Unsupervised Anomaly Isolation and Steady State Detection for Monitoring Dynamic Systems

EMAD ALI
TU-Dresden –Institute of Mechatronic Engineering

Mechatronic Systems Development
Bosch Rexroth AG, Bgm.-Dr.-Nebel Str.2, D-97816 Lohr,
GERMANY
emad.ali@boschrexroth.de http://www.boschrexroth.com

Abstract: - This paper deals with the problem of modelling and monitoring the fault-free states of an industrial process without complete knowledge about the entire machine components. The aim thereby is to automatically detect the deviations in performance as fault symptoms. For that type of data-based modelling, the algorithms of clustering are selected with an emphasis on the computational load and application complexity. Kohonen neural networks (self-organizing maps) are found suitable for the task due to the ability to efficiently operate on high dimensional data and because of their robustness against uncertainties. They reveal drawbacks from the perspective of identifying the deviating variable in the input space. A novel structure is designed to solve this dilemma by combining multi one-dimensional domains and their statistical relationships, where Kohonen and Bayesian algorithms would be directly applicable. The structure is introduced and applied to simulate the human supervisors in the way of learning normal operation and hence, attempts to automatically identify the deviating variable in a high amount of data. An example application is proposed for detecting the wear degradation fault in a real electrohydraulic drive that widely used in many industrial machines. The algorithm can be realized locally or integrated remotely in cloud architectures.

Key-Words: - condition monitoring, unsupervised machine learning, self-organizing maps, abnormality isolator, artificial neural networks, fault detections.

1 Introduction

Modern machines demand more smartness, usability and profitability. The application of machine learning (ML) algorithms in condition monitoring (CM) modules is a milestone towards these aims. ML is defined as *"automatic computing based on logical and binary operations to learn tasks and exploit facts from examples"* [1]. ML techniques can be categorized into supervised and unsupervised paradigms based on the learning form. The supervised ML is widely applied for tasks such as pattern recognition and time series predictions. The authors in [17], [5], [20] proposed the supervised ML to identify defects. The training data are typically gained through operation with injected faults; then a model is trained to classify the patterns assigned to each fault, as a class label. A popular algorithm is the feed forward artificial neural networks (FF-ANN). The output of the FF-ANN is bounded by the trained cases and the drawback of this, is incorrect classification by fault patterns that are not included in the training set. Therefore, the training data must contain all possible fault patterns in advance as in /31/ .This is practically hard to realize in CM problems as many faulty scenarios are not known practically in advance [16]. By the unsupervised ML, no need to classify the training data by labels and it is not necessary to include all possible faulty cases in the training data sets. The algorithms can automatically detect an anomaly based on the learned normal (i.e. fault-free) patterns. The abnormality is regarded in the data as a deviation or in other words, possible fault symptoms. The automatic recognition in this manner supports human supervisors for troubleshooting the cause and eventually adapt maintenance plans. For this concern, the related technique of unsupervised ML is data clustering which is defined as: *"an unsupervised learning approach, directly exploiting regularities in the data to be analyzed, that builds a higher level representation to be used for reasoning or prediction."* [2]. Comparing all clustering algorithm is beyond the scope of the paper. In [19], a large number of clustering algorithms are outlined and compared.

2 Kohonen Self Organizing Maps

2.1 Conventional Algorithm

Self-Organizing Maps (SOMs) are a special type of ANN that is suitable for clustering data and vector quantisation [12]. Each cluster could be interpreted as a local model. In the concern of CM, the data are collected from various machine components and hence are usually multivariate. Consider an input vector $x(t)$ of n variables that is gained during healthy operation:

$$x(t) = [\varepsilon_1(t), \varepsilon_2(t), \varepsilon_3(t), \dots \varepsilon_n(t)] \quad (1)$$

By clustering using conventional SOMs, a model m_i is associated with cluster i as depicted in Fig. 1. m_i. can be expressed as a vector of weights:

$$m_i(t) = [w_{1,i}(t), w_{2,i}(t), w_{3,i}(t), \dots w_{n,i}(t)] \quad (2)$$

Where $w_{k,i}$ is the weight of the input dimension k to the cluster node i. A data item will be associated into the node whose model is most similar to it [13].

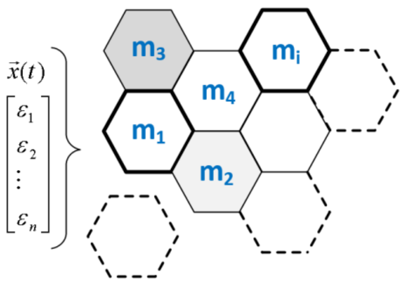

Fig. 1. Models associated with the clusters

The similarity measure is based on the geometrical Euclidean distance. The training performs an iterative adaption to the weights that takes the general form:

$$m_i(t+1) = m_i(t) + \alpha(t)h_{ci}(t)[x(t) - m_i(t)] \quad (3)$$

$$D_i = \|x(t) - m_i(t)\|, for i: 1 \to k$$
$$D_c = \min\ (D_i) \quad (4)$$

As $\alpha(t)$ is a learning rate. The subscript c indicates the node that has the minimal Euclidean distance, D_c to the input vector $x(t)$. The node m_c is called the winner neuron, whose model will be updated to be more similar to the training input vector. Due to this strategy, the SOMs are called competitive networks [11]. SOMs are typically constructed as 2D topological grids. Other map nodes in the neighborhood of m_c would be updated by a function $h_{ci}(t)$, Eq. (3), as a kind of smoothing kernel [13]. A typical use of the neighborhood function is the Gaussian [10]. The training is performed unsupervised in one of the following schemes

- Sequential Training: Data samples are fed one per iteration. The weights are updated following each sample
- Batch Training: The algorithm loops on the whole set iteratively, each run is called 'epoch'. The weights are updated per epoch.

After the training is complete, the weights of all the nodes, are registered in a matrix called *'the code book'* of the map. The map is then used to estimate the winning node, named then as *'the best match unit'* (BMU). The BMU is thus the output of a cost function that minimizes the quantization error , Eq. 5

$$Qe = ||\, x(t) - m_i|| = \sqrt[2]{(\varepsilon_1 - w_{i,1})^2 + (\varepsilon_2 - w_{i,2})^2 \ \dots + (\varepsilon_n - w_{i,n})^2} \quad (5)$$

The domain of BMUs describes the normal fault free operation of the target machine. Deviations are supposed to result in an increase in the training Q_e or produce a BMU out of trained normality domain.

2.2 Discussion

SOMs are generally powerful for clustering high dimensional data. In addition, they can be applied as a way of black box modelling [7]. Unlike other clustering algorithms, they reveal interesting characteristics for developing a practical methodologies for CM. Kohonen algorithm for clustering provides advantages that it maintains two features at the same time:

Feature 1, Application flexibility: The training can be carried out in a batch or sequential paradigm that requires low computational cost. So it can be operate online by PLCs that have limited resources. Therefore, the developed functions would be potentially applicable in a vast variety of systems.

Feature 2, Configuration simplicity: Human operators do not have precise knowledge in advance about the nature of the data or/and the number of the clusters.The SOM grid nodes do not necessarily have to be fully assigned to data points. Empty nodes are found frequently in such networks without claims on performance or accuracy.

Therefore, the definition of the exact number of cluster nods is not a prerequisite.

On the other hand, the conventional algorithm reveals drawbacks in the way of fault detections:

- Probably incorrect monitoring. The mechanism of estimating the similarity of a faulty multivariate state may result in an equivalent Qe as noa rmal one, Eq. 5.
- No possibility to identify, i.e. isolate, the deviating variable $\hat{\varepsilon}$ in the input vector. Therefore, no aid for further diagnostics.

3 Anomaly Isolation

To maintain the unsupervised paradigm and at the same time isolate the abnormal variable, it is necessary to cluster each dimension by a SOM in a separate domain (subgrid). Each domain is assigned to one variable. The nodes from the subgrids must be related together or, in other words, linked, to maintain the description of high dimensionality. The suggested linking is based on the probability of the nodes matching events along with all dimensions. The coupling takes the form of an additional layer that can be used to learn the probability of the nodes matching events, BMU Hits, and the joint probability of the hit events in between the subgrids.

The proposed training is divided into two stages, Fig. 2.

Stage1: The conventional Kohonen algorithm is used to train the 1D SOMs either in batch or sequential training with the target to lower Qe below a predefined limit $Qe_{max,i}$. The limit definition differs according to the physical relevance of each variable and the target clustering accuracy.

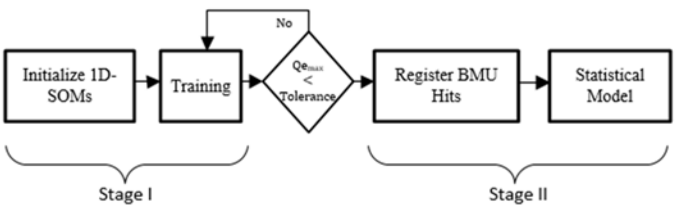

Fig. 2. Training stages

Stage2: The 1D SOMs are used to emit the BMUs by reusing similar training data set as in stage 1. For each observation vector, a vector of BMUs is emitted and registered in a matrix Ω for the rest of the training data set. Ω is then used to estimate the joint and conditional probability. For arbitrary variable j, the hits of BMUs can be resolved as a random process having discrete events

$$(SOM_j\ Hits) \in \{BMU_{j,1}, BMU_{j,2}, \dots BMU_{j,k}\}$$

As k is the number of the BMUs of trained SOM_j. The probabilistic space of that process is therefore abstract and, within which, the events are mutually exclusive since only one outcomes and no overlapping is permitted [14]. Furthermore, the hits are exhaustive since their probabilities cover the whole probabilistic space. Let A denotes the $BMU_{j,i}$ hit event from SOM_j and B denotes the corresponding hit event $BMU_{h,l}$ for an adjacent SOM_h.The conditional probabilities of A given B, where $P(B) \neq 0, P(A) \neq 0$, is given in [14] by

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A)P(B|A)}{P(B)} \quad (6)$$

The term $P(A \cap B)$ is the joint probability that both events are happening. If either $P(B)$ or $P(A) = 0$, then the events are described to be independent [14]. If both probabilities are nonzero then t,hey reveal independencies if one of the conditions is satisfied

$P(A|B) = P(A)$, or $P(B|A) = P(B)$

In this case, the following equation holds

$$P(A \cap B) = P(A)P(B) \quad (7)$$

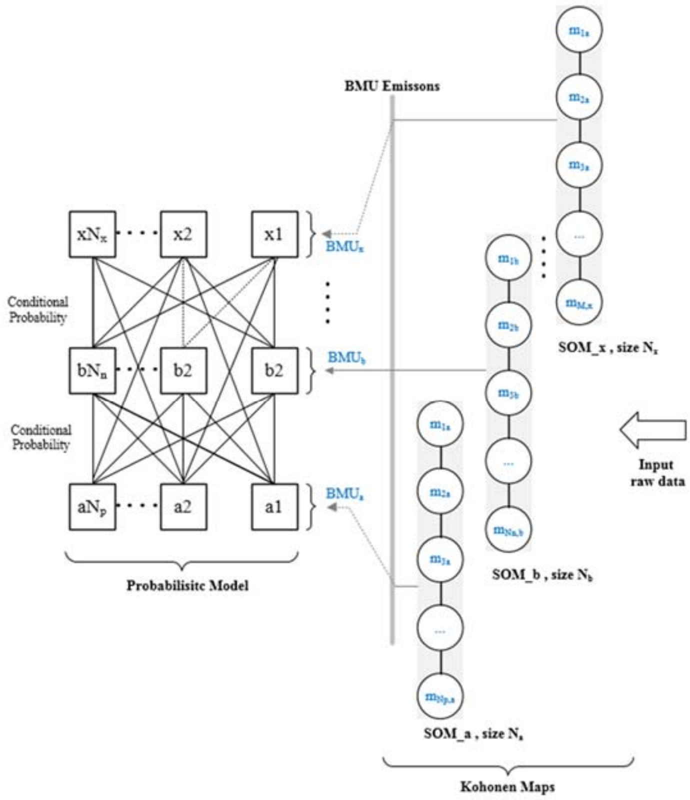

Fig. 3. Structure of the abnormality isolator

The probability of hitting certain groups of SOM nodes, each in a specific domain, is dependent on the collected data from the operation of the physical system. Non-hit grid nodes would have a negligibly low hit probability and therefore reveal statistical independence. We can conclude that the statistical relationships have physical relevance and can be therefore potentially used for CM. The relationships can be represented graphically as in Bayesian Networks (BN) or in abstract form, using Markov chains that can effectively determine the most likely sequence of numeric values, provided that the BMU events are introduced in sequential vectors of samples [4].

Fig. 3 depicts the proposed structure of the Abnormality Isolator (AbIso). A finite number of 1D SOMs on the right-hand side is intended to cluster each variable. The number of the grids and the input dimension is therefore identical. The most suitable topology of the grids in this case, is straight lines with lattice in form of rectangles.

N_i is the number of nodes (clusters) in each grid. It is not necessary that all subgrids have the same size. The subgrids can be initialized either randomly or linearly as the min and max limit can be estimated from the training data set or the operational limits. The probabilistic linkages are initialized by zero. Regarding the CM problem, the variables‘ values are physically bounded to the performance nominal and maximum limitations so that the usual procedure for determining N_i heuristically can be improved.

The parameters for the AbIso are listed as:

C : T raining length to converge
- Max number of training epochs for batch training.
- Max number of samples for sequential training.

$Qe_{i,max}$: Allowed clustering error (tolerance)

α: Learning rate

h_{ci} : Neighbourhood function, the Gaussian function is selected by default

R : Neighbourhood radius in number of grid nodes (= 1 by default)

Only the nodes whose hit probabilities as BMU are nonzero are registered in Ω, each variable occupies a column, and each observation, from the raw data feed, results in only one row. At the end of the training, the conditional probability can be estimated based on Ω, Fig. 3. Each BMU from the subgrids possesses the properties:

- The event hit probability for itself
- A finite number of connections to adjacent nodes in other layers

The connections are weighed by the estimated conditional probability for later utilization in CM modules. For isolation purposes, this step is established pairwise, so that the most likely chain combinations of BMUs are drawn and otherwise is assumed to be abnormal.

The classical SOM nodes represent an integral model of the data [7] in contrast to the procedure followed here. An operational point is represented by a prototype that consists of a set of BMUs from subdomains and statistical weighting of the hits. Therefore, the AbIso enables the anomaly detections in 2 levels of monitoring:

- Level 1, Values local ranges: Qe exceeds the grid training error. Each subgrid enables “out of range” detections of the variable, ε_i that it clusters or the clustering node has a low hit probability of being a BMU as trained.
- Level 2, Relational mismatch: The hit BMU does not correspond to subsequent nodes in the adjacent grids. This enables estimating the irregularity in operation even if all single variables values are drawn as normal (output from level 1).

4 Case Study

The concept is applied for the problem of detecting wear fault in hydrostatic pumps as a result of degradation effects.

4.1 Speed Variable Pump Drives

Speed variable pumping drives (SvPs) were recently developed essentially to improve energy efficiency in hydraulic systems [15]. These drives operate to control the pressure in a closed loop control scheme, Fig. 4. The controller varies the speed set point of an electro servo motor that drives a hydrostatic pump. The pump influences the output flow rate and, in turn, lower or increase the system pressure [18]. SvPs are used typically for pressure control tasks instead of servo valves. The system pressure is taken as the feedback control state, and the speed would be the controller output variable. This type of drives is used in many modern industrial applications such as plastic injection & moulding machines (IMMs).

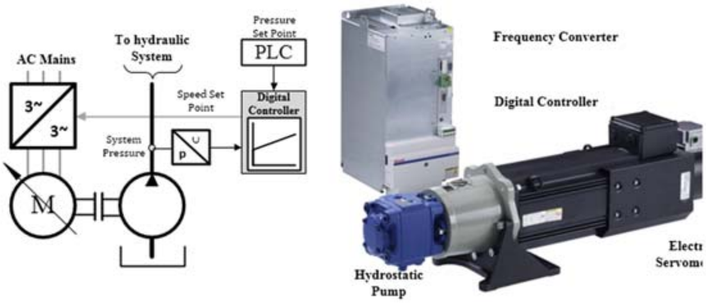

Fig. 4. Typical SvP drive

The three main variables to define an operational point are system pressure p, motor speed n and torque M. The pressure set points / traces differ according to the rest of the hydraulic system and the operating process. In addition, the conditions of the fluid varies sharply depending on the environmental effects such as temperature.
An essential aspect for CM tasks is the natural variance in the state variables depending on the environmental effects besides the loading conditions. This issue makes the threshold definition hard to set and predict. Fig. 5 depicts SvP operation points at typical IMM cycle [18]. The graphs show data from healthy operation in different environmental conditions.

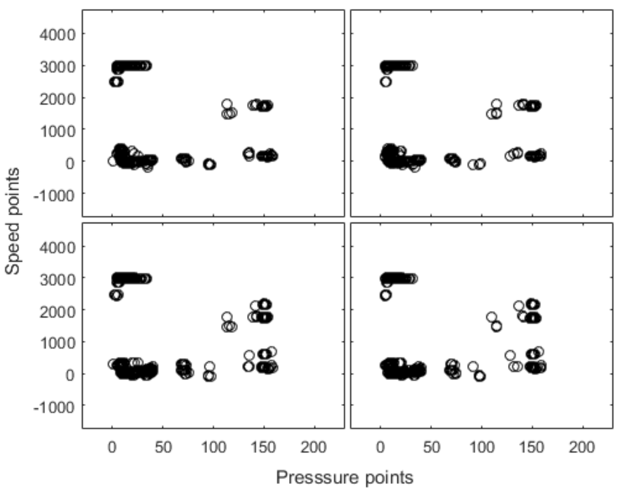

Fig. 5. $p - n$ operational points of SvP, steady states

4.2 Applying the Monitoring Model

The state variables reveal discontinuous domains that can be directly modelled by the 1D SOMs. The number of the nodes are defined normally in heuristic manner [12]. We propose to select the number of nodes in relationship to the nominal limits of the variables, by the equation :

$$SOM\ Size = \gamma\ (Nominal\ limit\ in\ the\ dimension\ data) / (tolerance\ per\ cluster) \quad (8)$$

For this example case, the nominal values and the tolerances are set outgoing from the standards in [8] as follows in Table 1.

TABLE 1 Variables Tolerances

Variable	Nominal limits	Tolerance [%]	Tolerance $Qe_{i,max}$	Approx. No. of Nodes, $\gamma = 0.2$
Pressure	315 [bar]	± 1	3.15	20
Speed	3000 [rpm]	± 1 :1.5	30	20
Torque	100 [Nm]	± 1	1	20

The software tool in [10] is used for training the SOMs. The data are collected for 3000 IMM cycles of operation and the training proceeds in batch form. The training parameters are summarised in Table 2.

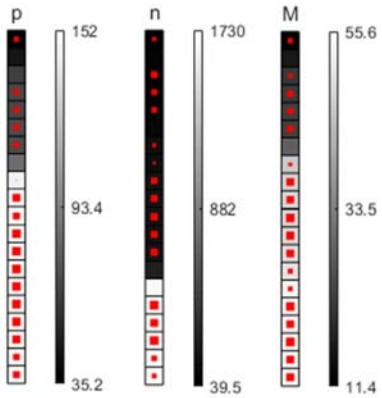

Fig. 6. SOMs for the state variables n, p, M

The resulted SOMs are represented graphically in Fig. 6. where the size of the red points signalises the hit probability.

TABLE 2 SOMs Design and Training Results

Item	SOM_p	SOM_n	SOM_M
State Variable	Pressure [bar]	Speed [rpm]	Torque [Nm]
Input dimension	1	1	1
Map grid size	20 x 1	20 x 1	20 x 1
Lattice type (rect/ hexa)	rect	rect	rect

Shape (sheet/cyl/toroid)	sheet	sheet	sheet
Neighbourhood type	Gaussian	Gaussian	Gaussian
Mask	-	-	-
Training status	initialized, trained 15 times	initialized, trained 25 times	initialized, trained 30 times
Average Qe	0.0385	0.1589	0.2655
Qe_{max}	1.3354	6.1622	1.5996

In Fig. 7 the statistical interrelationships are drawn, the size of the squares is proportional to the joint probability that both indexed nodes in the SOMs are hit as BMUs. The third graph of the nodes of speed and torque SOMs is not included in the example as it has no add-value for the target fault detection.

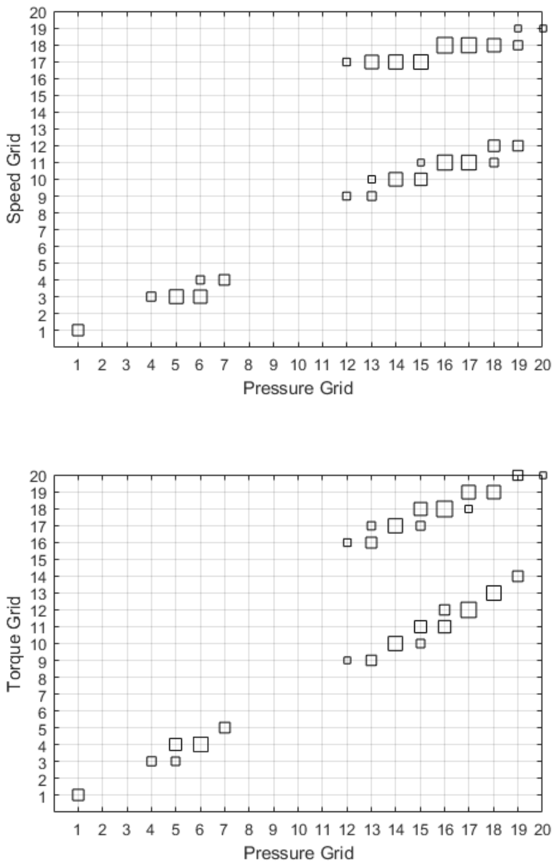

Fig. 7. BMU probabilities (p-n , p-M), 20 nodes

4.3 Degradation Fault Scenario

The AbIso is trained as explained in the last sections to learn the normal fault-free distributions of the data of SvP. The most sensitive component to fluid conditions and loading is the pump, whose volumetric efficiency decreases as a result of degradation wear. The fault can be simulated on a test rig by inserting an external leakage. Form the systematic point of view; the additional leakage acts as a disturbance on the control loops so that the speed demand to maintain the same pressure levels differs correspondingly. See Fig. 8 for the traces of the speed at a pressure set point = 100 [bar].

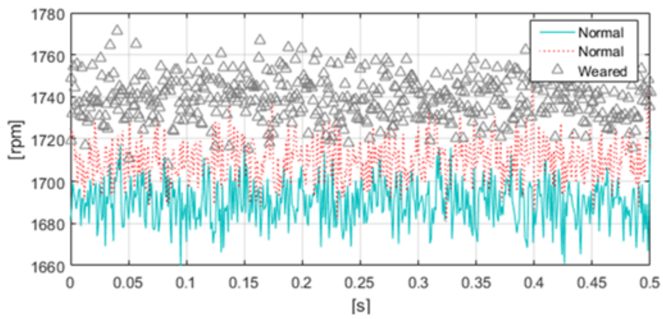

Fig. 8. Speed traces in normal and faulty operation, 100 [bar]

The detection is successfully done in the monitoring level 1, Qe of the BMU in SOM_n increases remarkably, Table 3.

Item	**Normal**	**Faulty**
SOM_p , Node	6	6
$SOM\_p\ Qe$	1.151	1.063
SOM_n, Node	3	3
$SOM\_n\ Qe$	4.278	**7.873**

TABLE 3 Comparison for Fault Detections

Although the detection in level 1 is sufficient for

- Input vector : $x(p, n, M)$
- Calculate the array of BMUs, $A = [S\_p, S\_n, S\_M]$ in SOM_p, SOM_n, SOM_M
- Register the respective Qe_{1x3} array

_________Level 1 Monitoring __________

- For i = 1: 3 do
  - If $Qe(i) > Qemax(i)$ then
    - Abnormality detected in domain i {pressure, speed, torque}
  - End_if
- End_for

_________ Level 2 Monitoring __________

- If $S\_p \notin SOM\_P_{BMUs}$ then
  - Abnormal pressure level
- Else
  - Extract k SOM_n_{BMUs} where $P(k|S\_p) > 0$
- If $k = empty$ then
  - No match → Abnormal speed point
- Else
  - Extract $M1$ SOM_M_{BMUs} where $P(M1|S_{\_p}) > 0$
  - Extract $M2$ SOM_M_{BMUs} where $P(M2|S_{\_n}) > 0$
  - If $M1 \cap M2 = empty$ (no common elements)
    - No match → Abnormal torque demand
  - End_if
- End_ if
- End_if

this case, a demonstration of level 2 is helpful to explore the working principle of the whole procedure. The statistical relationship can be calculated pairwise from the hits probability as outlined before. We construct the model flow on the basis of the physical dependencies of the variables. A key variable is the control state, system pressure p, and the speed demand is related to it. The motor torque is dependent on both. The sequence of checking the combinations starts from estimating the BMUs of the three subgrids. Assume the training output is SOM_p, SOM_n, SOM_M, Qe_{max} vector , the statistical relationships between the BMUs. The monitoring algorithm is outlined as follows:

5 Automatic Identification of Steady States

In the controlled operation, the process reveals transient and steady states (SS) of the controlled variable. An additional module to automatically distinguish the steady portions in the measurements aims to simplify and localise the CM functions. The automatic identification of the steady states is the core point in the research works [3], [6], [9] in different degrees of complexity. The difficulties here lay in the parametrization and the capability to run online. The straightforward concept of observing the control difference (command – actual state) is not suitable as for cases such as SvP; the drive may examine speed or torque limitations. In this work, an online steady state identifier (OSSI) is designed to simplify the identification using only the actual state signal. The basic idea is based on the convergence of the statistical variance around the moving mean of the discrete values stream.

The moving mean filtering for a discrete signal $x(t)$ is written as follows

$$\bar{x} = \frac{1}{K}\sum_{i=0}^{K-1} x(t) z^{-i} \qquad (10)$$

Moreover, the moving variance $\tilde{x}$ would be

$$\tilde{x} = \frac{1}{K}\sum_{i=0}^{K-1} (x(t)-\bar{x})^2 z^{-i} \qquad (11)$$

The steady state can be therefore identified using the tolerances in Table I as a threshold λ

$$SS = \frac{|\bar{x}-\tilde{x}|}{|\bar{x}|}\% < \lambda \ , \bar{x} \neq 0 \qquad (12)$$

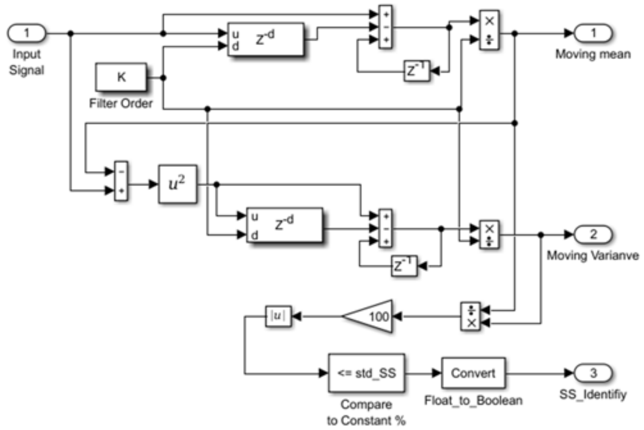

Fig. 9. Simulink discrete model for the OSSI

A suggested realisation is depicted in Fig. 9. This design requires only two parameters, k: Filter order, and λ: Steady state tolerance (= std_SS in the Simulink model).

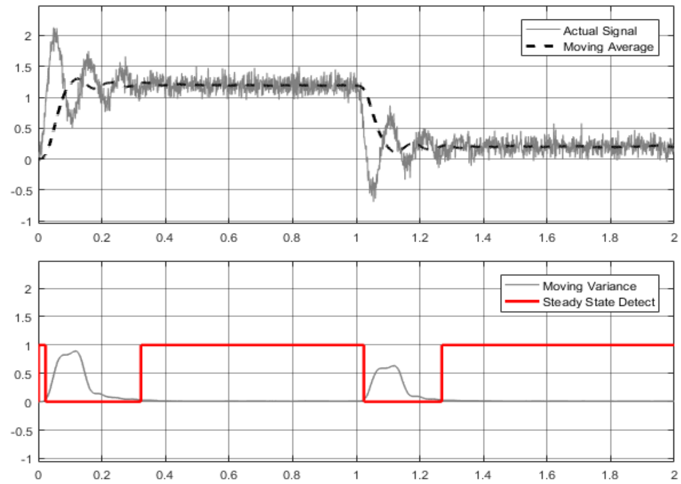

Fig. 10. Steady state identification for a 2nd order system

The value of k is the key parameter of the damping of fluctuations in the OSSI. Throughout the study, the value k =100 produces acceptable results. Fig. 10 depicts a test case of operation with a 2nd order stable system. The theory of operation of the OSSI encounters time delays and impulses during initialization, but for the aim of CM, the functions intend to extract long temporal portions; the following step would be features generation of the extracted portions, Fig. 8. That can in turn be used for the ML model.

6 Discussion and Outlook

The paper represented a novel structure for anomaly isolations. The structure, AbIso, is designed on the basis of Kohonen algorithm and it reveals practical impacts by a high degree of flexibility and handling simplicity. It can be practically used as a generalised framework for CM and troubleshooting in many systems. The structure combines one-dimensional SOMs (subgrids) and relates their output statistically. For linking the BMU hit events from each subgrid, the fundamental rules of probability were sufficient as there are no current requirements to infer facts or causalities.

The traditional application of SOMs is to cluster multi-dimensional data into 2D gird and not 1D as the contribution here. There are many other algorithms for 1D clustering. Gaussian mixture models and the radial basis ANNs can be used to learn continuous 1D data distributions. In the case of 1D discrete clustering, histogram algorithms are popular and efficient. SOM algorithm is chosen as it is suitable for discrete distributions and it does not require an exact predefinition of the number of clusters, besides the capability to perform sequential training. The sequential way of learning saves resources and computational load. Furthermore, the common task of normalizing each variable in the input space is no longer required. For multi-dimensional SOMs, normalization should not be omitted. Otherwise, input variables with high-value ranges would dominate the clustering nodes so that other low valued variables may have no influence on the clustering results.

The AbIso detects the anomaly in two levels by first comparing the quantisation error to the normal ranges in each subgrid, and then checks the BMU combination plausibility in level 2. There is no expected restriction on the number of input variables other than the increasing computational complexity, especially in the side of the statistical model.

An example case of degradation fault in speed variable hydrostatic pumps is demonstrated. The task used only three variables for CM and aimed to recognise the anomaly automatically. The isolated anomaly can be generally combined with expert knowledge so that a meaningful diagnosis is gained at the end. In this example, an excessive speed demand is detected because of degradation wear in the hydraulic pump. Other cases and applications in different fields and scales are left for future work.

Monitoring the features of the transient portions in signals may lead to undesired alarms. An automatic procedure to detect the steady states in the input signals is designed to switch the monitoring on/off and therefore; enhance the handling and support the concept of the low supervision demands. A simplified and practical method is introduced based on the online estimation of the statistical variance.

Bayesian Networks (BNs) are a known technique for visualizing complex statistical structures of random variables and extract causality for reasoning purposes. These Networks are widely applied in the field of reliability and diagnosis [21] as the BNs permit the integration of human knowledge in addition to the dependencies extracted from the observations in the datasets [4]. A suggestion for future works is to model the anomaly occurrences in temporal sequence and study their relationships to machine component failures in a global Bayesian diagnostics approach. The benefit beyond this is to gain a basis for fault predictions and therefore a ML model for the aim of adapting maintenance plans.

Acknowledgement:

The authors are grateful to the German Federal Ministry of Education and Research (BMBF) for funding the presented experimental work. The project ‘MultiCloud’ has funding code 02K14A063. Responsibility for the contents of this publication rests with the authors.

References:

[1] Aldrich, C. and Auret, L. 2013. *Unsupervised process monitoring and fault diagnosis with machine learning methods*. Advances in computer vision and pattern recognition, 2191-6586.

[2] Alhajj, R. S. 2014. *Encyclopedia of social network analysis and mining*. Springer, New York.

[3] Cao, S. and Rhinehart, R. R. 1995. An efficient method for on-line identification of steady state. *Journal of Process Control* 5, 6, 363–374.

[4] Du, K.-L. And Swamy, M. N. S. 2014. *Neural networks and statistical learning*. Springer, London.

[5] El-Betar A. et al. 2006. Fault Diagnosis of a Hydraulic Power System Using an Artificial Neural Network. *JKAU: Eng. Sci.* 17, 117–137.

[6] Fernandes G. et al., Ed. 2012. *Automated Method for Recognizing Steady State Intervals on The Series in Gas Turbines Test Bench* 1, Spain.

[7] Isermann, R. and Balle, P. 1997. Trends in the Application of Model-Based Fault Detection

and Diagnosis of Technical-processes. *Control Eng. Practice* 5, 709–719.
[8] Ivantysyn, J. and Ivantysynova, M. 2003. *Hydrostatic pumps and motors. Principles, design, performance, modelling, analysis, control and testing / Jaroslav Ivantysyn and Monika Ivantysynova*. Tech Books International, New Delhi.
[9] Jiang, T. e. a. 2003. Application of steady-state detection method based on wavelet transform. *Computers & Chemical Engineering* 27, 4, 569–578.
[10] Kohonen, T. *SOM Toolbox 2.0*. http://www.cis.hut.fi/projects/somtoolbox/. Accessed 15 May 2018.
[11] Kohonen, T. 1998. The self-organizing map. *Neurocomputing* 21, 1-3, 1–6.
[12] Kohonen, T. 2001. *Self-organizing maps*. Springer series in information sciences, 0720-678X 30. Springer, Berlin, London.
[13] Kohonen, T. and Honkela, T. 2007. *Kohonen network*.
[14] Navidi, W. C. 2011. *Statistics for engineers and scientists*. McGraw-Hill, New York.
[15] Neubert, T. 2002. *Untersuchungen von drehzahlveränderbaren Pumpen*. Dissertation.
[16] Penman, J. 1994. Feasibility of using unsupervised learning, artificial neural networks for the condition monitoring of electrical machines. *IEE Proc., Electr. Power Appl.* 141, 6, 317.
[17] Ramdén, T. 1998. *Condition monitoring and fault diagnosis of fluid power systems. Approaches with neural networks and parameter identification / Teresia Ramdén*. doctoral.
[18] Ristic, M. 2008. Development of Variable-speed Drives. *IFK*.
[19] Saxena, A. e. a. 2017. A review of clustering techniques and developments. *Neurocomputing* 267, 664–681.
[20] Torikka T. 2010. *Bewertung von Analyseverfahren zur Zustandsüberwachung einer Axialkolbenpumpe*. doctoral, RWTH Aachen.
[21] Zhou, D. 2014. *The Application of Bayesian Networks in System Reliability*. Master, ARIZONA STATE UNIVERSITY.