4: residuals are generated from all models, collected and fed into the fault condition as defined in 14 in order to discriminate between faulty and fault-free measurements x_1, \dots, x_n at an arbitrary point of time (thus, the time instance k is neglected in this figure).

Additionally, the residual calculated in 14 as the left hand sides of the inequalities is transferred to a so-called *error hint fuzzy* value, which normalizes the residual into the interval $[0, 1]$, as the transfer function *transfunc* fulfills the following conditions:

- $transfunc(0) = 0$
- $transfunc(thresh) = 0.5$, such that a value of higher than 0.5 can always be classified as a fault
- $transfunc(max_res) = 1$, where *max_res* denotes the maximal value of a residual
- *transfunc* is monotonic

This normalization is indispensable, when taking into account that different components in a system can exist which should interact and come to a on overall plausibility check and fault detection statement in a second stage. Indeed, above defined fault condition can be applied in a generic way in all model-based fault detection modules, more or less, but sometimes first the threshold need to be tuned a little bit which will make the residuals incomparable amongst the modules and second other components can use non-model-based fault detection which generates no residuals but directly fault probabilities. In these cases, a normalization of the outcome of the fault detection modules, in form of an *error hint fuzzy* value guarantees comparability amongst modules and therefore a valid consolidation in order to obtain overall fault detection statements. An example for a transformation function is given by

$$f(x) = \begin{cases} a_1x^2 & x < T \\ a_2x^2 + bx + c & T \leq x < max_res \\ 1 & x \geq max_res \end{cases} \quad (15)$$

From above mentioned conditions, the function value at the 'critical position' $T = thresh$ should always be 0.5 for all transformation functions. Besides, the condition $f(max_res) = 1$ should be fulfilled for the second parable in equation, and the gradient at the position *max_res* should have value 0 to get a smooth transition from parable part 2 to 3; hence we obtain following formulas for the parameters a_1 , a_2 , b and c :

$$a_1 = \frac{1.0}{2T^2} \quad (16)$$

$$a_2 = \frac{-1.0}{2(max_res - T)^2} \quad (17)$$

$$b = \frac{max_res}{(max_res - T)^2} \quad (18)$$

$$c = 0.5 + \frac{T^2 - 2Tmax_res}{2(max_res - T)^2} \quad (19)$$

The advantages of employing this transformation function are the comparability of the linguistic interpretability of residuals in the case of different fault detection methods with different critical positions (T) and that a residual with a value 0 is transferred to an error hint with value 0. For proofing the comparability of the linguistic interpretability assume that 2 different critical positions ($T_1 \neq T_2$) for 2 different fault detection methods are fixed and $f(T - \frac{T}{\alpha})$ can be linguistically interpreted as "tight under the critical position" in the case of $\alpha > N$, where N a big positive

integer and as "tight over the critical position" in the case of $\alpha < -N$, then it follows that for T_1 $a_1 = \frac{1}{2T_1^2}$ and therefore (for the case $\alpha > 0$)

$$f(T_1 - \frac{T_1}{\alpha}) = \frac{1}{2T_1^2}(T_1 - \frac{T_1}{\alpha})^2 = \frac{1}{2} - \frac{1}{\alpha} + \frac{1}{2\alpha^2} \quad (20)$$

For T_2 we obtain $a_1 = \frac{1}{2T_2^2}$ and therefore

$$f(T_2 - \frac{T_2}{\alpha}) = \frac{1}{2} - \frac{1}{\alpha} + \frac{1}{2\alpha^2} \quad (21)$$

In the same way it can be shown that this holds also for $\alpha < 0$.

3.3 Obtaining Fault Probabilities and a Multi-Classification Approach

The residual calculation together with the fault condition described in the previous chapter possesses two deficiencies:

- It just yields a degree of deviation from a fault-free reference situation, but it does not evaluate and state a real fault probability, i.e. delivering no statement how likely it is that there is a fault in the actual measurement
- It demonstrates only a 2-class approach (class #0 describing the faulty-free case, class #1 the faulty case) and cannot be applied when a multi-class approach for assigning faults to more than 2 fault patterns is demanded.

In order to overcome these drawbacks discriminant functions were considered to be applied onto features calculated out of model-based fault detection modules. In figure 5 an extension of the fault detection scheme as shown in figure 4 is demonstrated which can be also applied for the box 'Fault Detection' in figure 1. Instead of the fault condition defined in 14 which brings all the residuals from all models together and computes an overall statement discriminant analysis functions are applied denoted by $g_0(\vec{t}), g_1(\vec{t}), \dots, g_{c-1}(\vec{t})$. For each class out of c classes the discriminant functions deliver a value g_i which can be seen as an a-posteriori probability of \vec{t} belonging to class i . Obviously, the class with the highest probability wins and the corresponding measurement causing the feature vector \vec{t} is assigned to this class. A widely-used approach for discriminant analysis functions is the so-called Bayes' rule,

$$P(\omega_i|\vec{t}) = \frac{p(\vec{t}|\omega_i)P(\omega_i)}{p(\vec{t})} \quad (22)$$

where $P(\omega_i)$ is the a-priori probability for class ω_i , $p(\vec{x})$ is the probability density function for \vec{t} and $p(\vec{t}|\omega_i)$ is the probability density function for \vec{t} conditioned on ω_i . From this equation, it becomes clear that the probability density functions have to be known a-priori. Usually, this knowledge can not be provided analytically, hence train data representing all fault classes has to be used for estimating density functions and frequentness of appearing classes. Advanced methods such as PCA-based or Fisher discriminant analysis functions exist for obtaining even exacter classifications ([CRB01]).

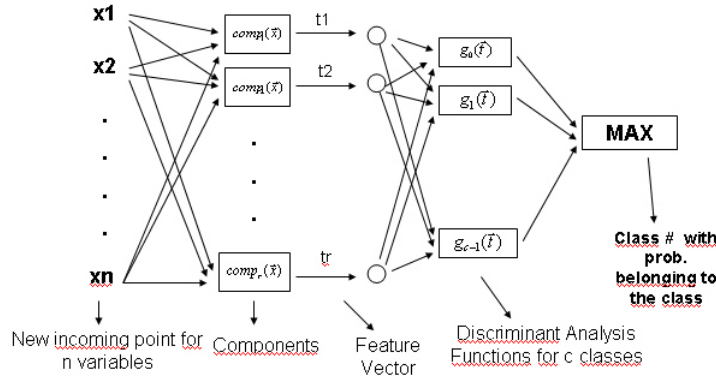


Figure 5: Fault detection scheme with applying discriminant analysis function in order to discriminate amongst several fault classes

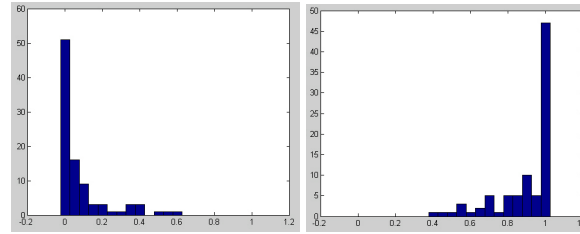


Figure 6: Left image: density distribution of *error hint fuzzy* for the fault-free case; right image: for the faulty case

Moreover, the fault detection scheme demonstrated in figure 5 generalizes the one in figure 4 by applying components instead of models. Components can be models, but can also be a whole collections of models, e.g. a collection of fuzzy systems or neural networks, or furthermore whole fault detection systems, each of them delivering feature vectors with the same meaning. What is left now, is the question which feature values should be derived and processed further to the discriminant functions. This question is unanswerable in a generic meaning, i.e. it depends on the whole approach how fault detection is carried out. For model-based fault detection and 2 fault classes, namely fault-free case and faulty case, it seems to be evident to apply residuals or *error hint fuzzy* as they are derived as in chapter 3.2, which is underlined in figure 6, where one single model is applied for residual calculation, entailing a feature vector with one component and hence a one-dimensional density estimation of the fault-free (left image) and the faulty case (right image): obviously, the 2 classes can be distinguished, leading to a high a-posteriori probability from Bayes' rule for the faulty case when the *error hint fuzzy* is high and a high a-posteriori probability for the fault-free case when the *error hint fuzzy* is low. If this distinction can be not made clearly, i.e. the density functions of 2 or more classes can be not separated over the whole domain, which is demonstrated in figure 7, the feature vector is probably not well suited for discrimination amongst classes. Figure 7 represents density estimations again due to *error hint fuzzy* for the fault-free (left) and faulty (right) case, using a collection of models as one component and obtaining maximal *error hint fuzzy* value from this collection: no suitable discrimination can

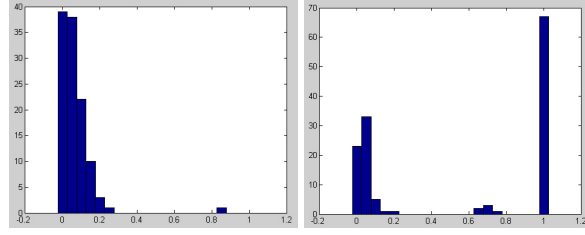


Figure 7: Left image: density distribution of *error_hint_fuzzy* for the fault-free case; right image: for the faulty case - in the lower region no clear discrimination between a faulty sample and a fault-free sample can be made as a similar fault probability value is obtained

be performed in the lower region.

Note: when using more components, the feature vector gets multi-dimensional entailing multi-dimensional density estimations.

3.4 Advanced Aspects

During verification and validation of model-based fault detection from the residual view as it is described in chapter 3.2 2 aspects were essential in order to guarantee good performance with respect to correctness of fault detection statements.

The first one was taking care of models with a low quality, i.e. models whose approximation was weak in the sense that they could not really generate a good formula for one specific target channel. This can happen in the case of data-driven models, as for each well-measured variable an approximation is tried with a subset of the remaining variables in the system. So, if no useful dependency for a certain target channel exists inside the system, the approximation gets unfeasible which causes a bad model quality. It finally turned out, that models with a quality of lower than 0.6, where 1 is a perfect model and 0 an almost nonsense model, should be omitted in order to avoid unpleasant overdetection rates. In the case of data-driven models, the model quality can be calculated by famous *r-squared-adjusted* formula, defined by

$$mod_{qual} = R^2_{adjusted} = 1 - \frac{(k-1)(1-R^2)}{k-df} \quad (23)$$

which normalizes the *r-squared* value R^2 and takes in account the discrepancy between the number of train data points (k) and the number of degrees of freedom (df). For analytical or knowledge-based models the model quality is either fixed due to specialist's experience or can be evaluated again via *r-squared-adjusted* formula with a few dozen of first incoming online data points.

The second aspects stems from the numerical instability of the fault condition in inequality 14, whenever the *model error* is very tiny because of a very exact approximation (\rightarrow low *bias error*) and in addition a low noise level (\rightarrow low *variance error*) in the data. Then, the denominator in inequality 14 approaches 0, the residual becomes unstable (can converge to infinity) which leads again to unpleasant overdetection rates. In order to overcome this drawback, a small artificial disturbance around 0.05 is added to the denominator.

3.5 Measure Values for Verification

In order to be able to verify, validate and compare the performance of model-based fault detection approach amongst various components containing models with a different origin, nature and architecture, e.g. for example comparing FD using data-driven neural network models with FD using data-driven fuzzy systems or with FD using analytical models as reference situation, we considered about measure values which should bring in an objective benchmark how the methods perform. On the basis of rated check data sets, i.e. where each row of the data set is marked as faulty or error-free, we defined *detection rate* by the relative frequency of detections, hence

$$A_{det} = \frac{N_{FD}}{N_{FM}} \quad (24)$$

where N_{FD} denotes the number of measurements with a correct detection and N_{FM} denotes the number of faulty measurements in the check data set, and the *overdetection rate* by the relative frequency of overdetections, hence by

$$A_{ov} = \frac{N_{FOD}}{N_{FFM}} \quad (25)$$

where N_{FOD} denotes the number of measurements with a wrong detection and N_{FFM} denotes the number of faulty-free measurements in the check data set. Obviously, a method FD_M1 with a higher *detection rate* and simultaneously a lower *overdetection rate* than another method FD_M2 is always superior to this method. But what in the case, if the *detection rate* is higher, but the *overdetection rate* is also higher for one method than the other? - for this case we considered an additional measure value, so-called *external FD-method quality*, simply *external quality*, which combines *detection rate* and *overdetection rate* in one value and incorporates the relative frequency of faulty data in the check data set at all:

$$ext_{qual} = (1 - w_{ov}A_{ov}) - \frac{m}{n}(1 - w_{ov}A_{ov} - w_{det}A_{det}) \quad (26)$$

where n is the amount of rated check data records, m the amount of faulty check data records, $n - m$ the amount of faulty-free check data records. The weights w_{ov} and w_{det} reflect the impact of the correct detections, respectively overdetections onto *external quality* measure. The weights can be adjusted in a way, such that the importance of correct detections dominates the importance of overdetections or vice versa.

4 Validation Results

In this chapter validation results for the model-based fault detection approach described in the previous chapter are given. 2 different data sets are used for validation:

- Real-life measured data at an engine test bench containing sensor inaccuracies as noise in the data together with a rated check data set containing real-life occurring faults and fault-free points
- Simulated data for a special diesel engine, containing no noise, together with a rated check data set containing faults with small deviations of 5% and 10% in some channels

Table 1: Comparison of detection and overdetection rates as well as external qualities of model-based FD approach amongst several methods (=components)

Method	Det. Rate	Overdet. Rate	Ext. Qual
Global Corr.	51.97%	0.00%	0.743
Local Corr.	64.57%	3.51%	0.797
Local Regr.	42.52%	0.00%	0.697
Fuzzy Sys.	55.15%	0.88%	0.75
Overall	70.59%	4.39%	0.82

The first data base denotes exactly the same data base as it is used for performance verification in [SRG⁺01] in the last chapter. Opposed to the figure in this chapter, where 3 different methods, namely FD using correlation models, FD using neural networks and FD using map approximation which was performed with fuzzy systems, were applied, the table 1 reflects better results, above all with respect to overdetection rates, due to improvements in data-driven modelling algorithms as well as through the fault detection method proposed in this paper: while the detection rates stayed nearly the same around 40% till 60%, the overdetection rate could be reduced from above 10% to below 1% which was a crucial point, as too much overdetections lead to a low operator's confidence in a fault detection system. 4 different kinds of methods for data-driven modelling were applied: global correlation method (in [SRG⁺01] simply stated as correlation models), local correlation method, local regression method (see [LG02]) and fuzzy systems estimation (similar to map approximation in [SRG⁺01]) based on the approach as described in [YF94] and [LK03], each of them obtaining several models describing dependencies of some measurement variables inside the system. Additionally, an overall fault detection result is shown, which combines the FD statements of all data-driven methods (=components) together to an overall statement (5th row in the table), hence more or less the scheme demonstrated in figure 5 is applied, but without using discriminant analysis functions, just by applying the fault condition for residuals as shown in figure 4.

In table 2 results with respect to the performance of detecting small errors based on simulated engine data is shown. Opposed to table 1, the results with respect to *detection rates* are much more weaker, which gets clear when inspecting the characteristics of the faults: in the simulated data file only artificially built-in errors deviating 5% or 10% from their original values are included, while the real-recorded engine test bench data contains intense faults like drifts, outliers or sensor drop outs.

5 Conclusion

All in all, it can be summarized that through the fault detection approach proposed in this paper better results with respect to *detection rate*, *overdetection rate* and *external quality* than in [SRG⁺01] and using a real-life recorded engine test bench data are achieved. Also the results for very small faults with deviations of around 5% or 10% for a simulated engine data are not as bad as somebody could expect. Another important observation arose when inspecting the impact of the amount of train data points on the data-driven models and thereafter on the whole fault detec-

Table 2: Comparison of detection and overdetetection rates as well as external qualities of model-based FD approach amongst several methods (=components) for simulated data containing faults with 10% deviation and 5% deviation (marked in the second column)

Method	Dev.	Det. Rate	Overdet. Rate	Ext. Qual
Global Corr.	10%	20.47%	0.00%	0.62
	5%	20.04%	0.00%	0.62
Local Corr.	10%	49.08%	0.00%	0.76
	5%	37.12%	0.00%	0.7
Local Regr.	10%	20.67%	0.00%	0.62
	5%	12.88%	0.00%	0.58
Fuzzy Sys.	10%	21.49%	0.00%	0.62
	5%	20.25%	0.00%	0.62
Overall	10%	53.67%	0.00%	0.78
	5%	42.33%	0.00%	0.73

Amount of Data	Optimal Threshold
100	100
200	80
300	60
400	50
500	40
600	40
.....
1810 = complete set	20

Figure 8: Dependency of optimal threshold leading to similar or even the same results as in table 1 in dependency of the used amount of train data for generating fuzzy models

tion process: the optimal threshold depended not only on the inflexibility of the models and on the uncertainties due to noise in the data, but also on the amount of train data for generating the models, e.g. for fault detection based on fuzzy system estimation the dependency between optimal threshold and number of train data points (in the case of the first data set) as shown in figure 8 was discovered: obviously, the threshold goes hand in hand with the amount of train data with a monotonic decreasing relationship. Therefore, in order to improve the automatization of model-based fault detection, an advanced thresholding strategy for data-driven models will be further considered and applied in an online fault detection framework.

References

- [AA96] Satnam Alag and Alice M. Agogino. A methodology for intelligent sensor measurement, validation, fusion, and fault detection for equipment monitoring and diagnostics, 1996.

- [BHB01] Martin Burger, Josef Haslinger, and Ulrich Bodenhofer. Regularized optimization of fuzzy controllers. Technical Report SCCH-TR-0056, Industrial Mathematics Institute and Software Competence Center Hagenberg, A-4232 Hagenberg, Austria, 2001.
- [CP99] Jien Chen and R.J. Patton. *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Kluwer Academic Publishers, 101 Philip Drive, Assinippi Park, Norwell, Massachusetts 02061, 1999.
- [CRB01] L.H. Chiang, E.L. Russell, and R.D. Braatz. *Fault Detection and Diagnosis in Industrial Systems*. Springer Verlag London Berlin Heidelberg, London, Great Britain, 2001.
- [ESR03] Hajrudin Efendic, Andreas Schrempf, and Luigi Del Re. Data based fault isolation in complex measurement systems using models on demand. In *Proceedings of the 5th IFAC Symposium Fault detection supervision and safety of technical processes - Safeprocess 2003*, pages 1149 – 1154. IFAC, June 2003.
- [FAB92] Jing Fang, Les Atlas, and Gary Bernard. Advantages of cascaded quadratic detectors for analysis of manufacturing sensor data. In *Proceedings IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pages 345–8, 1992.
- [HLS95] Pau-Lo Hsu, Ken-Li Lin, and Li-Cheng Shen. Diagnosis of multiple sensor and actuator failures in automotive engines. *IEEE Transactions on Vehicular Technology*, 44(4):779–789, November 1995.
- [IB96] R. Isermann and P. Ball. Trends in the application of model-based fault detection and diagnosis of technical processes. In *Proc. of the 13th IFAC World Congress, volume N*, pages 1–12. IEEE Press, 1996.
- [KI] Frank Kimmich and Rolf Isermann. Model based fault detection for the injection, combustion and engine-transmission.
- [LG02] Edwin Lughofer and Werner Groissboeck. Generating correlation and regression models from high dimensional measurement data - advanced aspects, strategies and validation. Technical Report FLLL-TR-0212, Fuzzy Logic Laboratorium Linz-Hagenberg, A-4232 Hagenberg, Austria, September 2002.
- [LK03] Edwin Lughofer and Erich Peter Klement. Online adaptation of Takagi-Sugeno fuzzy inference systems. In P. Borne, E. Craye, and N. Danguoumau, editors, *Proceedings of CESA'2003—IMACS Multiconference, Computational Engineering in Systems Applications, Lille*, 2003. CD-Rom, paper S1-R-00-0175, 8 pages, ISBN: 2-9512309-5-8.
- [LPK⁺95] E.G. Laukonen, K.M. Passino, V. Krishnaswami, G.-C. Lub, and G. Rizzoni. Fault detection and isolation for an experimental internal combustion engine via fuzzy identification. *IEEE Trans. on Control Systems Technology*, 3(9):347–355, September 1995.
- [Mil02] Alan Miller. *Subset Selection in Regression Second Edition*. Chapman and Hall/CRC, Boca Raton, Florida 33431, 2002.

- [Nyb99] Mattias Nyberg. *Model Based Fault Diagnosis, Methods, Theory, and Automotive Engine Application*. PhD thesis, Department of Electrical Engineering Linköping University, SE-581 83 Linköping, Sweden, May 1999.
- [Sch93] C Schaffer. Overfitting avoidance as bias. *Machine Learning*, 10:153–178, 1993.
- [SRG⁺01] Andreas Schrempf, Luigi Del Re, Werner Groissboeck, Edwin Lughofer, Erich Peter Klement, and Guenther Frizberg. Automatic engine modelling for failure detection. In *Proceedings 2001 ASME International Mechanical Engineering Congress and Exposition*, March 2001.
- [Wan92] L.X. Wang. Fuzzy systems are universal approximators. In *Proceedings of the IEEE International Conference on Fuzzy Systems*, pages 1163–1169, 1992.
- [YF94] Ronald R. Yager and Dimitar P. Filev. Generation of fuzzy rules by mountain clustering. Technical Report MII-1318R, Machine Intelligence Institute, Iona College, New Rochelle, NY 10801, January 1994.