

Monitoring of Complex Industrial Processes based on Self-Organizing Maps and Watershed Transformations

Christian W. Frey, Fraunhofer Institute IOSB, Fraunhoferstrasse 1, 76131 Karlsruhe, Germany
Tel.: +49-721-6091-332, Fax: +49-721-6091-413, E-Mail: christian.frey@iosb.fraunhofer.de

Abstract— An efficient operation of complex industrial processes requires the continuous diagnosis of the asset functionality. The early detection of potential failures and malfunctions, the identification and localization of present or impending component failures and, in particular, the monitoring of the underlying physical process behaviour is of crucial importance for a cost-effective operation of complex industry assets. With respect to these suppositions a monitoring concept based on machine learning methods has been developed, which allows an integrated and continuous diagnosis of the physical process behavior and phases. The present paper outlines briefly the architecture of the developed distributed diagnostic concept and presents in detail the developed approach for the identification of intrinsic process-phases and the monitoring functionality of the unknown process behaviour based on self-organizing-maps and watershed transformations.

I. MOTIVATION

TECHNOLOGICAL progress in process engineering presents great challenges for system operators and especially for implementation and maintenance personnel. With respect to flexibility, functions which have hitherto been performed by mechanical or electromechanical devices are being increasingly replaced by software-based mechatronic systems. Due to the decentralization of intelligence and the transition to distributed automation architectures, the engineering, implementation and maintenance of these systems is becoming ever more complex. The associated costs meanwhile exceed the basic hardware costs of the components by a large factor. Effective diagnostic concepts offer great potential for cost savings - verification that a system is functioning correctly, fault diagnosis, early detection of impending component failures and, in particular, monitoring of the physical process behaviour is of crucial importance for the effective operation of the asset. The implementation of a diagnosis system e.g. for process-engineering applications usually requires a comprehensive development effort, calling on detailed expert knowledge about the physical interactions involved in the process. With respect to varying products and successive modifications of production process itself usually an on-going maintenance of the diagnosis concept itself has to be performed.

With respect to these suppositions a hierarchical diagnosis

and monitoring concept for industrial processes has been developed, which allows an integrated and continuous diagnosis of the communication network and the underlying physical process behavior. The present paper outlines briefly the architecture of the developed distributed diagnostic concept and presents in detail the developed approach for the identification of intrinsic process-phases and the monitoring functionality of the unknown process behaviour.

II. DIAGNOSTIC CONCEPT BASED ON SOFTWARE AGENTS

The field bus forms the "central nervous system" of distributed automation systems. With respect to process diagnosis the field bus itself subject of the diagnosis functionality as well as access point for the diagnosis concept to retrieve information to monitor the system components and physical behaviour of the process.

The diagnosis of technical systems can be understood as a two-stage hierarchical process, in which quantitative information in the form of sensor signals is transformed into qualitative information, that is to say, diagnostic results [1]. In the first stage, the feature generation, the objective is to convert the measurable state variables of the system by transformation into a suitable compressed representation format, such that the possible diagnostic results are reliably reflected. The second step of the diagnostic process, the feature evaluation, represents a logical decision-making process in which the compressed quantitative knowledge in the form of features is transformed into qualitative diagnostic knowledge.

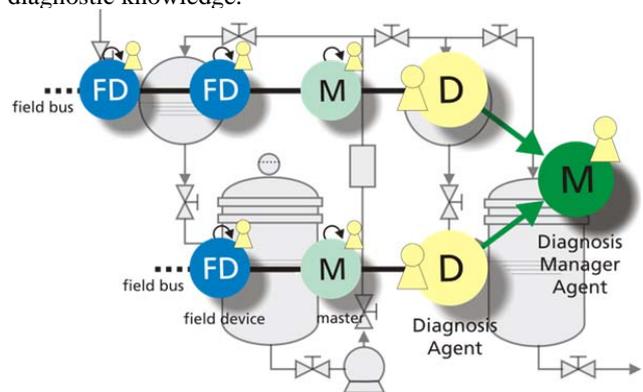


Fig. 1. Schematic overview of the developed software agent-based diagnostic concept for field bus based automation networks.

The developed diagnostic concept, illustrated in Fig. 1, is based on the idea of continuously analysing the messages transmitted on the field bus and extracting characteristic features, which describe in compressed form the behaviour of the respective field device and its interaction with the overall automation process. Features can be generated from the field bus messages concerning both the communication behaviour, for example, by analysing the response time of a field device, and the actual physical process behaviour, for example, by analysing a manipulated variable or a measurement value. The features represent a compressed description of the system functionality on the communications as well as on the process levels, which in turn can be linked together by a logical decision-making process in the higher-level feature-analysis stage to form an integrated diagnostic result.

The permanent analysis of the field bus data stream requires an effective compression of the data transmitted over the field bus, as otherwise the quantity of data produced would no longer be manageable. As an example of this, in a automation network with 12 Mbit/s transmission speed, as much as 1 MByte of data is transmitted per second – in Ethernet-based systems with transmission rates of up to one Gbit/s this can be over 100 MByte per second. It is clear that under these boundary conditions even a selective sampling of field bus data messages with subsequent off-line analysis would fail due to the non-availability of the requisite levels of computing capacity or storage media. The generation of meaningful features based on the immense data stream coming from the field bus places the highest demands on the efficiency and intelligence of the software design that is used. When considering the design of the diagnostic system, and especially in view of the heterogeneous communications networks or the different software platforms deployed on the control level in an automation system, particular attention must be paid to the effective coordination and cooperation of the distributed software components. Taking account of these requirements, a hierarchical diagnosis and monitoring concept for field bus based systems has been developed, based on software agents.

The diagnostic agents, in the lowest level of the hierarchy, are software components which as far as possible run autonomously on dedicated embedded systems, e.g. so-called field bus gateways. The task of the various diagnostic agents is to analyse the field bus messages of the field devices assigned to the section of the field bus and to generate meaningful features. The higher-level diagnosis manager is also implemented as a software agent and contains the actual diagnostic functionality. Implemented as a separate executable application or as a component of the host system, the diagnosis manager cyclically retrieves the features generated in the diagnostic agents of the respective strands of the field bus and passes them to the feature analysis stage.

With regard to integration into existing automation systems, the newly developed concept provides additional software agents, for example, for linking to different physical implementations of the field bus (Profibus, Foundation Fieldbus), for displaying the diagnosis results

(GUIAgent) or for archiving them (ArchiveAgent).

There are various development environments available for building and implementing multi-agent systems, which provide the developer with basic functions for deploying software agents. For the implementation of the proposed diagnostic concept the JADE development environment (Java Agent Development Framework) was selected [2,3].

III. DIAGNOSIS AND MONITORING OF PROCESS BEHAVIOUR

When considering an integrated diagnosis of industrial processes, the field bus, being the central communications network of the system components, represents both the object of the diagnosis in terms of communications technology, and at the same time it is an ideal access point for the diagnosis of the physical process behaviour of the underlying automation process. To guarantee diagnostic functionality that is as transparent and robust as possible, the concept that has been developed splits up the diagnosis task into the diagnosis of the communication behaviour of the field bus and the diagnosis or monitoring of the physical system behaviour of the underlying automation process.

To diagnose the communication behaviour, relevant features are generated from the field bus messages, such as the response time of a field device. These features are evaluated by frequency-based neuro-fuzzy membership functions and an expert knowledge base implemented in a fuzzy-rulebase (*if ... then ...*) [5]. A detailed presentation of this neuro-fuzzy based design is given in [3].

In contrast to the diagnosis of the communications layer, in which only signal-based feature generation methods are used, the diagnosis of the underlying physical process behaviour relies on model-based methods. The basic principle of model-based diagnosis for technical systems is based on a quantitative mathematical model of the process to be monitored. Using a suitable distance measure, the measured process variables are compared with the calculated process variables of the process model. The greater the distance between measured and modelled variables, the greater the probability of a deviation from the normal behaviour of the process assumed in the modelling step [1,6]. The difficulty in implementing such model-based diagnostic systems lies in generating a suitable process model, in particular if analytical modelling approaches are used. This means that, for example, in industrial process-engineering applications, setting up a robust model for feature generation often requires a comprehensive development effort, calling on detailed expert knowledge about the physical interactions involved in the process.

An alternative to the analytic model-based methods is offered by data-driven adaptive modelling approaches. In these approaches, the most prominent technique is that of artificial neural networks. On the basis of measured historical process variables (training data), neural networks can use a learning algorithm to acquire the static and/or dynamic transmission behaviour of the process [4,7]. The advantage of this approach is that no analytically formulated process model is needed *a priori*, and thus the developer is not hampered by "unsafe" assumptions about the physical

interactions of the process that are present when modelling. Frequently used methods in the so-called Black-Box Modelling approach are back-propagation networks. These neural structures are based on a supervised learning task, which requires a correct assignment of the available process variables to input and output variables of the process model.

In the context of the diagnosis of an unknown physical behaviour in field bus based automation systems, however, the following problem arises. While in principle all in- and output values of the physical sub-processes can be retrieved from the messages in the field bus data stream, the allocation in terms of input/output values itself requires a very detailed knowledge of the underlying system, which has usually to be obtained from the system documentation or expert knowledge. Based on these boundary conditions, the concept developed for the diagnosis of the unknown physical behaviour uses the properties of self-organizing maps for data-driven modelling of the process behaviour.

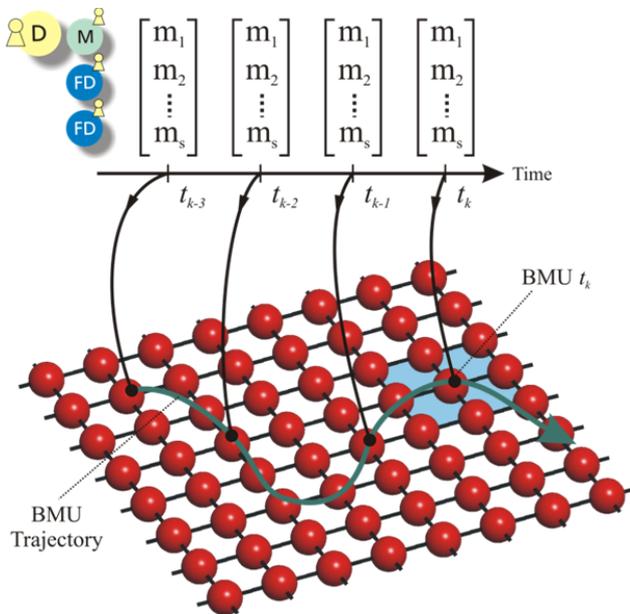


Fig. 2. Self-organizing map with a two-dimensional lattice topology.

The so-called self-organizing feature maps (SOM), a special neural network, are capable of generating a topology-preserving mapping of a high-dimensional feature space into an output space of lower dimensionality [8]. These neural models, also known as Kohonen maps, are capable of extracting and displaying unknown clusters in the database to be analysed (structure discovery), "unsupervised" without a priori information in terms of input/output assignment.

The basic components of SOMs are referred as neurons, but they differ from the neuronal model of the back-propagation networks in their basic function. The neurons do not function as processing units that respond to particular inputs with particular outputs, but assume the role of simple memory units, the content of which, known as stored pattern vectors or also prototypes, can be read out and rewritten with new data. The neurons of the SOMs, as shown in Fig. 2, are arranged in a specific topology, that is, the neurons are in a topological relationship to one another. This quality of the

network is generally referred as neighbourhood. Self-organizing maps are based on an unsupervised, concurrent learning process – during the learning process a feature vector M from the learning task is presented to the network and its distance from the prototype vectors W stored in the neurons is calculated. The neuron with the smallest distance to the input vector, the so-called Best Matching Unit (BMU) i , is determined. In the next step the prototype W of the BMU and its neighbours are adapted using the learning rule for a single neuron o :

$$W_o^{k+1} = W_o^k + \eta(o, i)\mu(M - W_o^k)$$

Beside the BMU ($o = i$), the Kohonen learning rule also includes neighbouring neurons in the learning process. The introduction of neighbourhood learning enables "similar" feature vectors M to be projected onto topographically similar regions of the map, respectively to neurons o . The decreasing function $\eta(o, i)$ of the map distance between the BMU i and the neighbour neuron o can be implemented as an Gaussian function such that $\eta(i, i) = 1$. A measure of how well the map represents a training dataset is provided by the so-called quantization error Q , which is calculated using, for example, the Euclidean distance between input feature vector M and prototype vector W of the winner neuron.

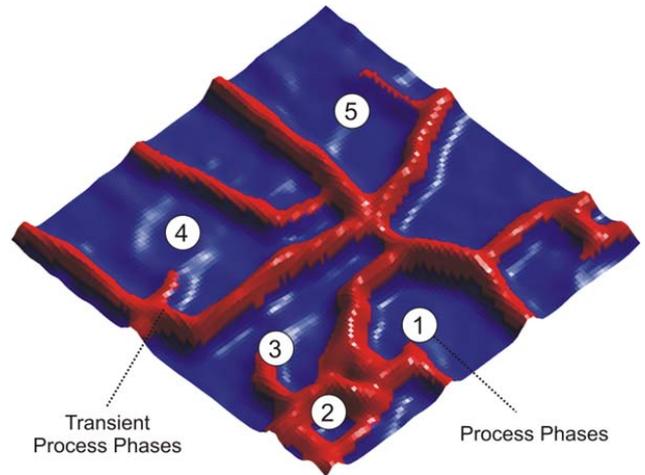


Fig. 3. UMatrix representation of on 2D SOM.

By applying the so-called UMatrix-representation of a self-organizing map, it is possible to perform a classification or a clustering of the feature space [7]. The UMatrix-transformation is based on the idea to add an additional dimension to the SOM lattice topology, which reflects the distance between the individual neuron to its surrounding neighbours. As an example of this, such an UMatrix representation of a 2D SOM is visualized in Fig. 3. The valleys in this UMatrix-plot (blue) represent regions in which the stored pattern vectors are very similar, the regions shown in red characterise a transition from one feature-space cluster to the next cluster.

In terms of the diagnosis and monitoring of a technical process, the valleys in the UMatrix-plot can be interpreted as stationary process phases, which are separated by so-called transient process phases. It should be noted here that the

topology of the SOM corresponds to a toroid: opposite edges of the UMatrix-plot are joined together, that is, valleys of the UMatrix can extend across the boundaries of the map to the opposite side.

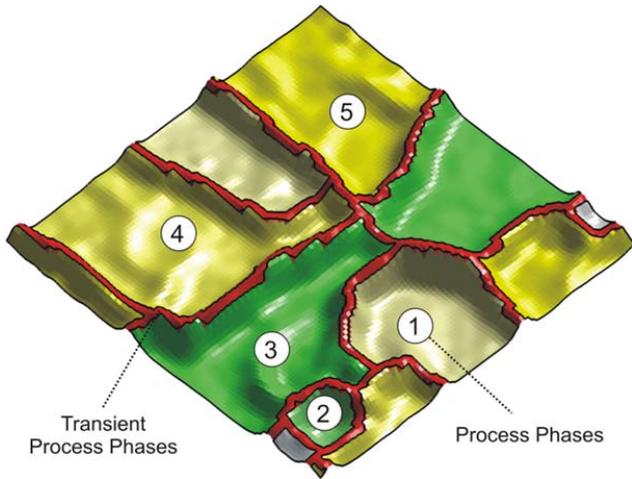


Fig. 4. Segmentation of an UMatrix into cluster regions by applying the Watershed transformation.

In principle, the segmentation or subdivision of a UMatrix into cluster regions can be performed manually by the observer. This method rapidly becomes impractical however with increasing size of the map or structural complexity of the UMatrix. For the automatic segmentation of the UMatrix, the diagnosis and monitoring concept we have developed uses the method of the Watershed transformation which is well-known from the image processing field [9,10,11]. The basic idea of the method can be made clear with the analogy of "drops of water" which fall on the UMatrix "mountains". Through the "mountain ridges" flooded regions are formed, which correspond to the clusters of the UMatrix. Fig. 4 shows the result of such a Watershed transformation – the UMatrix has been segmented into 6 different clusters, corresponding to 5 major process phases.

As already indicated, in the data stream of the field bus, or alternatively in the messages of the field devices, all process variables are in principle retrievable and can be extracted by the diagnostic agents without detailed expert knowledge. Depending on the complexity of the system under

consideration and the transmission speed of the field bus, the state variables of the process are summed over time and combined into an integrated state variable vector M (cf. Fig. 2). This high-dimensional state variable vector describes the interaction of all field devices with the overall system and constitutes the basis of the model-based diagnosis. In the initial training phase of the diagnostic concept the data-driven model of the system is generated in the form of a SOM using the feature vector. The duration of the training process depends essentially on the complexity of the process (number of field devices), or in the case of cyclical process behaviour, on the duration of the batch. The success of the learning process can be evaluated with the aid of the resulting quantisation error of the map. It is clear that during this data-driven procedure, as close as possible to all operating states of the system should be acquired in order to obtain a robust diagnostic performance. Based on this trained map, an online diagnosis and monitoring of the process behaviour can then be performed. By analysing the quantisation error of the map, or the progress of the process phases respectively the BMU-trajectory, deviations from the normal behaviour of the system can be detected and thus traced back to possible errors in the behaviour of the system.

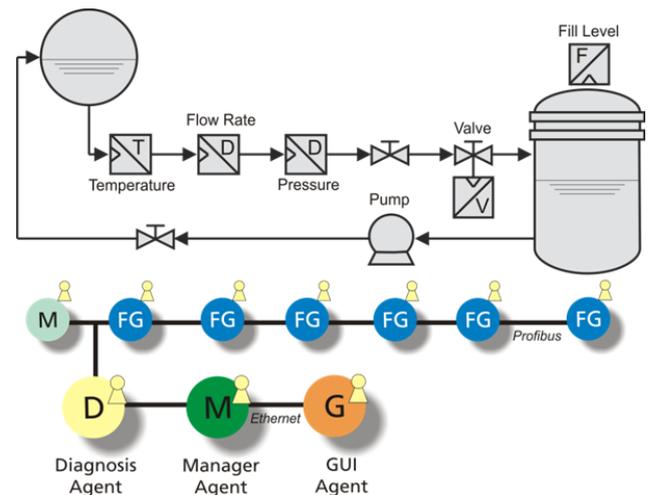


Fig. 5. Schematic overview of the installed demonstrator system.

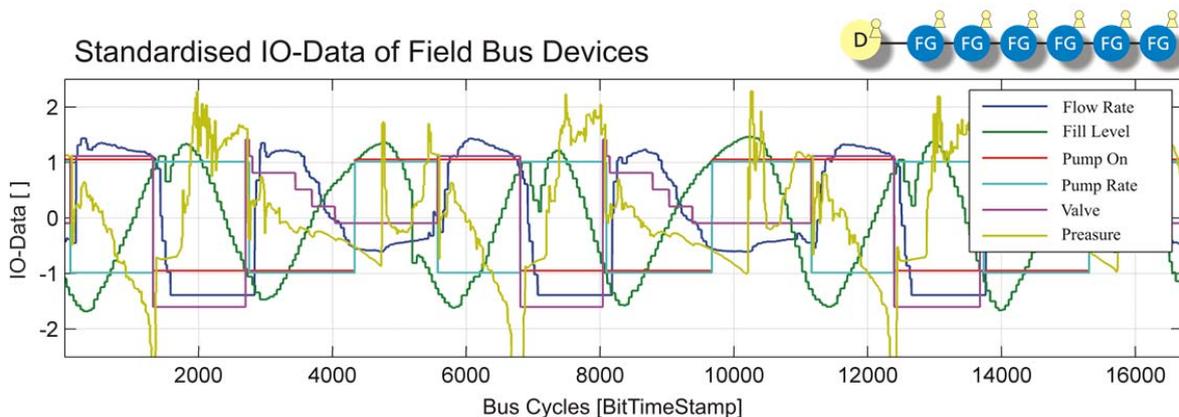


Fig. 6. Time course of the process state variables extracted from the field bus messages during normal operation.

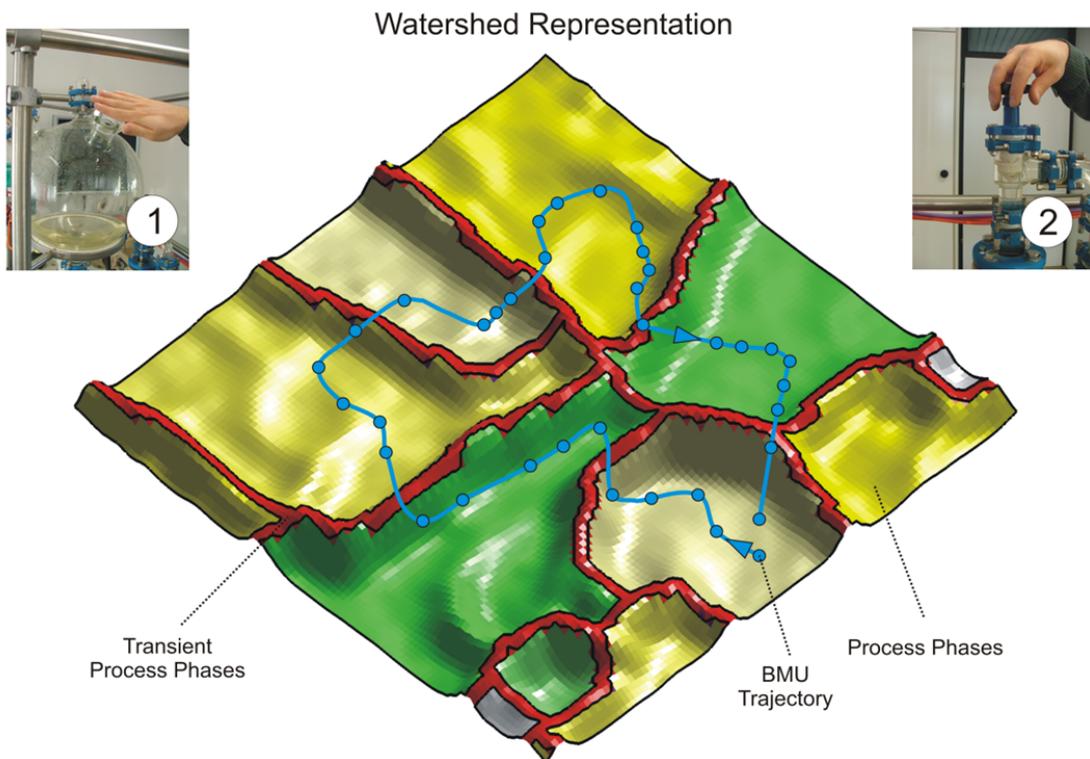
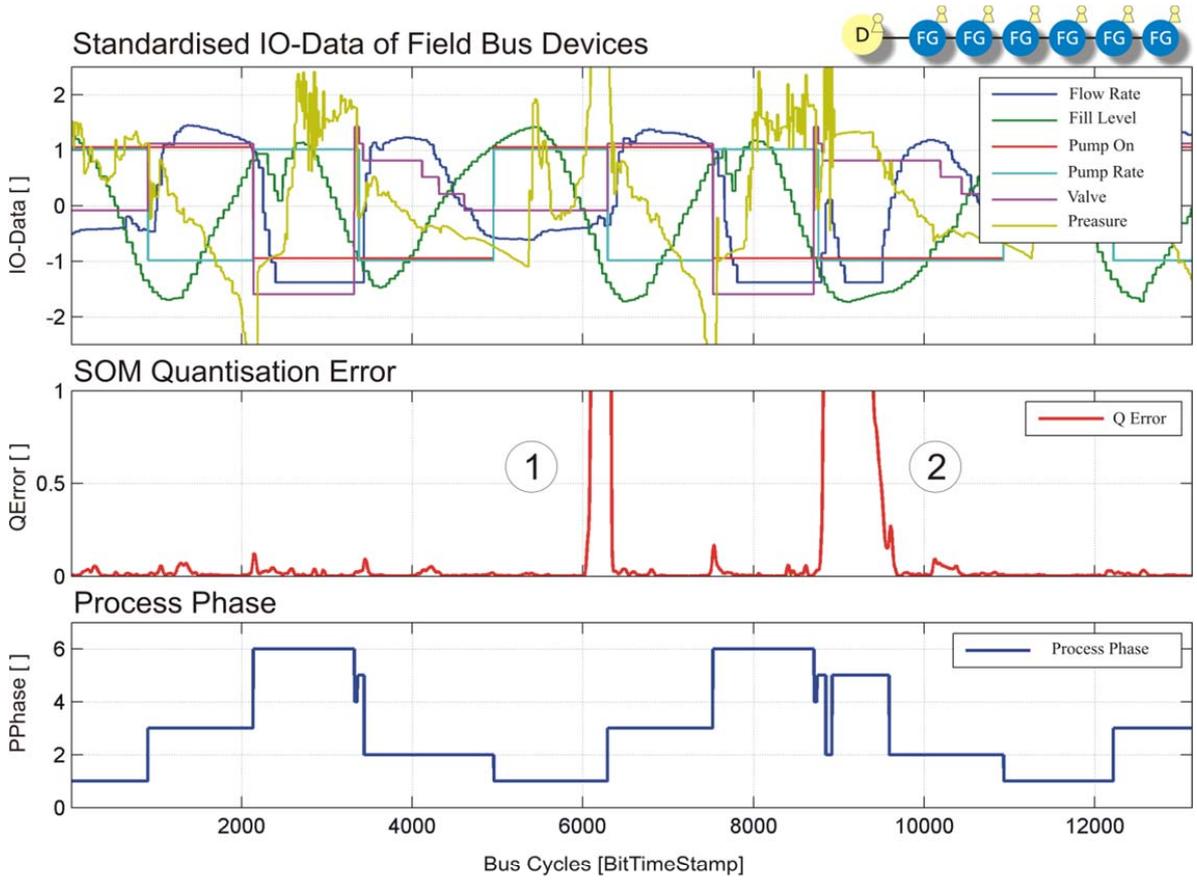


Fig. 7. Time course of state variables, quantisation error and process phase during abnormal process behaviour.

IV. TESTING THE DIAGNOSTIC CONCEPT

The experimental system, illustrated in Fig. 5, consists essentially of two containers between which liquid is pumped around in cycles at varying pumping powers and valve positions. It should be noted here that in the case of the demonstrator system the nature of the field devices (measured or manipulated variable), the physical meaning of the process variables acquired (e.g. flow rate meter, temperature sensor), and in principle the processing behaviour implemented by the control system are all known. The diagnostic concept developed however does not rely on this additional information.

Fig. 6 shows an example of the temporal course of the standardised process data obtained from the field bus data stream, or field bus messages as appropriate. From this curve the cyclically recurring behaviour of the process can be clearly recognised. On the basis of this training data, a data-driven model of the system in the form of a self-organizing map was generated. With the aid of the UMatrix transformation, the various process phases or operating states of the system (e.g. emptying containers or filling containers) were identified using the Watershed transformation. Based on this trained feature map an online-diagnosis of the process behaviour is then made. To do this, the IO-data of the field devices is filtered out online from the field bus data stream by the diagnosis agents and presented to the feature map. By analysis of the quantisation error of the map, deviations from the learned normal behaviour can be detected. As an example, Fig. 7 shows the behaviour of the quantisation error over the duration of the process during abnormal process behaviour. In error case 1 the venting of the system was reduced in order to disrupt the behaviour of the process, while in error case 2 the flow cross-section was reduced. These interventions in the process behaviour of the system are clearly revealed in the curve of the quantisation error of the map.

By observing the BMU trajectory in combination with the process phases found by watershed segmentation, it is also possible to analyse the progress of the process phases. Fig. 7 shows, in addition to the quantisation error, the time course of the identified process phase – the modified progress of the process phases in error case 2 can be clearly seen.

V. SUMMARY

The agent-based diagnostic concept that has been developed can be applied as an integral component of an automation system, and enables a continuous integrated diagnosis and monitoring of the process under consideration. The present paper has outlined the architecture of this distributed diagnostic concept based on software agents and explained its embedded functionality for diagnosing the unknown process behaviour using self-organizing-maps and watershed transformations. Based on historical or online process observations the data driven concept is capable of learning the unknown system behaviour without the need of a comprehensive expert knowledge about the physical interactions involved in the process.

The functionality and performance of the developed

diagnostic concept was validated with the aid of a process-engineering demonstrator system. The monitoring concept is transferred to an industrial application for a wide range of industrial process in chemical and pharmaceutical industry.

REFERENCES

- [1] Isermann, R.: *Model-based fault-detection and diagnosis – status and applications*. Annual Reviews in Control, vol. 29, iss. 1, pp. 71-85, 2005
- [2] Bradshaw, J.M.: *Software Agents*. AAAI Press, April 1997 ISBN 0-262-52234-9
- [3] Endl, H.: *FELDBUS - Erhöhung der Sicherheit und Zuverlässigkeit von eingebetteten, echtzeitfähigen Softwaresystemen durch die Entwicklung von Methoden und Werkzeugen zur agentenbasierten Diagnose von Feldbusnetzen in verteilten Automatisierungssystemen*. Schlussbericht BMBF Förderprojekt, Förderkennzeichen 01 IS C 16 A (Final Report of BMBF-funded project 01 IS C 16 A)
- [4] Vermasvuori, M.; Enden, P.; Haavisto, S.; Jamsa-Jouela, S.-L.: *The use of Kohonen self-organizing maps in process monitoring*. Intelligent Systems, 2002. Proceedings. 2002 First International IEEE Symposium, vol. 3.
- [5] Nauck, D.; Klawonn, F.; Kruse, R.: *Neuronale Netze und Fuzzy Systeme* Vieweg Verlag, Braunschweig/Wiesbaden 1994
- [6] Frey, C.; Kuntze, H.-B.: *A neuro-fuzzy supervisory control System for industrial batch processes*. IEEE Transactions on Fuzzy Systems 9 (2001), 4, S. 570-577
- [7] Haykin, S.: *Haykin, S.: Neural Networks and Learning Machines* 3rd Edition. Prentice Hall, Upper Saddle River, New Jersey 2008
- [8] Kohonen, T.: *Self-Organizing Maps*. 3. Auflage Springer, Berlin, 1995
- [9] Vincent, L.; Soille, P.: *Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulation*. IEEE Transaction on Pattern Analysis and Machine Intelligence 1991
- [10] Roerdink, J; Meijster, A.: *The Watershed Transform: Definitions, Algorithms and Parallelization Strategies*. Fundamenta Informaticae, Special issue on mathematical morphology Volume 41 Issue 1-2, 2000
- [11] Robinson, K.; Whelan, P.F.: *Efficient morphological reconstruction: A downhill filter*. Pattern Recognition Letters 25(15), 1759–1767, 2004