Computers and Chemical Engineering 95 (2016) 58–70

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

A diagnostic method based on clustering qualitative event sequences

A. Tóth [a,*], K.M. Hangos [b]

[a] Department of Electrical Engineering and Information Systems, University of Pannonia, Veszprém, Hungary
[b] Process Control Research Group, Computer and Automation Research Institute, Hungarian Academy of Sciences, Budapest, Hungary

ARTICLE INFO

Article history:
Received 11 March 2016
Received in revised form 15 August 2016
Accepted 1 September 2016
Available online 8 September 2016

Keywords:
Fault diagnostics
Qualitative diagnosis
Clustering
Tennessee Eastman process

ABSTRACT

A diagnostic algorithm is described in this article that is based on clustering qualitative event sequences called traces. A sufficient number of training traces are used instead of an internal model to specify the faulty models of the system. The diagnosis consists of two phases. In the off-line training phase diagnostic clusters representing nominal and faulty behavior are formed from the set of training traces, while the centroids of these clusters are stored. Arbitrary measured traces in the on-line diagnosis phase are compared with the centroids, to recognize the most probable faulty scenario for the trace. The effects of different mapping functions and different qualitative ranges on the clustering are investigated, and the diagnostic resolution of the method is compared and discussed using a simple process system. A diagnostic case study using the benchmark of Tennessee Eastman process (TEP) is utilized to illustrate the efficiency of the proposed method.

1. Introduction

Early and accurate fault diagnostics is one of the most important challenges during the operation of modern day process systems. Primeval fault mitigation and isolation due to proper diagnostics plays a crucial role in avoiding huge losses and plant breakdowns caused by the consequences of initially smaller and isolated but propagating failures discovered too late.

Due to the high importance of the field, the relevant literature is extensive with model-based diagnostic methods traditionally being the most widespread. Process fault diagnostics based on process and fault models had been widely described by Venkatasubramanian et al. (2003a,b,c) in review articles. According to Venkatasubramanian et al. (2003b), model based a priori knowledge can be broadly classified as quantitative and qualitative. Fault detection using these qualitative models can be performed by using expert systems with different kind of reasoning, using signed directed graphs (SDGs) for modeling cause-effect relations (for instance in Vedam and Venkatasubramanian, 1997) or fault trees describing the relations between primary events to top level events or hazards. Fault propagation analysis (Gabbar, 2007) can be also used for the identification of faults, causes and consequences in a systematic manner.

Qualitative physics is also used for process system modeling as a common sense reasoning about physical systems. This approach is based on qualitative or ordinary differential equations describing the process system to be diagnosed. These qualitative dynamic models together with many different methods (like the one in Tóth et al., 2014) use an abstract hierarchy of process knowledge which is based on decomposing the process system into subcomponents, in order to decrease computational complexity and speed up the diagnostics task.

The information collected by hazard identification can be also regarded as a special form of process models. An attempt to unite the diagnostic information stored in HAZOP and FMEA analysis results, called the blended HAZID methodology was described in Németh and Cameron (2013) together with its use for process system diagnosis tasks. This approach has been further extended in Guo and Kang (2015) using dynamic fault trees.

Fault diagnosis includes two sub-steps even in the most general case: fault detection and fault isolation or identification. While the first sub-step needs an accurate model of the process in its normal, i.e. non-faulty operation mode, fault models of the considered faulty modes are needed for the latter. Therefore, the most important aspect of a fault diagnostics algorithm for process systems is the fault model which requires significant amount of human expertise and work to set up and maintain. Our main aim in this article is to suggest a data-driven diagnostic procedure which may require less amount of human assistance as compared to a model-based approach during set-up and operation and still remains feasible as a fault diagnostic method. While a satisfactory model of a

* Corresponding author.
E-mail addresses: atezs82@gmail.com (A. Tóth), hangos@scl.sztaki.hu (K.M. Hangos).

http://dx.doi.org/10.1016/j.compchemeng.2016.09.001
0098-1354/

possibly complex process system in each of its considered faulty mode is needed that requires skilled human efforts, informative enough observed data set that are annotated with the recognized fault(s) by the plant operators may form the basis of a data-driven diagnostic procedure.

In the last review article of the series by Venkatasubramanian et al. (2003c) on process systems diagnosis, process history based methods are surveyed. Instead of an a priori model, these methods require a large amount of historical process data, and they can be classified by the way they extract information from the process data (this operation is called feature extraction). Feature extraction can be qualitative (for example using rule-based expert systems or qualitative trend analysis) and quantitative (using statistical methods, such as PCA or neural networks).

For describing arbitrary output signal values qualitative trend analysis (QTA) can be used, by comparing qualitative trends of nominal and actual signal values (a good example can be found in Maurya et al., 2005). In some newer results (in Maurya et al., 2007), these methods have been even combined to perform fault diagnosis.

A special type of historical process data are the so called alarms, the timed sequence of which has been utilized for early fault detection and diagnosis in Agudelo et al. (2013). These alarm sequences can be also regarded as event logs. In van der Aalst et al. (2007) a process mining tool called ProM is described which is capable of discovering process models in the form of Petri Nets, using event logs collected from process systems.

This tool also supports conformance checking, verification, model extension and transformation as well as model discovery. A ProM extension described in Alves de Medeiros et al. (2008) uses *K*-means clustering for categorizing event logs prior to mining them, in order to achieve faster operation. In a slightly different approach described in Rozinat et al. (2008), Petri nets are used to build up models from event sequences, and the fitness and appropriateness of the model is calculated.

In the approach described in this paper similar metrics to ProM are used to perform the validation (in the way the fitness of the model is calculated) after an initial training phase performed on the historical process data. As a technique used thoroughly in machine learning, clustering is widely used in systems used for process diagnosis. The algorithm described in this paper is based on the *K*-means clustering algorithm (described in Alpaydin, 2010b) like a modeling approach described in Alves de Medeiros et al. (2008). Different other approaches are using the fuzzy c-means clustering (FCM, described in Alpaydin, 1998), a method based on the concept of fuzzy sets and logic (described originally in Zadeh, 1975). For example, fuzzy c-means clustering for fault classification is reported in Mercurio et al. (2009) and Petković et al. (2012) while it is used for process control in Kim and Kim (2014).

The most widely used quantitative feature extraction procedures use statistical methods (e.g. PCA or PLS) for process monitoring and fault detection, for which good review papers have appeared recently, see Yin et al. (2012), Qin (2012) or MacGregor and Cinar (2012). A recent improvement of the PLS method capable of detecting small faults have been reported in Harrou et al. (2015). However, these methods usually assume steady-state operation condition of the system to be diagnosed, and fail during transient operations. This fact and the need for diagnosing process systems outside of their steady-state regime have motivated our research to overcome this constraint.

The structure of this paper is as follows. First, basic notions about qualitative event sequences (traces) and their representations are introduced. After that, the proposed diagnostic procedure is described in detail, finally the diagnostic capabilities of the algorithm are demonstrated using a simple and composite case study (the Tennessee Eastman Challenge Process).

2. Qualitative events, traces and their distances

In case of a process system working under transient conditions (i.e. it is not steady-state) its operation can be described as sequences of events. These events refer to the actual values of measured quantities of the system at specific times, such as the values of the *system inputs* including the possibly discrete valued (on/off or open/close) states of the actuator elements (for example pumps or valves) and the values of the *system outputs* which are the values of sensors (such as level or pressure sensors).

2.1. Events with qualitative range spaces

System inputs and outputs are signals, i.e. time-dependent quantities (as described in Hangos et al., 2004). Their range space can naturally be discrete (such as open or close for a two-state valve) or real (a positive real value for a pressure signal).

In case of uncertain values for a real valued measured signal, one can describe the actual value using a qualitative range space, which is a set of ordered mutually disjoint set of real intervals. The number and the actual end-point set of these intervals (i.e. the resolution of the qualitative range set) depend on the accuracy of the measured signals and on the desired accuracy of the diagnostic results. In order to be able to investigate the effect of the resolution on the diagnostic accuracy, we define and use two different qualitative range sets in this paper.

First we define a simple natural set of intervals that fits to positive valued signals, such as temperatures or levels. One may associate verbal *labels* to the intervals following the normal operational value of the signal as follows: "*N*" stands for the normal range, "0", "*L*" and "*H*" denote the empty, low and high but acceptable values (still inside normal ranges), respectively, while "*e*−" and "*e*+" refer to values which are outside nominal ranges, respectively. Formally, this basic *qualitative range set* is described in the following way:

$$Q = \{e-, 0, L, N, H, e+\} \tag{1}$$

It is possible to create a refined qualitative range set from the qualitative set Q in Eq. (1) by placing a new qualitative value between two already existing ones. Such *refined qualitative range set* is given below

$$Q_{refined} = \{e-, -0, 0, 0L, L, LN, N, NH, H, H+, e+\} \tag{2}$$

with the newly introduced labels "−0" small negative values, "0*L*" very low, "*LN*" a bit low, "*NH*" a bit high, "*H*+" very high.

One can further refine the qualitative range set by adding new intermediate values and achieve the range space of real values in the limit.

The range space of binary discrete valued signals, such as the status of a valve with two states, can be described by the range space

$$\mathcal{B} = \{0, 1\} \tag{3}$$

where "0" can be associated to the closed and "1" to the opened status.

The qualitative sets defined in Eqs. (1) and (2) can be also seen as a boundary case of a fuzzy set (as defined in Zadeh, 1975) which does not contain fuzziness, in this case every membership function has a constant value for a defined interval and those intervals does not overlap each other, like ordinary fuzzy sets do.

Events. An event $event_\tau$ associated to a signal or to a set of signals is an ordered pair of a time instance τ and the actual qualitative value(s) $x(\tau)$ at this time instance, i.e. $event_\tau^{(x)} = (\tau; x(\tau))$. Formally, the syntax of an input–output event (at time instant τ of an n-input m output system) is:

$$event_\tau = (\tau; input_1, \ldots, input_n; output_1, \ldots, output_m),$$

where the time τ is also discrete. τ is also called the *sequence number* of the event.

Examples of events in a system with a single two state input and a single real valued output from Eq. (1) are $event_1$ = (1;"0";"*N*") or $event_5$ = (5;"1";"0").

Ordered sequences formed from the events above are called *traces* and defined as:

$$T_{(t_1,t_n)} = event_{t_1}, \ldots, event_{t_n}$$

Events in the same trace always contain the same number of inputs and outputs with possibly different values, while τ is strictly monotonically increasing in consecutive events in the trace (traces are ordered by the time instants in the events). Note that τ can be unevenly spaced. In this case the difference between consecutive sequence numbers need to be the same in all considered traces. In this article we are dealing with events with evenly spaced τ values only.

L-*neighbourhood of a qualitative value*. Given a qualitative set Q, for every qualitative value $q \in Q$ the $neighbourhood(q, \ell)$ is defined as a set containing all elements from Q which, respecting the ordering of Q, are not farther than a given ℓ>0 natural number from q. The neighbourhood does not contain the element itself, $q \notin neighbourhood(q, n)$. The parameter ℓ is called the level of neighbourhood.

Given the qualitative set defined in Eq. (2) then

$$neighborhood(N, 2) = \{L, LN, NH, H\},$$

as these are the qualitative values not farther than 2-levels from "*N*" based on the considered qualitative set.

Measurement errors. Qualitative output values usually come from measurements which might be prone to measurement errors. These errors might be large enough (or the qualitative set can be fine enough) so the observed value does not match the actual value even in the considered qualitative range space (it will take on a neighboring value instead from the range).

Given a set of qualitative values Q, such as the ones described in Eq. (1) or Eq. (2), a function

$$ERRSIM(q, \ell, p) : Q \mapsto Q$$

can be defined to simulate the effect of such measurement error, transforming a qualitative value q to a qualitative value w from the neighbourhood (considering neighbourhood level ℓ) of the value q with a given probability p. This function is useful to simulate the effect of sporadic measurement errors in large number of training-input traces. Simulating these errors played an important role in evaluation of the different forms of the diagnostic approach in the simple case study.

2.2. Mapping of qualitative values to real ones

In order to be able to define distances between events and traces, one can convert the qualitative values of the outputs present in events and traces back to real numbers using a mapping function $M : Q \mapsto \mathbb{R}$. In the case of qualitative range space defined in Eq. (1) the linear mapping function defined in Eq. (4) can be used.

$$M_{linear}(q) = \begin{cases} -1.0 & \text{if } q = e- \\ 0.0 & \text{if } q = 0 \\ 1.0 & \text{if } q = L \\ 2.0 & \text{if } q = N \\ 3.0 & \text{if } q = H \\ 4.0 & \text{if } q = e+ \end{cases} \tag{4}$$

The mapping function is application and signal (output) specific at the same time. Separate mappings can be used for different outputs (due to different output ranges, for example) and it is possible to define a mapping function which weights more the possibly faulty output values (compared to the nominal values) for a single output. Such *non-linear* mapping can be defined with Eq. (5) for the qualitative range set Q in Eq. (1). The nominal values ("0", "*L*" and "*N*") are placed next to each other, while the possibly faulty output values ("*e*−", "*H*" and "*e*+") are placed farther away in both directions.

$$M_{non\text{-}linear}(q) = \begin{cases} -10.0 & \text{if } q = e- \\ -2.0 & \text{if } q = 0 \\ -1.0 & \text{if } q = L \\ 0.0 & \text{if } q = N \\ 10.0 & \text{if } q = H \\ 20.0 & \text{if } q = e+ \end{cases} \tag{5}$$

In this case nominal outputs are "0", "*L*", "*N*", while "*H*", "*e*+" and "*e*−" denote a fault, therefore they are weighted accordingly.

A linear mapping function can be defined for the refined qualitative range set of Eq. (2), as well with Eq. (6) below.

$$M_{finer}(q) = \begin{cases} -1.0 & if \quad q = e- \\ -0.5 & if \quad q = 0- \\ 0.0 & if \quad q = 0 \\ 0.5 & if \quad q = 0L \\ 1.0 & if \quad q = L \\ 1.5 & if \quad q = LN \\ 2.0 & if \quad q = N \\ 2.5 & if \quad q = NH \\ 3.0 & if \quad q = H \\ 3.5 & if \quad q = H+ \\ 4.0 & if \quad q = e+ \end{cases} \tag{6}$$

For the sake of completeness, it is worth mentioning that event *input* values can be also converted to real numbers. For instance, in the case of the two-valued qualitative set of Eq. (3) the mapping in Eq. (7) can be used. (This function can be considered as an identifying function for the set.)

$$M_{boolean}(q) = \begin{cases} 1.0 & if \quad q = 1 \\ 0.0 & if \quad q = 0 \end{cases} \tag{7}$$

The effect of using different mapping functions for the output values is described in the first case study in Section 4.

2.3. Coordinate-vectors of events and traces

Coordinate-vector of events. Events with quantitative inputs and outputs can be converted to real-valued vectors (coordinates) using an *event mapping function* $G_{IO} : E \mapsto \mathbb{R}^r$, where E is the space of events and r can be defined as:

r = (number of inputs + number of outputs)

The individual qualitative mapping functions described in Section 2.2 can be inverted, it is possible to define mappings which transform from Q back to $\mathbb{R}^r$. Therefore for the transformation of the individual input and output values to $\mathbb{R}^r$, inverse functions of these mappings can be used. For example, the event (1;"1", "0";"N") having two inputs ("1" and "0") and one output ("N") is mapped to vector [1.0, 0.0, 2.0] using the inverse of qualitative mapping function M_{linear} described in Eq. (4) for the single output and the inverse of the input mapping $M_{boolean}$ from Eq. (7) for the single input. The sequence number of the event is not converted in this case, because it is assumed to be always monotonically increasing.

If inputs are also assumed to be failure-free they can be removed from the event representation for diagnostics. In that way a different mapping function $G_O : E \mapsto \mathbb{R}^p$ can be used, where

$p =$ (number of outputs).

The output of G_O is called the *event coordinate form*. This form contains only the transformed values of the outputs.

For example, the same event (1;"1", "0";"N") in event output coordinate form is just [2.0], using the inverse of mapping function M_{linear} from Eq. (4). (Note that because inputs are considered error-free the mapping function $M_{boolean}$ is not used anymore.)

Coordinate-vectors of traces. A trace can be also converted to an m length list of r dimensional real-valued vectors, where the sequence number of an event is omitted, r is the dimension of the event as before, and m is the length of the trace. This form can be considered as a piece-wise linear trajectory in an r dimensional space, like the centroids for the first case study described in Section 4 in Fig. 6.

For example, a trace T consists of 4 consecutive events

$T =$ (1; "1", "0"; "0"), (2; "1", "0"; "L"), (3; "1", "0"; "N"), (4; "1", "1"; "N")

can be converted to the following 4 long list of 3 dimensional real vectors using function G_{IO}:

$G_{IO}(T) =$ [[1.0, 0.0, 0.0], [1.0, 0.0, 1.0], [1.0, 0.0, 2.0], [1.0, 1.0, 2.0]]

The inverse of qualitative mapping function M_{linear} from Eq. (4) is used for converting individual outputs and the inverse of the input mapping $M_{boolean}$ from Eq. (7) for individual inputs. In every element of the vector the first two real numbers correspond to the inputs (two in this case) followed by the real value of the single output (we have a single output only in this example case).

In a similar fashion to events, if inputs are considered error-free they can be removed from the trace representation, by using the same mapping function G_O for every event. This vectorial form is called the *trace coordinate form*. As before, this form only uses the output mapping function M_{linear} from Eq. (4).

For example, the trace from the previous example in trace coordinate form is the following:

$G_O(T) =$ [[0.0], [1.0], [2.0], [2.0]].

2.4. Event and trace distances

Event-to-event distance. Distance between events are calculated by using a distance function D between the corresponding coordinates of the events (already in *event coordinate form*). For example, the distance between the two-output event

$G_O(event_1) = [2.0, 2.0]$

and two-output event

$G_O(event_2) = [4.0, 2.0]$

is calculated as follows (using the Euclidean distance as D):

$$D(G_O(event_1), G_O(event_2)) = \sqrt{(2.0-4.0)^2 + (2.0-2.0)^2} = 2.0.$$

Trace-to-trace distance. Distance between traces in trace coordinate form are calculated by summing the distance values between corresponding events in the traces. Trace to trace distance is interpreted only between traces of equal length.

Let $\varphi(i)$ denote the ith event in trace φ. Based on this, the distance between trace φ_1 and trace φ_2 can be calculated as described in Eq. (8) formally (where m is the number of events):

$$E(\varphi_1, \varphi_2) = \sum_{i=1}^{m} D(\varphi_1(i), \varphi_2(i)) \tag{8}$$

For instance, the distance between the two-output trace in *trace coordinate form*

[[0.0, 0.0], [1.0, 2.0], [2.0, 3.0]]

and the two-output trace in *trace coordinate form*

[[0.0, 0.0], [1.0, 1.0], [1.0, 1.0]]

can be calculated as (using the Euclidean distance as D like in the case of events):

$$\begin{aligned} E(\varphi_1, \varphi_2) &= \sqrt{(0.0-0.0)^2 + (0.0-0.0)^2} \\ &+ \sqrt{(1.0-1.0)^2 + (1.0-2.0)^2} \\ &+ \sqrt{(1.0-2.0)^2 + (1.0-3.0)^2} \\ &= \sqrt{0+0} + \sqrt{0+1} + \sqrt{1+4} \\ &= 0 + 1 + 2.236 = 3.236 \end{aligned}$$

It is theoretically possible to use other distance functions. However in this article only the Euclidean distance is used for calculating distances between events and traces. Note that this simple distance function in its current form can only compare traces of equal length.

3. The diagnostic method

The proposed diagnostic method uses training traces which belong to the identified normal or faulty modes of the system in order to recognize the faulty mode of a not known trace (further referred as *measured trace*). The traces in the training set are annotated and labeled by the operating personnel with the faulty mode they recognized. This label may refer to a variety of faults, disturbances or malfunctions or even to a combination of those. This opens up possibilities to diagnose both internal faults, such as a broken pipe or leaking tank in the system, and external disturbances such as changes in the process feed using the proposed method.

The normal operation is considered as a special faulty mode with no fault, therefore we also need observed traces in the training set characterizing this situation. When only traces of normal operation are available with some threshold distance characterizing its accuracy, then only fault detection is possible, i.e. one can decide if a measured trace belongs to the normal operation mode, or some fault occurred the nature of which is not known.

3.1. Basic assumptions

- Time is always monotonically increasing and each time instance is present.
- Inputs are error-free while outputs might contain errors coming from the measurement or from faulty behaviour (which we are interested in finding).
- Training traces are long enough to capture the transition which will be diagnosed.

- Faults are permanent during all traces (training and measured), and their number n is fixed a priori. (There are no random faults present or faults that happen during trace execution.)
- The length of training traces and diagnosable traces are the same. This is required because the Eucledian distance function used (see Section 2.4) works on traces with the same length.

Objective of diagnosis. Given a set of training traces from different faulty and fault-free operational scenarios, and a possibly faulty measured trace (both is given in trace coordinate form) identify the operational scenario to which the measured trace is – most likely – belongs to, based on its distance from the centroids calculated from the training data.

3.2. Clustering of traces

As described in Alpaydin (2010a) in detail, clustering in general is a form of unsupervised learning where the objective is to find regularities (certain patterns occur more often than others) in a vector space. In our case the vector space is formed by the traces in trace coordinate form in accordance with the assumptions listed in Section 3.1. In this regard, a cluster can be defined as a set of training traces with similar patterns (having the same fault) while a centroid (center of a cluster) can be defined as a mean of these traces in trace coordinate form.

A popular method of clustering a vector space with distance metrics is the K-Means Clustering algorithm where the number of clusters, K is given as an input. For more details, see Alpaydin (2010b). After conversion of the traces, the K-Means clustering algorithm is executed with $K = 1$ for every diagnostic scenario (faulty and fault-free) to find a single centroid for the set of training traces having the same pattern. Because of this, the diagnostic approach described here – not like clustering in general – can be considered as a form of *supervised learning*, the centroids representing the different scenarios are trained separately.

In order to describe the clustering algorithm formally, a few *basic definitions* are needed.

1 Given a distance metric D (such as the Euclidean distance described in Section 2.4).
2 Given a set X, let us denote the number of elements in X by $|X|$.
3 Given n diagnostic scenarios let i be the scenario index going from 1 to n.
4 Given a set of traces Y in trace coordinate form, and centroids Z and W. Let the relation *Y belongs to Z* denote the set of traces from Y which are closer to Z than W using distance metric D. Similarly, *Y belongs to W* denotes the set of traces from Y which are closer to W than Z using the same distance metric D. Consequently, $|Y| \geq |Y \text{ belongs to } C|$ for every centroid C.

Acquiring and validating the cluster centres. For every faulty scenario i a set of traces in trace coordinate form are provided for creating and validating the centroids. This given set is split into a training set T_i (for creating the centroids) and a validation set V_i (for performing validation of the centroids). The split is homogeneous and the ratio $\frac{|T_i|}{|V_i|}$ is application specific. Executing the K-Means clustering with $K = 1$ on every training set T_i, the centroid C_i is formed for scenario i. These centroids are created in single trace coordinate form, and they might not be equal to any specific input trace of the training set. A centroid, like a trace is a piece-wise linear trajectory in an m dimensional space where m is the number of outputs, and the length of the piece-wise linear trajectory is the length of the trace (number of events in the trace). For example, for a training set which contains 100 event long traces, and 20 output values for every event, the representation of the trace will be a 100 long line in 20 dimensional space.

After every centroid C_i is formed from the training sets, the validation sets are used to calculate the *fault detection rate* (FDR_i) for every faulty scenario i using the formula defined in Eq. (9). The sequence $\{FDR_i | i = 1, \ldots, n\}$ also gives an overall fitness of the model, where

$$FDR_i = \frac{|V_i \quad belongs \quad to \quad C_i|}{|V_i|} \tag{9}$$

Note that this FDR value is conceptually the same as the value which was the base for the comparison for the different diagnostic approaches in review article (Yin et al., 2012).

3.3. Steps of the diagnostic procedure

The steps are executed in two phases: (i) an off-line **training phase** which creates the trace clusters identified with a fault label and its centroid, and (ii) an on-line **diagnosis phase** which can be executed with the known clusters for an measured trace we want to diagnose.

1 **Training phase.** Every input trace is converted to trace coordinate form using the method in Section 2.3 for every training scenario. Because inputs are considered as fixed and error-free (both in their number and value) and sequence numbers are increasing strictly monotonically (due to the basic assumptions laid down in Section 3.1), only outputs are participating further in clustering (the inputs and the sequence numbers are not present in this form).
2 As defined in Section 3.2, sets T_i, V_i and C_i are created for each training scenario $i = 1, \ldots, n$.
3 The diagnostic model is validated using C_i and sets V_i after all centroids are determined. FDR_i values are calculated for every training scenario as described in Section 3.2 Eq. (9) for each $i = 1, \ldots, n$.
4 Each cluster centre C_i is labeled with the inputs of training scenario i and the particular fault (those are fixed).
5 **Diagnosis phase.** Given a measured trace which is converted into *trace coordinate form*, the nearest centroid can be determined by computing its distances from centroids C_i for training scenarios $i = 1, \ldots, n$, using a distance such as the one described in Section 2.4, and finding the closest C_i. The fault index i which corresponds to the nearest centroid is regarded as the most probable fault mode of the system during the execution of the measured trace.

3.4. Dealing with faults not considered a priori in the training set

When *unknown faults* (faults missing from the training set for the diagnoser) are present during the *Diagnosis phase* of the diagnosis (see Section 3.3), we can define a modified distance function having a *distance threshold* T in order to separate these cases from the known faults. With this value, a modified distance function in Eq. (10) (based on the simple Eucledian distance E described in Eq. (8)) can be used

$$E_T(\varphi_1, \varphi_2, T) = \begin{cases} E(\varphi_1, \varphi_2) & E(\varphi_1, \varphi_2) \leq T \\ \infty & \text{otherwise} \end{cases} \tag{10}$$

in a way that if the distance is ∞ then the result of diagnosis is "Unknown". The distance threshold T can be chosen based on the diameter (the maximum distance between elements in the training set belonging to the same centroid). In general, it can be said that a diagnostics method shall prepare for the presence of unknown faults. By using a distance function as the one described in Eq. (10)

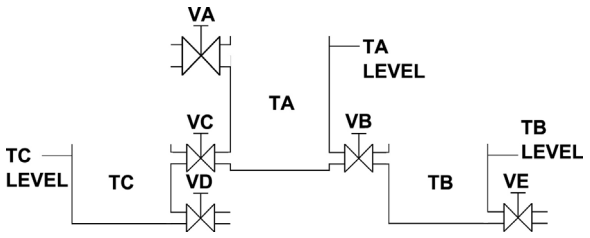

Fig. 1. Simple process system used for the case studies.

this can be achieved. Application of a distance function like this is out of the scope of this article.

4. Simple process example

The aim of this simple example is to compare the resolution of the diagnostic method on a simple process system controlled by an operational procedure under the presence of multiple faults. Qualitative sets with different resolution and different mapping functions were evaluated and the cases were compared based on their *FDR* values (as described in Section 3.3.)

4.1. Process system and trace description

A simple example of a controlled composite process system with three tanks (see Fig. 1) that is driven by an operational procedure is used. The operational procedure in this case was filling up the system with fluid.

First VA was opened. Later, when TA reached nominal level then VC and VB were opened and TB and TC were filled up with fluid until they reached nominal level. Finally, output valves VD and VE were opened. The corresponding operational procedure in a tabular form can be seen in Table 1.

The following faults were taken into account for each tank:

- The leak of the tank. In the case study two different leak types were used.
  - For the cases described in Sections 4.2–4.4 the size of the leak prevents any fluid from staying inside the tank, therefore fluid level constantly stays at qualitative value 0. This is called a *rupture*.
  - On the other hand, in Section 4.5 the diagnosability of a more realistic, smaller leak is surveyed – this results in 10% loss of fluid per time instant from a tank. This is called a *leak*.
- The positive bias failure of the level sensor. The level sensor always detects a qualitative value one degree higher than the actual level of the tank. Given the qualitative set defined in Eq. (1), the level sensor outputs "*H*" instead of "*N*". This fault is the same for all presented cases.
- The negative bias failure of the level sensor. The level sensor always detects a qualitative value one degree lower than the actual level of the tank. Given the qualitative set defined in Eq. (1), the level sensor outputs "*L*" instead of "*N*". This fault is the same for all presented cases.

Based on these faults, reference operation traces were created which contained all single and dual occurrences of the faults for all three components. Training trace sets were formed from each reference trace copying them 5000 times and applying a simulated measurement error function (as described in Section 2.1) on each set with 6% error probability with neighbour level L=2 in the refined case of Section 4.4 and neighbour level L=1 in the other two case studies. (We have performed a few measurements and this specified number of traces, error probability and neighbour level turned out to be a good choice for visualizing the results in the case studies.) Later, the diagnostic procedure described in Section 3.3 was used to find the centroids and validate them. The ratio of the training and the validation set size was chosen to be 1:1 for the sake of simplicity. We tested other ratios (such as 4:1) for distributing the traces but no significant differences were observed in the results for this simple case study. The *FDR* values (refer to Eq. (9)) were calculated for each set (shown in ascending order on the graphs in Fig. 2–5), these were used for comparison in the case studies.

4.2. Single and dual faults in the system

The first part of the case study used a linear mapping function (see Eq. (4)) to map qualitative outputs to numeric values. The *FDR* values for each scenario can be seen in Fig. 2 in this case.

4.3. The effect of nonlinear output mapping

The second part of the case study used a non-linear mapping function (see Eq. (5)) instead of the linear one. This mapping function, due to its non-linear characteristic, made centroids (denoting different faulty scenarios) farther away from each other.

The *FDR* values also had slightly decreased, their values are depicted in Fig. 3. The random noise – added by the simulated measurement error, and the increased distance between centroids – in the training set caused the cluster centres to be less accurate, even though they are farther from each other, this made the overall diagnosis slightly less accurate.

4.4. The effect of using refined qualitative set

In this case the refined qualitative set from Eq. (2) was used during simulating the measurement errors in the training trace sets (all reference traces remained the same). A slightly refined version of the linear mapping function described in Eq. (6) was used.

The *FDR* values of the scenarios for this case study can be seen in Fig. 4. The figure shows the accuracy of the diagnosis was slightly better in this case compared to both earlier case studies.

4.5. The effect of realistic leaks

In this case instead of a rupture (full loss of containment), a 10% loss per time instant (a leak) was present, with the same qualitative set as in Section 4.4. The results of this change can be seen in Fig. 5. Comparing the results with the ones in Fig. 4 a significant change in the *FDR* values are observed in this case, and the *FDR* values were above 0.8 for every scenario.

4.6. General observations on simultaneous fault detection

In this section a few observations on the simple process example are described.

A few of the determined centroids for the faulty scenarios described in Section 4.4 can be seen in Fig. 6. On the figure a coordinate of the centroid is represented as a piece-wise linear trajectory in the three dimensional space, with the three axes as the three output dimensions (levels in tanks TA, TB and TC) and the separate points represent different sequence numbers in the trace.

In general, the proposed diagnostic method was able to perform simultaneous fault detection and isolation in some cases for this simple case study. The following observations could be made regarding the detection of dual faults (with the assumptions described in Section 3.1) in this simple process example:

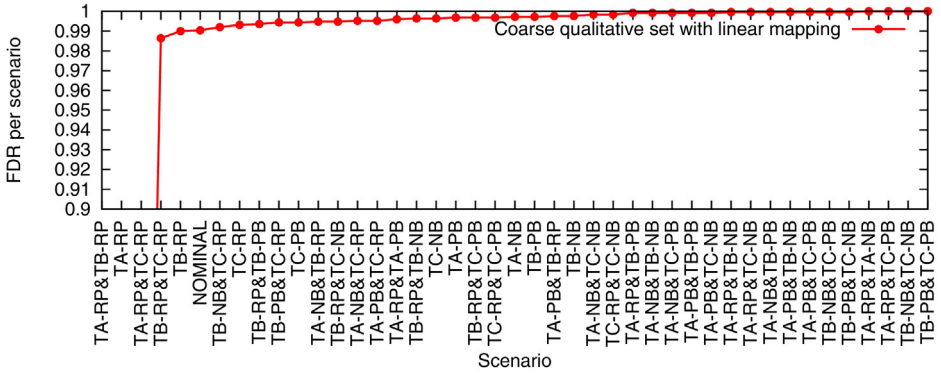

Fig. 2. Case study with linear mapping function and tank rupture. Scenarios are sorted in ascending order by their *FDR* value. Values for *TA – rupture* (*TA – RP*), *TA and TB ruptures* (*TA – RP* & *TB – RP*) and *TA and TC ruptures* (*TA – RP* & *TC – RP*) are smaller than 0.9 hence not shown (they are placed around 0.3, refer to Section 4.6 for details).

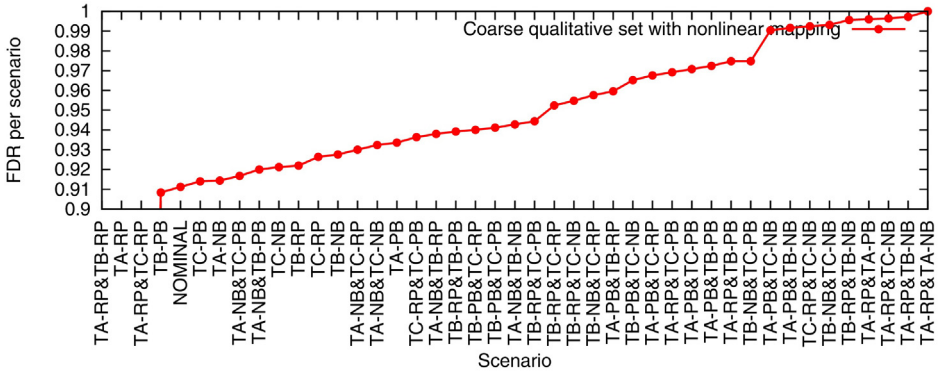

Fig. 3. Case study with nonlinear mapping function and tank rupture. Scenarios are sorted in ascending order by their *FDR* value. Values for *TA – rupture* (*TA – RP*), *TA and TB ruptures* (*TA – RP* & *TB – RP*) and *TA and TC ruptures* (*TA – RP* & *TC – RP*) are smaller than 0.9 hence not shown (they are placed around 0.3, refer to Section 4.6 for details).

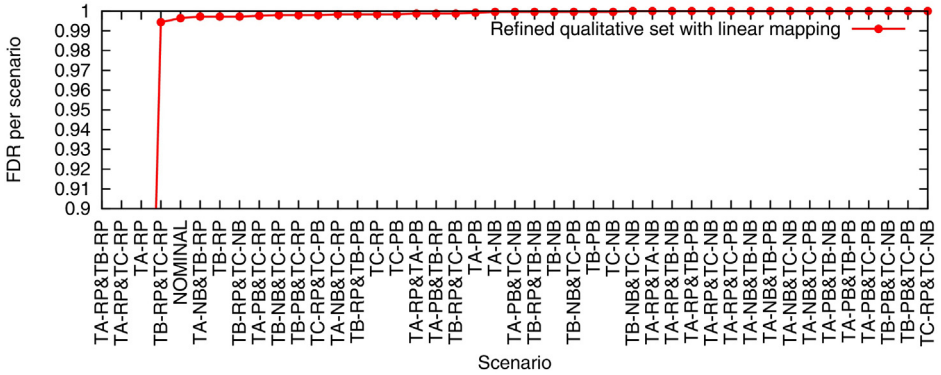

Fig. 4. Case study with refined qualitative values and tank rupture. Scenarios are sorted in ascending order by their *FDR* value. Values for *TA – rupture* (*TA – RP*), *TA and TB ruptures* (*TA – RP* & *TB – RP*) and *TA and TC ruptures* (*TA – RP* & *TC – RP*) are smaller than 0.9 hence not shown (they are placed around 0.3, refer to Section 4.6 for details).

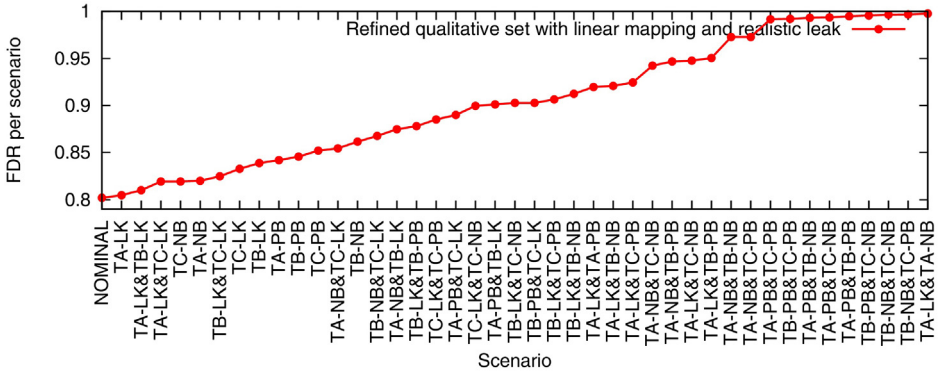

Fig. 5. Case study with refined qualitative values and tank leak. Scenarios are sorted in ascending order by their *FDR* value.

Table 1
Nominal trace for the case study. System input "0" means "closed", "1" means "opened" valve states, while system outputs (based on qualitative set Eq. (1)) "0" means "no level", "*L*" means "low", "*N*" means "normal" levels in the tank.

Sequence number	System inputs					System outputs		
	VA	VB	VC	VD	VE	TA	TB	TC
1	1	0	0	0	0	0	0	0
2	1	0	0	0	0	L	0	0
3	1	1	1	0	0	N	0	0
4	1	1	1	0	0	N	L	L
5	1	1	1	1	1	N	N	N

1 If the faulty scenarios affect *different* output variable(s) on the process system, and their effects are *independent* of each other, they can be distinguished even though they appear simultaneously. Taking a simple example a bias failure in TA and a rupture on TC at the same time affect different outputs (level sensor of TA and TC), they are not related to each other, hence they can be detected and distinguished from the rest of the faults. See the scenario *TA pos bias and TC rupture* in Fig. 6.
2 If the faulty scenarios affect the *same* output variable on the process system, but their effect is *independent* of each other, they can be still distinguished. For example, a tank rupture causes the level to be constant "0" in the tank, but a positive bias failure changes the sensor value to constant "*L*" (low) level. See the centroids from faults *TC rupture* and *TC rupture and TC pos bias* in Fig. 6.
3 Dual faults cannot be separated from each other if they are not *independent* (i.e. there exists a causal relationship between them). In a simple case, taking the process system in Fig. 1 if a rupture in TA occurs, the level of the fluid will be "0" in TB and TC. This causes the detection of rupture(s) in TB and in TC **practically indistinguishable** from the rupture in TA, because faults *TA rupture*, *TA and TB ruptures* and *TA and TC ruptures* produce exactly

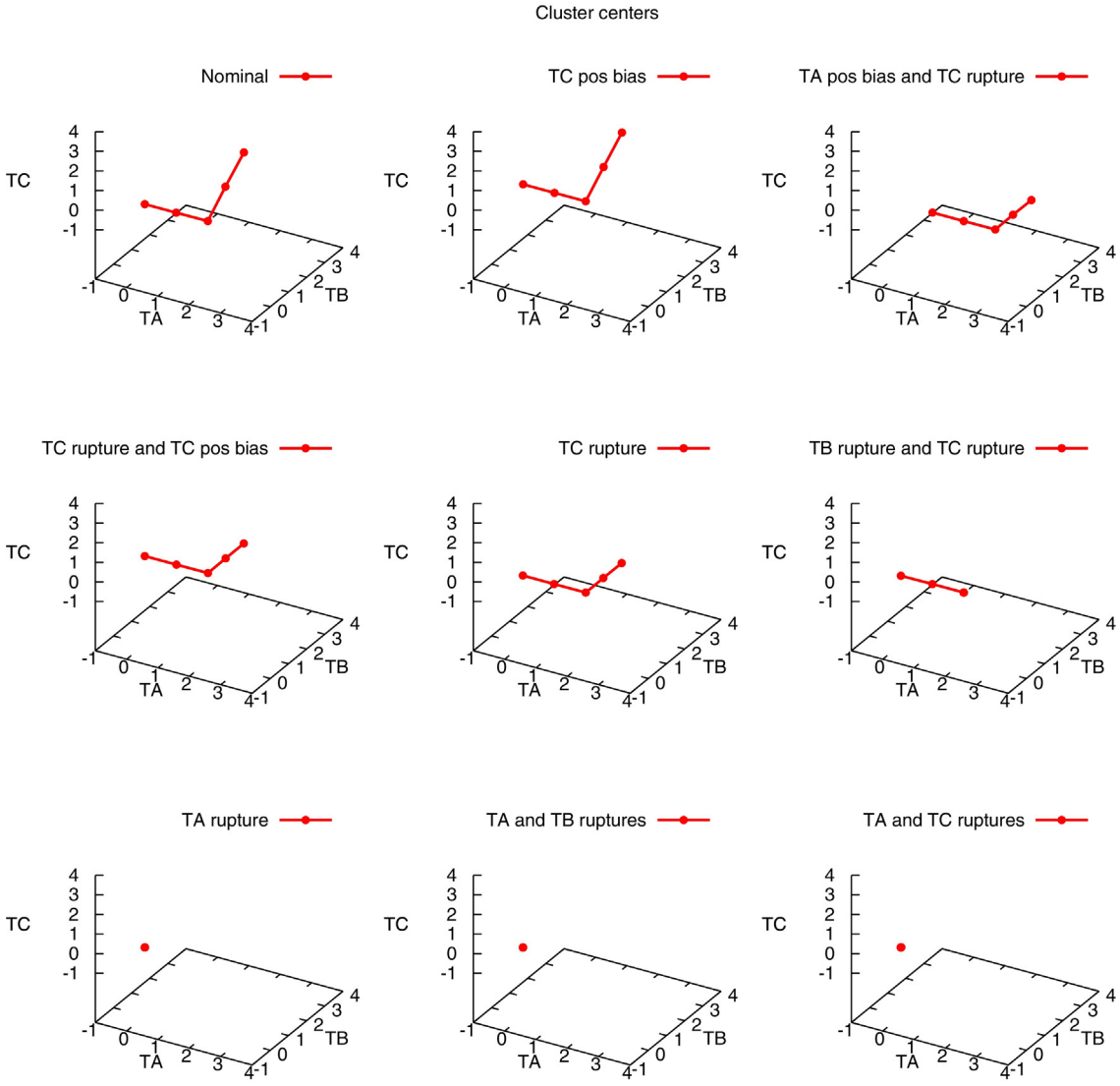

Fig. 6. Centroids for a few scenarios for the case study in Section 4 in three dimensional form. Axes represent level output values for the three tanks, with qualitative mapping $e - = -1$, $0 = 0$, $L = 1$, $N = 2$, $H = 3$. Single dot represents a centroid where all values are the same over time.

the same traces (constant zero level in all three tanks) due to the rupture in source tank TA. This observation is relevant for the cases described in Section 4.2–4.4 (only these cases have tank ruptures).

Due to the fact that they cannot be distinguished, the *FDR* values for these scenarios are as low (around 0.33) as they do not even fit to the display interval of [0.9, 1.0] in Fig. 2–4. The centroids are displayed in the bottom row in Fig. 6.

4 If the simulated measurement errors were eliminated, then the diagnostics had 100% accuracy in every case, using every mapping function – except in the case when the faults were not independent as described above. If simulated measurement errors were present, then the following observations can also be made on the nature of the output mapping functions, in contrast to the linear mapping function with its *FDR* distribution in Fig. 2:
   (a) The diagnostic accuracy *improved* when a linear mapping function with refined qualitative set was used (see the *FDR* distribution in Fig. 4).
   (b) The diagnostic accuracy *worsened* when a non-linear mapping function was used (see the *FDR* distribution in Fig. 3). In this case the combination of the simulated measurement error and the non-linear mapping had been responsible for the loose position of the centroids and the less accurate diagnostics.

5 When a leak (10% loss) was present instead of a rupture (full loss of containment) in Section 4.5, then the diagnostic accuracy greatly improved for the previously indistinguishable cases having a rupture (*TA rupture*, *TA and TB rupture* and *TA and TC rupture*). In this case faults *TA leak*, *TA and TB leaks* and *TA and TC leaks* became **distinguishable** from each other (because they did not produce the same traces anymore, like the corresponding rupture faults did). This effect is responsible for the increased *FDR* values for all scenarios (all above 0.8). On the other hand, due to the fact that the 10% leak caused only a minor difference between leaky and leak-free scenarios in terms of the qualitative values, the overall diagnostic accuracy worsened. (The majority of the *FDR* values were between 0.85 and 0.95 in Fig. 5, while in the original refined case most of them were above 0.99 in Fig. 4.)

5. Case study

As a more serious example, a commonly used process system (the Tennessee Eastman Challenge problem) is used to demonstrate the diagnostics capabilities of the algorithm. As in the previous case, for the various disturbances (faults) of the problem the fault detection ratio (*FDR*) is calculated (as described in Section 3.2). A similar survey for many different statistical methods had been performed in Yin et al. (2012) on the same process system and disturbances for statistical methods.

5.1. Tennessee-Eastman process

The Tennessee Eastman process (later mentioned as TEP) is widely used and accepted for developing, studying and comparing process control and diagnostics algorithms. It consists of a reactor/separator/recycle arrangement involving two simultaneous gas-liquid exothermic reactions and two additional byproduct reactions. The process has 12 available valves for manipulation and 41 available output measurements for monitoring or control. In the case study the first 15 of the original 20 simulated disturbances are considered (see Table 2 for details), due to the fact that the last 5 disturbances are of type "Unknown", and we wanted to emphasize diagnosing the known faults in the case study. Note that these "not known" disturbances were part of the actual diagnosis for the rest of the disturbances, but their *FDR* values have not been calculated, and observations have not been made for them.

Table 2
A subset of the disturbances, along with their types of the Tennessee Eastman Challenge problem, as described in the original article (Downs and Vogel, 1993).

Identifier	Disturbance	Type
IDV(1)	A/C feed ratio, B composition constant (stream 4)	Step
IDV(2)	B composition. A/C ratio constant (stream 4)	Step
IDV(3)	D feed temperature (stream i)	Step
IDV(4)	Reactor cooling water inlet temperature	Step
IDV(5)	Condenser cooling water inlet temperature	Step
IDV(6)	A feed loss (stream I)	Step
IDV(7)	C header pressure loss – reduced availability (stream 4)	Step
IDV(8)	A, B, C feed composition (stream 4)	Random variation
IDV(9)	D feed temperature (stream 2)	Random variation
IDV(10)	C feed temperature (stream 4)	Random variation
IDV(11)	Reactor cooling water inlet temperature	Random variation
IDV(12)	Condenser cooling water inlet temperature	Random variation
IDV(13)	Reaction kinetics	Slow drift
IDV(14)	Reactor cooling water valve	Sticking
IDV(15)	Condenser cooling water valve	Sticking

The TEP produces two products, an inert and a byproduct from four reactants (there are eight components altogether, A, B, C, D, E, F, G and H). The following reactions take place based on the components in this example process system (based on Downs and Vogel, 1993):

$$A(g) + C(g) + D(g) \rightarrow G(liq) \quad (Product \quad 1)$$

$$A(g) + C(g) + E(g) \rightarrow H(liq) \quad (Product \quad 2)$$

$$A(g) + E(g) \rightarrow F(liq) \quad (By \quad product)$$

$$3D(g) \rightarrow 2F(liq) \quad (By \quad product)$$

These components are also shown on the flow-sheet of the process system in Fig. 7. For more details, refer to Downs and Vogel (1993). In order to be more consistent with the original article, the term *disturbance* will be used for faults in the description of this case study.

The original model was written in FORTRAN, but in this case study the revised MATLAB version of the TEP (described in Bathelt et al., 2015) was used for generating the training traces for the algorithm.

The MATLAB model has already contained simulated measurement errors, so the measurement error generation approach described in Section 2.1 was not used in this case, the raw values were just taken from the simulated model without change. Due to the available functionality of the model, only single disturbances were considered in this case.

5.2. Preparation of the data

In this case study the diagnostic algorithm's ability to identify the various disturbances (considered as fault modes from the algorithm's perspective) are surveyed for the TEP. Two operational modes, an "open-loop" mode and a controlled steady-state mode (refer to "Mode 1" in Downs and Vogel, 1993) were considered. Inputs were modified by the simulated controller in steady state mode but were not taken into account. Also, only a subset of the original 41 outputs (22 "Continuous process measurements", see Table 4 in Downs and Vogel, 1993) were taken into account in the case study. The reason for this is that we wanted to focus on the continuous measurements only during diagnosis (the rest

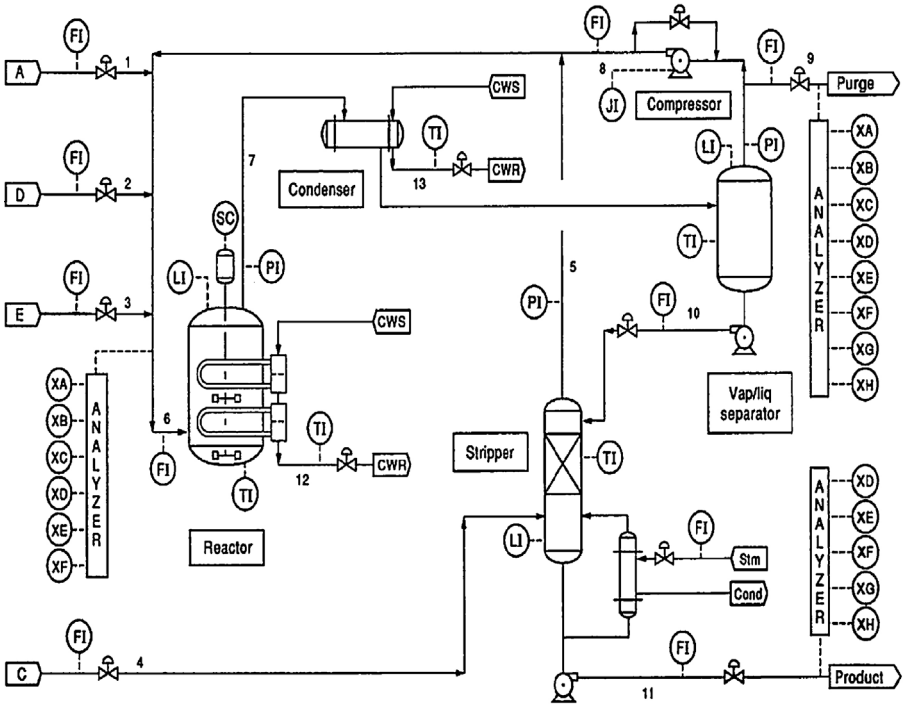

Fig. 7. Tennessee Eastman Challenge problem from Downs and Vogel (1993).

of the outputs were relatively infrequently sampled process measurements measuring concentrations in the output of the process system, which would not change the findings of the case study). Therefore the output of the simulation was a set of events (each containing 22 output values) ordered by time. A single event was describing a state of the system at a different sequence number for a single execution. This is converted to the trace format required by the algorithm where inputs were not considered (there was no input changes during the execution of traces), while the list of outputs contained every output from the simulation.

After the results had been collected from the MATLAB simulator, the traces were trimmed to equal length (this was one of the assumptions from Section 3.1). Moreover, the raw data was sampled at different intervals to determine the effect of sampling on the diagnostic accuracy. The reactor in the "open loop" case always shut down after approximately 1 h of operation (due to the high pressure threshold built into the MATLAB model), while in the steady-state case ("Mode 1") always a 5-h model MATLAB simulation was performed (the model could have been executed longer in this case).

5.3. Results

The corresponding output values were normalized and converted to trace coordinate form using a qualitative mapping function. Normalization was performed so that the same qualitative function could be used for all outputs, this made the execution of the diagnosis simpler. Finally, the diagnostic algorithm was executed for the traces, centroids are formed from the training traces and the fault (disturbance) detection rates (*FDRs*) are calculated from the validation set for every disturbance case.

Table 3 shows a summary about the most important properties of the executed cases. These are the following:

1 **Number of training and validation traces.** The number of times the simulation was executed for every disturbance scenario described in Table 2 to get the traces required by the algorithm. The first half of the traces was used for training while the second half is for validation of the trained centroids (the *FDR* values were calculated from diagnosis on the validation set, see Section 3.2 for details). We have experimented with other training/validation ratios, such as 4:1 but we have not seen significant differences in the results of the case study (like in the case of the simple case study in Section 4). The exact number of training and validation traces (simulator runs) was chosen in a way that we have enough traces for each disturbance to compare the diagnostic accuracy between them. A couple of hundred traces per disturbance proved to be more than enough for this purpose.

2 **Trace trim.** Traces are trimmed at this length after conversion. Trimming is needed so that every trace participating in the diagnosis will have the same length (to comply to the assumptions in Section 3.1). This was required because for some of the disturbances the MATLAB simulation (see Bathelt et al., 2015) produced fewer number of events due to the fact that internal error thresholds (eg. "Low stripper liquid level" in the case of IDV(6) in Mode 1) were met, which correctly caused the simulation to halt immediately. This resulted in shorter traces for these disturbances. In order to comply with the diagnostic assumptions in Section 3.1 (all traces shall have the same length), longer traces for other disturbances were trimmed accordingly, so that every trace would have the same length.

3 **Sampling rate.** This item describes how the simulation output was sampled. For example, "7" means that every seventh event was taken from the simulation output (the rest was thrown away), "1" means that every event was kept. In the "Mode 1" case due to the chosen sampling rate, the traces needed to be trimmed at an earlier event, so that traces for every disturbance have the same length. This effect can also be seen in Table 3.

4 **Qualitative sets.** The resolution of qualitative sets used to represent the previously normalized output values. After normalization, every output had the same range, so the same qualitative output mapping function could be used for them. In every case (except for ∞) a linear qualitative mapping function (like the one

Table 3
Basic properties of the executed case studies.

Case study	Qualitative sets	Number of training and validation traces	Trace trim	Sampling rate
Mode 1 #1	∞	366	100	1
Mode 1 #2	Refined	366	100	1
Mode 1 #3	Coarse	366	100	1
Mode 1 #4	∞	366	59	7
Mode 1 #5	Refined	366	59	7
Mode 1 #6	Coarse	366	59	7
Open Loop #1	∞	300	100	1
Open Loop #2	Refined	300	100	1
Open Loop #3	Coarse	300	100	1

described in Eq. (4)) was used, which divided the interval [0, 1] into n qualitative sets evenly.

(a) ∞ means that qualitative sets are not used for the output.

(b) *Refined* means a 8-element refined qualitative set for the output:

$$Q = \{0, 0L, L, LN, N, NH, H, H+\} \tag{11}$$

(c) *Coarse* means a 3-element qualitative set for the output:

$$Q = \{0, L, N\} \tag{12}$$

The parameters in Table 3 were chosen based on preparatory simulation experiments, so they are valid for this case study only, and might be different for other process systems or operational procedures.

After the traces were converted, they were given as input to the diagnostic algorithm. The first half of them were used for training (calculating the centroids using K-Means clustering with $K = 1$), while the remainder was used for validating these clusters and determine the *FDR* values for every disturbance in every case study. These values are collected in Table 4. The original article also contained disturbances from type "Unknown". Due to the not known nature of these disturbances they are not displayed in Table 4 for the case studies – they were only taken into account during calculating of the *FDR* values for the other, known disturbances.

5.4. Observations based on the results

Based on these results the following observations can be made:

1 It can be seen in Table 4, that many of the *Step* type disturbances, such as IDV(1), IDV(6) or IDV(7), of the TEP were detected with 100% *FDR*. This means that the diagnostic algorithm could isolate a separate centroid – farther from the rest – for these disturbance types successfully. A reason for this in the case of IDV(7) can be seen in Fig. 8 which shows a TEP output ("A and C feed (stream 4)") for all the considered disturbance scenarios and cases, with IDV(7) as bold black line while the rest of the disturbances with ordinary red lines. This figure shows a significant contrast between IDV(7) and all the rest of the disturbances in the value of the TEP output for all the executed case studies described in Table 3. The dissimilarity in the value is so outstanding in this case that not even:
   - a significant change in the Sampling Rate (from "1" to "7", where $\frac{6}{7} \approx 85\%$ of the traces were dropped, see diagrams in the first row in Fig. 8),
   - the use of a very coarse 3-element qualitative set

   could not affect the diagnostic capability of the algorithm for this disturbance significantly. Differences like this placed the centroid for IDV(7) farther than the centroids for the other disturbance scenarios, and this is responsible for the 100% accuracy in this case. Similar differences to this in output values are responsible for the 100% accuracy in some of the other cases when *Step* type disturbances were diagnosed. *Step* type disturbances were assumed among the diagnostic assumptions in Section 3.1. (Faults (disturbances) are permanent and their number is fixed a priori.)
2 *Step* type disturbances could be well diagnosed in case of both the "Open Loop" operational mode and the steady-state ("Mode 1") operational mode of the simulation.
3 In case of the "Mode 1" case studies the use of different Sampling rates ("1" and "7") had no significant effect on the diagnostic result for *Step* type disturbances IDV(1), IDV(2), IDV(6) and IDV(7). These disturbances could still be diagnosed for both Sampling rates, despite the fact that the diagnosis was based on a fraction of the available events only (every seventh event). Due to the loss of data during sampling, the *FDR* for some other *Step* type disturbances (IDV(4) and IDV(5)) reduced drastically.
4 The use of the three different qualitative sets (as described in Section 5.3) had also no effect on the diagnostic result.
5 Disturbances from other types (such as *Random Variations*) could be detected with a very low *FDR*. In this case individual centroids are created by the clustering overlapped each other, therefore the disturbance could not have been identified by the algorithm properly using distance calculation.

5.5. Discussion

The experiences obtained with the TEP case study highlighted some of the advantages and also the limitations of the proposed diagnostic method that can be summarized as follows.

1 Our diagnostic method is *data-driven*, that is, black box type in nature. It compares the observed traces with "patterns" of transient behaviors of the system in different faulty modes. Therefore, the diagnostic results cannot be easily explained by the physics and chemistry of the process: for this a first principles dynamic model would be necessary. On the other hand, the diagnostic accuracy and resolution can be adaptively improved, when new characteristic patterns are found, associated to faulty modes, and added to the training set.
2 An important feature of the proposed method is that it can handle traces that correspond to strongly transient operation of the plant to be diagnosed. Any fault or disturbance that has a noticeable effect on the transient that was chosen as a normal operation can possibly be diagnosed. This means, that *diagnosability is strongly related to the sensitivity of the normal operation transient to the fault*. In addition, the type of the transient can be selected taking into account which faults are to be diagnosed using operating experience.

Table 4
FDR values in % for the different case studies. The "Dist. ID" is the identifier from Table 2.

Dist. ID	Mode 1 #1	Mode 1 #2	Mode 1 #3	Mode 1 #4	Mode 1 #5	Mode 1 #6	Open Loop #1	Open Loop #2	Open Loop #3
IDV(1)	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	83.333%	84.0%	**100.0%**
IDV(2)	98.37%	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	9.333%	8.0%	19.333%
IDV(3)	27.174%	34.783%	34.239%	21.196%	15.761%	26.630%	7.333%	12.0%	40.0%
IDV(4)	**100.0%**	**100.0%**	**100.0%**	38.043%	73.370%	86.957%	**100.0%**	99.333%	**100.0%**
IDV(5)	**100.0%**	**100.0%**	**100.0%**	58.696%	**100.0%**	95.652%	**100.0%**	**100.0%**	**100.0%**
IDV(6)	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**
IDV(7)	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**
IDV(8)	21.196%	32.065%	14.674%	47.283%	71.739%	55.435%	1.333%	5.333%	5.333%
IDV(9)	14.674%	21.196%	16.848%	16.848%	23.370%	23.370%	6.0%	3.333%	8.0%
IDV(10)	8.696%	28.804%	24.457%	34.783%	30.435%	53.804%	8.0%	4.667%	10.0%
IDV(11)	5.978%	46.196%	29.891%	72.283%	48.370%	88.587%	6.667%	8.667%	21.333%
IDV(12)	9.783%	42.935%	35.87%	22.283%	38.587%	65.217%	14.667%	9.333%	37.333%
IDV(13)	2.717%	5.435%	45.109%	18.478%	20.652%	19.022%	4.667%	8.0%	10.0%
IDV(14)	92.391%	94.565%	97.826%	88.043%	67.935%	97.826%	5.333%	6.667%	6.0%
IDV(15)	13.587%	23.913%	18.478%	17.935%	16.848%	17.391%	6.0%	4.667%	8.0%

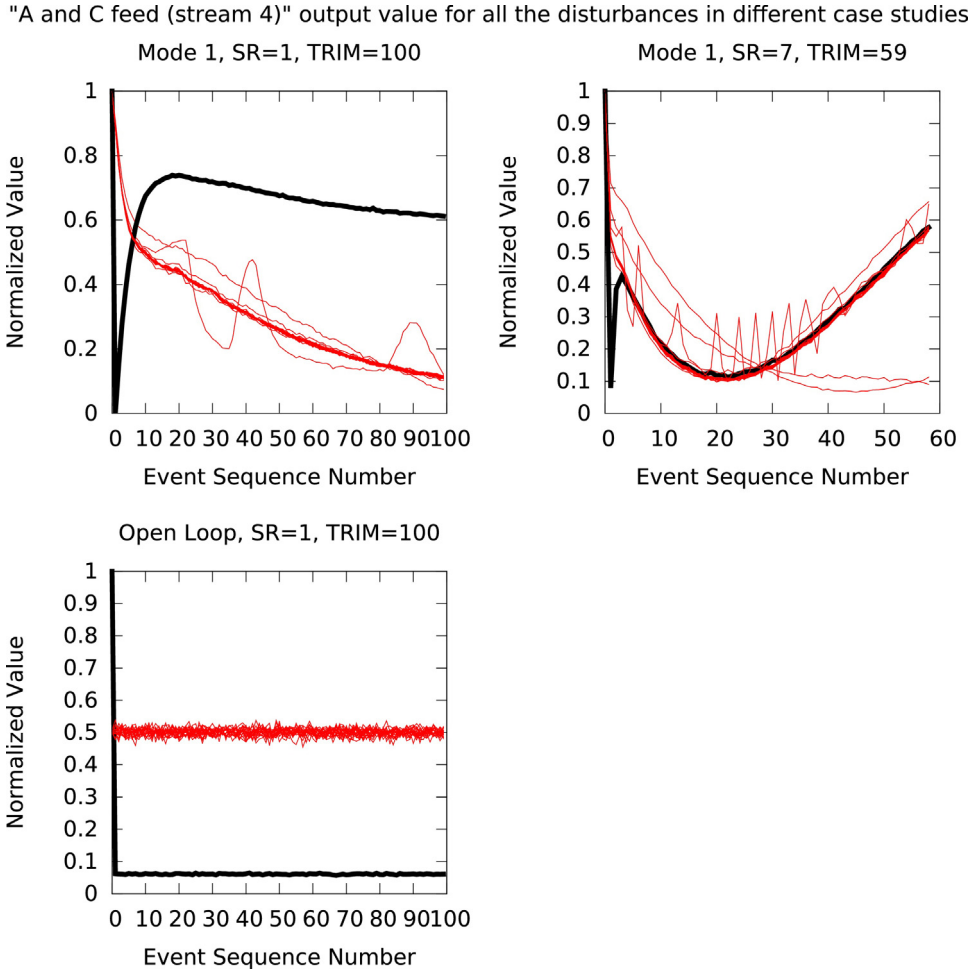

Fig. 8. Distribution of a single output over all scenarios and case studies. TEP disturbance IDV(7) with bold black line, while the rest of the disturbances are shown with ordinary red lines. SR = Sampling Rate, TRIM = Trace trim (according to Table 3). Normalization was performed by transforming the values of "A and C feed" to the interval of [0,1] for every disturbance per scenario right after the simulation. This operation was required to be able to use the same qualitative mapping function for all the outputs during diagnosis.

3 It is often the case that complex dynamic systems (including process plants) are more sensitive for any disturbance or fault if they perform an open-loop transient as compared to the controlled steady-state operation. *Step change disturbances add a good excitation* to even a controlled system to provoke characteristic-enough transient behavior, this explains the better diagnosability of step change disturbances or faults compared to random variation, drift or sticking.

6. Conclusion

A data-driven diagnostic approach was described in this paper based on clustering qualitative event sequences. The method was based on a sufficiently high number of training traces recorded from different nominal and faulty scenarios. After training, input traces were categorized (diagnosed) by the most likely scenario based on the training traces. The method had two

main phases, the off-line training and the on-line diagnostics phase.

After preprocessing, the event sequences were converted to an m-dimensional vector space with a distance metric defined. K-means clustering was used for every faulty and nominal scenario to find a single centroid. After every centroid was found the on-line diagnostic is executed.

In the on-line diagnosis phase, arbitrary measured traces were converted to coordinate vector form. Using this form, the closest centroid was determined which is the result of the diagnosis for the trace.

The aim of the simple process example was to examine the diagnostic accuracy of the proposed method on the same composite process system driven by an operational procedure, under the presence of multiple faults and different output mapping functions. Three types of mapping functions (coarse and finer linear, nonlinear) were used and their positive or negative effects on the accuracy were compared. We also provided a discussion on how the diagnostic algorithm can be used for simultaneous fault detection.

A complex diagnostic case study using the benchmark of Tennessee Eastman process (TEP) was also presented to illustrate the efficiency of the proposed method and to compare its performance with some of the statistical methods. It was found that not only constant step-type faults (disturbances) could be detected with a high fault detection rate but also during a transient operation of the process.

As a future improvement, provided the diagnosable process system can be decomposed into smaller sub-process systems, the algorithm can be modified to perform diagnosis on separate lower-level components and combining the failures found in them. This would have a better performance compared to the composite approach, albeit the complexity would be higher (due to the fact that different faults can be in causal relationship with each other).

Currently the algorithm works only on historical data. As an improvement, the diagnostic approach can also be extended with more real-time operational capabilities, working on partial traces and comparing them to (relevant parts) of the already calculated centroids as individual events arrive for the traces. In that way an operational procedure under execution can be also diagnosed – before its execution is complete.

Acknowledgements

We acknowledge the financial support of this work by the National Research Development and Innovation Office through grant No. K115694. We would also like to thank the reviewers of this article for their insightful comments, which helped us to improve its quality greatly.

References

Agudelo, C., Anglada, F.M., Cucarella, E.Q., Moreno, E.G., 2013. Integration of techniques for early fault detection and diagnosis for improving process safety: application to a fluid catalytic cracking refinery process. J. Loss Prev. Process Ind. 26, 660–665.

Alpaydin, E., 1998. Soft vector quantization and the EM algorithm. Neural Netw. 11, 467–477.

Alpaydin, E., 2010a. Introduction to Machine Learning, 2nd ed. The MIT Press, Cambridge, Massachusets, London, England, pp. 11.

Alpaydin, E., 2010b. Introduction to Machine Learning, 2nd ed. The MIT Press, Cambridge, Massachusets, London, England, pp. 145–149.

Alves de Medeiros, A.K., Guzzo, A., Greco, G., van der Aalst, W.M.P., Weijters, A.J.M.M., van Dongen, B., Saccà, D., 2008. Process Mining Based on Clustering: A Quest for Precision. In: Business Process Management Workshops, vol. 4928. Springer Berlin Heidelberg, pp. 17–29, ISBN 978-3-540-78237-7.

Bathelt, A., Ricker, N.L., Jelali, M., 2015. Revision of the Tennessee Eastman Process Model 9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015 Whistler , Canada, 2015 Volume 48, Issue 8, pp. 309–314.

Downs, J.J., Vogel, E.F., 1993. A plant-wide industrial process control problem. Comput. Chem. Eng. 17, 245–255.

Gabbar, H.A., 2007. Improved qualitative fault propagation analysis. J. Loss Prev. Process Ind. 20, 260–270.

Guo, L., Kang, J., 2015. An extended HAZOP analysis approach with dynamic fault tree. J. Loss Prev. Process Ind. 38, 224–232.

Hangos, K., Bokor, J., Szederkényi, G., 2004. Analysis and Control of Nonlinear Process Systems. Springer.

Harrou, F., Nounou, M.N., Nounou, H.N., Madakyaru, M., 2015. PLS-based EWMA fault detection strategy for process monitoring. J. Loss Prev. Process Ind. 36, 108–119.

Kim, H.G., Kim, C., 2014. Interval clustering algorithm for fast event detection in stream monitoring applications. Pattern Recognit. Lett. 36, 171–176.

MacGregor, J., Cinar, A., 2012. Monitoring, fault diagnosis, fault-tolerant control and optimization: data driven methods. Comput. Chem. Eng. 47, 111–120.

Maurya, M.R., Rengaswamy, R., Venkatasubramanian, V., 2005. Fault diagnosis by qualitative trend analysis of the principal components. Chem. Eng. Res. Des. 83, 1122–1132.

Maurya, M.R., Rengaswamy, R., Venkatasubramanian, V., 2007. A signed directed graph and qualitative trend analysis-based framework for incipient fault diagnosis. Chem. Eng. Res. Des. 85, 1407–1422.

Mercurio, D., Podofillini, L., Zio, E., Dang, V.N., 2009. Identification and classification of dynamic event tree scenarios via possibilistic clustering: application to a steam generator tube rupture event. Accid. Anal. Prev. 41, 1180–1191.

Németh, E., Cameron, I.T., 2013. Cause-implication diagrams for process systems: their generation, utility and importance. Chem. Eng. Trans. 31, 193–198.

Petković, M., Rapaić, M.R., Jeličić, Z.D., Pisano, A., 2012. On-line adaptive clustering for process monitoring and fault detection. Expert Syst. Appl. 39, 10226–10235.

Qin, S.J., 2012. Survey on data-driven industrial process monitoring and diagnosis. Ann. Rev. Control 36, 220–234.

Rozinat, A., van der Aalst, W.M.P., 2008. Conformance checking of processes based on monitoring real behavior. Inf. Syst. 33, 64–95.

Tóth, A., Werner-Stark, Á., Hangos, K.M., 2014. A structural decomposition-based diagnosis method for dynamic process systems using HAZID information. J. Loss Prev. Process Ind. 31, 97–104.

van der Aalst, W.M.P., van Dongen, B.F., Gunther, C.W., Mans, R.S., Alves de Medeiros, A.K., Rozinat, A., Rubin, V., Song, M., Verbeek, H.M.W., Weijters, A.J.M.M., 2007. ProM 4.0: comprehensive support for real process analysis. In: Kleijn, J., Yakovlev, A. (Eds.), Application and Theory of Petri Nets and Other Models of Concurrency (ICATPN 2007). Lecture Notes in Computer Science, vol. 4546. Springer-Verlag, Berlin, pp. 484–494.

Vedam, H., Venkatasubramanian, V., 1997. Signed digraph based multiple fault diagnosis. Comput. Chem. Eng. 21, 655–660.

Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri, S.N., 2003a. A review of process fault detection and diagnosis. Part I: Quantitative model-based methods. Comput. Chem. Eng. 27, 293–311.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., 2003b. A review of process fault detection and diagnosis. Part II: Qualitative models and search strategies. Comput. Chem. Eng. 27, 313–326.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., Yin, K., 2003c. A review of process fault detection and diagnosis. Part III: Process history based methods. Comput. Chem. Eng. 27, 327–346.

Yin, S., Ding, S.X., Haghani, A., Hao, H., Zhang, P., 2012. A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process. J. Process Control 22, 1567–1581, Elsevier Ltd.

Zadeh, L.A., 1975. Fuzzy logic and approximate reasoning. Synthese. 30 (3-4), 407–428.