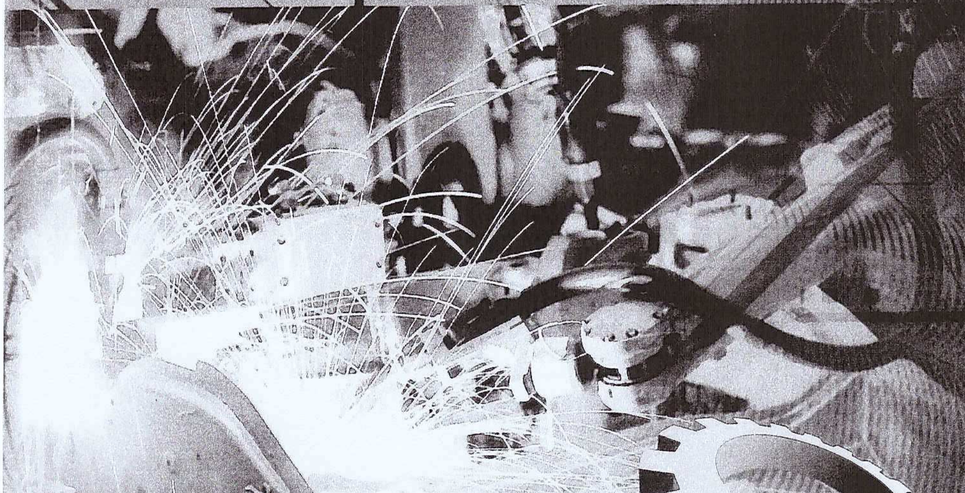
GYÁRTÁS-AUTOMATIZÁLÁS
FACTORY AUTOMATION

2013



MATE
Méréstechnikai, Automatizálási és Informatikai Tudományos Egyesület

University of Pannonia
Veszprém, Hungary

May 21-22, 2013

IEEE

Cover design: Eckert László
Published by University of Pannonia, Veszprém, 2013.
Pressed by Tradeorg
ISBN: 978-615-5044-80-9

A model based diagnosis method for discrete dynamic processes using event sequences

ATTILA TÓTH[1,2,5], KATALIN M. HANGOS[1,2,3], ÁGNES WERNER-STARK[1,4]
[1]University of Pannonia, Department of Electrical Engineering and Information Systems,
Veszprém, HUNGARY
[2]Computer and Automation Research Institute, Process Control Research Group, Budapest, HUNGARY
[3]hangos@scl.sztaki.hu
[4]werner.agnes@virt.uni-pannon.hu
[5]atezs82@gmail.com

Abstract: A novel model-based fault detection and diagnosis method is proposed in this paper that is based on following event sequences measured in a discrete dynamic process. The model of the nominal and faulty operation modes are given in the form of event sequences, that are decomposed according to the components and sub-components present in the process system. The event sequences are defined using extended procedure HAZID tables [4].

Keywords: Process monitoring, diagnosis, discrete event systems, qualitative models

1 Introduction

Fault prevention and mitigation in the field of process system management is a task of crucial importance in avoiding serious accidents. Thus, numerous hazard identification (HAZID) techniques had been developed in the past decades to ensure the safe operation of process systems and to relieve effects of faults (see [1] for a broad presentation of the domain). Among these techniques, the most important methodologies involve the function-driven HAZOP (HAZard and OPerability, see [7]) analysis and the component-driven FMEA (Fault Mode and Effect Analysis). There had been results in the past decade for automating the creation of HAZOP analyses ([6] with a concrete application described in [5]). Blending the component-driven and function-driven analyses also resulted in a novel hazard identification approach described in [3].

Although the information collected in the HAZOP and FMEA studies serve the purpose of hazard identification, these studies can be the basis of diagnostic procedures, too. A model-based diagnostic method based on HAZOP and FMEA information is reported in [2].

It is important to note, that the above techniques concentrate on the static case when the deviation from a normal behavior is of importance. Therefore, the case when the plant is controlled by an operational procedure is not addressed in these results. A recent study ([4]) tries to deal with the diagnostic task by using a specially constructed **P-HAZID** analysis and a diagnostic algorithm. In this paper this diagnostic idea is extended to be able to handle more complex diagnostic tasks - by taking advantage of a possible decomposition of typical process systems along their similar components.

2 Basic notions

2.1 Qualitative range spaces

Current values of continuous outputs in process systems can be described using a properly selected qualitative range space. For example, to describe the value of a level sensor in a tank, the following range space can be used:

$$Q_e = \{\mathrm{e}-, 0, L, N, H, \mathrm{e}+\} \qquad (1)$$

Here 0 means an empty tank, L, N and H means low, normal and high fluid level respectively, while e$-$ and e$+$ refer to unmeasurably low and high fluid levels (this might mean a failure in the level sensor itself). This range space will be used to describe system outputs during operation.

2.2 Input-output event sequences

Operational procedures in process systems are detailed list of instructions for the plant operator personnel to perform certain operations on the plant. Procedures can be formally described using finite input-output event sequences where a single event describes a change in either the inputs or the outputs of the system at a specific time instant. Therefore the syntax of a single input-output event (at time instant t) is the following:

$$event_t = (t; input\ values; output\ values)$$

The inputs in an event always refers to a state of an actuator component in the process system (eg. in the case of a valve it can be **open** (op) or **closed** (cl)). On the other hand, the outputs in an event refer to a value of an output of the process system in the qualitative range space using the qualitative set defined in (1). Sequences formed from these events are called traces and defined as:

$$T(t_1, t_n) = event_{t_1}, ..., event_{t_n}$$

Separate events in a trace contain the same inputs and outputs.

For every operational procedure there exists a trace (called the nominal trace) which describes its behavior under fault-free conditions. The method compares this trace to other traces which may have been executed under faulty conditions (called characteristic traces), and the differences (called deviations) are later used to find possible malfunctions of components in the system.

2.3 Deviations

Nominal and characteristic traces can be compared by comparing their corresponding event fragments. The difference between two corresponding event fragment is described by a deviation. Deviations are formed from a deviation guideword and the nominal event from which the corresponding characteristic trace event is deviating from. The following deviation types are used during diagnosis: .

- **never-happened**: When the particular event never happened in the characteristic trace.
- **later**: When the event happened in the characteristic trace, but at a later time instant.
- **earlier**: When the event happened in the characterictic trace, but at an earlier time instant.
- **greater**: When a particular output's qualitative merit was higher in the characteristic event.
- **smaller**: When a particular output's qualitative merit was lower in the characteristic event.

For the detailed description of the **greater** and **smaller** qualitative relations, please refer to [4].

Cause	Deviation	Implication
TANK-LEAK	NH(2;op,cl;L)	NH(3;op,cl;N)
NH(2;op,cl;L)	NH(3;op,cl;N)	NH(4;op,op;N)
TANK-LEAK	SML(2;op,cl;L)	SML(3;op,cl;N)
SML(2;op,cl;L)	SML(3;op,cl;N)	SML(4;op,op;N)
POS-BIAS	GRE(1;op,cl;0)	GRE(2;op,cl;L)
GRE(1;op,cl;0)	GRE(2;op,cl;L)	GRE(3;op,cl;N)
GRE(2;op,cl;L)	GRE(3;op,cl;N)	NH(4;op,op;N)
POS-BIAS	NH(1;op,cl;0)	EAR(2;op,cl;L)
NH(1;op,cl;0)	EAR(2;op,cl;L)	EAR(3;op,cl;N)

Table 1: A simple example of a **P-HAZID** table. Inputs: op=**open**, cl=**closed**. Outputs: 0=**no**, L=**low**, N=**normal**. Deviations: NH=**never-happened**, LAT=**later**, EAR=**eariler**, SML=**smaller** and GRE=**greater**. Faults: **TANK-LEAK** is the leak of the tank and **POS-BIAS** is the positive bias failure of the tank level sensor.

2.4 Procedure HAZID

As a combination and extension of the widely used FMEA and HAZOP analyses (for details, refer to [4] or [1] and partly [3]), the procedure HAZID or (**P-HAZID**) analysis can be used for fault diagnosis during operational procedures in a given process system. The result of this **P-HAZID** analysis is given in the form of a spreadsheet, and it consists of deviations and possible root causes. Using the initial set of differences (deviations) between the characteristic trace and the nominal trace, the set of possible root causes can be found using simple reasoning. For details, refer to [4]. A simple example of a **P-HAZID** table can be found in Table 1.

The algorithm uses this technique first to find possible **P-HAZID** row(s) to start from (using the set of initial deviations). Then, following the deviation chains defined by these rows, proceeds towards a possible root cause by traversing new rows based on the initial set of deviations. Using this procedure, it may end up at a root cause, or at a row with deviations from which it cannot proceed forward - because they are not contained in the initial set of deviations. The algorithm assumes that the root causes are static, and they happened before the execution of the procedure has begun.

2.5 Component based structural decomposition

The above mentioned fault diagnosis based on the **P-HAZID** analysis is only developed for process systems consisting of different individual components in [4] - the possible redundancy of such systems (e.g. multiple components of the same characteristics) were not taken into account. However, complex process systems in practice can be decomposed into a connected network of more simple components. For example, the process system in Figure 1 can be decomposed into three smaller components each consists of an input and an output valve and a tank. It is possible that some elements are part of multiple subsystems, as in the case of valve VB and VC in Figure 1. These elements are called boundary components, and are assumed to be error-free during the diagnosis. Traces affecting different components can also be also decomposed into a chain of trace fragments each referring to a single component of the trace. Events in such a trace fragment have only a subset of inputs and outputs of the united trace (only the inputs and outputs of the particular component is present in them). Fragments also have information about the next trace fragment (called the next trace), and there is a starting condition (an event) that is associated with them to help the diagnosis. Along with the trace fragment, each component has its own **P-HAZID** spreadsheet associated.

3 Component based diagnosis

Applying the diagnostic approach described in [4] on a decomposed process system, the components can be diagnosed separately against faults, treating them as a whole system during diagnosis. After the separate diagnosis, the root causes can be collected and the resulting set of root causes yields to the set of root causes in the united system. Using the component decomposition, the size of the HAZID information required can be made lower in cases when similar connected subsystems form the process system to be diagnosed. On the other hand, the global deviations need to be converted into component deviations by aligning their time and reducing the inputs and outputs in their associated events to component-level inputs and outputs.

The diagnosis is performed by comparing the whole nominal trace with the characteristic trace, and then distribute the deviations (differences) among the components. Before distributing, time alignment and reduction of input and output states to component level is performed (including the component's boundary elements). After the distribution, component-level diagnosis may begin to explore possible problems on component-level.

The diagnosis starts from the first component, takes the deviations, generates the set of possible root causes from them, and then tries to proceed to the next component by checking the starting condition of the next trace fragment - if there is any. If the start condition is fulfilled, then the diagnosis continues, otherwise it halts. For example, in the case of Figure 1 the second fragment might have a start condition containing a statement about the minimum level of fluid in tank TA, and in the case of the congestion of valve VA no fluid is coming into a system - therefore tank TA is not even filling up to the specified minimum level. In this case the diagnosis stops. The result of the diagnostic algorithm is always the union of identified and non-identified root causes created by the consecutive diagnostic algorithm that runs on the components of the consequent nominal trace fragments.

For reference, the whole diagnostic algorithm is presented as a pseudo-code in Algorithm 1. The algorithm collects all root causes (sets **INC** and **IRC**) given a component decomposition, a starting component, and a possibly faulty characteristic trace.

4 Case study

In the case study the diagnosis procedure for the process system in Figure 1 is used. Every tank may contain no fluid (**no** (0) state), may be low on fluid (**low** (L) state) or might have normal fluid level (**normal** (N) state). Tank level is considered as an *output*. In every time instant the level increases by one "level" (from **no** to **low** or from **low** to **normal**) if fluid is coming through via the input valve but the output valve is closed. Due to the same size of the valves fluid flow out of the system is similar, but in the opposite direction (from **normal** to **low** or from **low** to **no**). The states of the valves (**open** (op) or **closed** (cl)) can be changed by the operator, they are considered as *inputs* of the system. Leak in the tank is assumed to be equal to a size of an open valve.

The operational procedure used is the initial filling of all the three tanks with fluid, and is described in detail in Table 2. The process system can be decomposed into three components, therefore the fill operational procedure can be partitioned into three identical procedure fragments along the component boundaries (VB and VC valves).

This fragment can be observed in Table 3, it has only the subset of inputs and outputs (which are directly related to the particular component - the input and output valve and the tank level). The corresponding component **P-HAZID** table can be found in Table 5 with some of the component faults and deviations leading to them. This **P-HAZID** table is used in the case of all three tanks during diagnosis, after the time instants of deviations are shifted backwards properly.

Algorithm 1 Component-based reasoning procedure

```
INC $\leftarrow \{\emptyset\}$
IRC $\leftarrow \{\emptyset\}$
$component \leftarrow startComponent$
$continue \leftarrow$ true
$shift \leftarrow 0$
while $continue$ do
    COMPONENTDEV $\leftarrow$ COLLECTDEVIATIONS($component,\ chrTrace,\ shift$)
    FDP $\leftarrow$ COLLECTFINALDEVIATIONPAIRS(COMPONENTDEV)
    for all $pair \in$ FDP do
        $startDeviation \leftarrow proj_1(pair)$
        $startImplication \leftarrow proj_2(pair)$
        STEP($startDeviation, startImplication, component.HAZID$)
    end for
    if $component.trace.hasnext$ and
         $component.trace.next.startcondition =$ true then
        $shift \leftarrow length(component.trace) - 1$
        $component \leftarrow$ GETCOMPONENT($component.trace.next$)
    else
        $continue \leftarrow$ false
    end if
end while
function COLLECTDEVIATIONS(component, chrTrace, shift)
    DEV $\leftarrow \{\emptyset\}$
    $nomTrace \leftarrow component.trace$
    for $T := 1$ to $length(nomTrace)$ do
        for all deviation $D$ of $chrTrace$ from $nomTrace$ where ISRELATED($T$,$component$) do
            DEV $\leftarrow$ DEV $\bigcup(T - shift,$ CONVERTTOLOCALDEVIATON$(D))$
        end for
    end for
    return DEV
end function
function COLLECTFINALDEVIATIONPAIRS(DEV)
    $\mathbf{t}^* \leftarrow$ GETLASTTIMEINDEVIATIONSET(DEV)
    return $\{(\mathbf{t}^* - \mathbf{1}, d1) \in \mathbf{DEV}, (\mathbf{t}^*, d2) \in \mathbf{DEV}, (\mathbf{t}^* - \mathbf{1}, d1) \times (\mathbf{t}^*, d2)\}$
end function
procedure STEP(deviation, implication, hazid)
    if $\exists R, deviation = dev_{hazid}(R), implication = imp_{hazid}(R)$ then
        for all $\{R, dev_{hazid}(R) = deviation$ and $imp_{hazid}(R) = implication\}$ do
            if $cause_{hazid}(R) \in$ RC then
                IRC $\leftarrow$ IRC $\bigcup cause_{hazid}(R)$
                return
            else if $cause_{hazid}(R) \in$ DEV and $cause_{hazid}(R) \prec dev_{hazid}(R)\ in$ DEV then
                STEP($cause_{hazid}(R),\ dev_{hazid}(R),\ hazid$)
            else
                INC $\leftarrow$ INC $\bigcup cause_{hazid}(R)$
                return
            end if
        end for
    else
        INC $\leftarrow$ INC $\bigcup cause_{hazid}(R)$
        return
    end if
end procedure
```

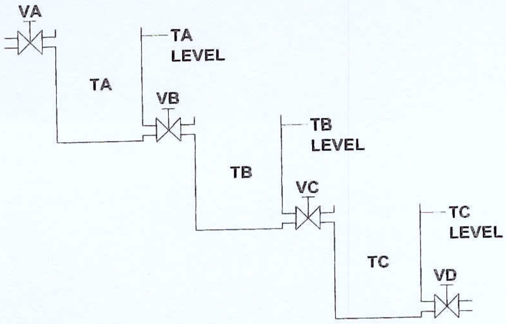


Figure 1: Process system consisting of 3 similar tanks.

An analysis trace about the rupture of the second tank can be seen in Table 4. The size of the leak is bigger or equal to a size of an outbound pipe, therefore the tank cannot fill up and no fluid can be transported to the third tank TC.

The starting condition of the two consequent component's (TB and TC) trace fragment is the appropriate "normal" level in the preceding tank. In that way it can be ensured that diagnosis will be extended to the operational components only.

Diagnosis of this faulty scenario begins by starting with the first tank component TA. There are no differences (and therefore no deviations) regarding this component, and the start condition of the second fragment is fulfilled, the diagnosis moved towards the next component TB. The following deviations are found after comparison (due to that two consequent events did not happened, instead two events happened with lower output values at time instant 4 and 5 in the operational procedure):

- **never-happened**(4;**open**,**closed**;**low**)
- **never-happened**(5;**open**,**open**;**normal**)
- **smaller**(4;**open**,**closed**;**low**)
- **smaller**(5;**open**,**open**;**normal**)

The time instant of the deviations were shifted back by 2 because the second component's first event happens at time instant 3. After that, the diagnosis was initiated on the HAZID table. Searching for the already found deviations and connecting them to possible root causes (as in the case of the original diagnostic idea in [4]) the leak of the second tank could be found. Because of the lack of fluid in the second tank, the start condition of the third fragment is not fulfilled, therefore the diagnostic process halted at this step, resulting in a single probable root cause.

Time	Input values				Output values		
	VA	VB	VC	VD	TA	TB	TC
1	op	cl	cl	cl	0	0	0
2	op	cl	cl	cl	L	0	0
3	op	op	cl	cl	N	0	0
4	op	op	cl	cl	N	L	0
5	op	op	op	cl	N	N	0
6	op	op	op	cl	N	N	L
7	op	op	op	op	N	N	N

Table 2: Tank fill operational procedure.

Input valve	Output Valve	Tank Level
op	cl	0
op	cl	L
op	op	N

Table 3: Normal fill in a single tank with no faults.

Time	Input values				Output values		
	VA	VB	VC	VD	TA	TB	TC
1	op	cl	cl	cl	0	0	0
2	op	cl	cl	cl	L	0	0
3	op	op	cl	cl	N	0	0
4	**op**	**op**	**cl**	**cl**	**N**	**0**	**0**
5	**op**	**op**	**op**	**cl**	**N**	**0**	**0**
6	op	op	op	cl	N	0	0
7	op	op	op	op	N	0	0

Table 4: Tank fill operational procedure with a leak in the second tank TB. The leak caused two different events in the operational procedure related to TB (in **bold**), these differences resulted in the four deviations the diagnosis could start from.

Cause	Deviation	Implication
TANK-LEAK	NH(2;op,cl;L)	NH(3;op,op;N)
TANK-LEAK	SML(2;op,cl,L)	SML(3;op,op;N)
NEG-BIAS	LAT(1;op,cl;0)	NH(2;op,cl;L)
LAT(1;op,cl;0)	NH(2;op,cl;L)	NH(3;op,op;N)
NEG-BIAS	SML(1;op,cl;0)	SML(2;op,cl;L)
SML(1;op,cl;0)	SML(2;op,cl;L)	SML(3;op,op;N)
POS-BIAS	NH(1;op,cl;0)	EAR(2;op,cl;L)
NH(1;op,cl;0)	EAR(2;op,cl;L)	NH(3;op,op;N)
POS-BIAS	GRE(1;op,cl;0)	GRE(2;op,cl;L)
GRE(1;op,cl;0)	GRE(2;op,cl;L)	GRE(3;op,op;N)

Table 5: **P-HAZID** table of a single tank component with two valves for a reference trace of Table 3. Faults: **TANK-LEAK** is leak of the tank, **POS-BIAS** is the positive bias fault of the level sensor and **NEG-BIAS** is the negative bias failure of the level sensor.

5 Conclusion

A novel component-based extension of the single component diagnostic algorithm presented in [4] is described in this paper. Using the extension, the domain of application can be extended to more complex composite process systems. Driven by the decomposition of the overall system into components, the **P-HAZID** tables used for diagnosis are processed at component level by the diagnostic algorithm. The extended method is efficient in the cases when the overall process system consists of similar small components.

The component-based diagnostic procedure was described on a formal level, along with its proposed pseudo-code. A case study for a process system of multiple components and a simple failure was also provided.

6 Future work

The following improvements are planned to extend the component-based diagnostic approach:

- The procedure is based on the assumption that the boundary elements between different components are free of failures. As a future work, this limitation might be removed by using a higher level reasoning above the components (as in a form of a system-level HAZID table, for example).
- At the moment the algorithm is only working for already coded static event information in order to find faults in the system. Diagnosis would me more valuable if events would be processed dynamically, thus the procedure would be executed real-time along with the operational procedures.
- Diagnosis would be more accurate if the derivatives of internal states (eg. the derivative of the tank level) would be present in the events.

Acknowledgements: The research was supported by the Hungarian Research Fund through grant $K83440$. We acknowledge the financial support of the Hungarian State and the European Union under the TAMOP-4.2.2.A-11/1/KONV-2012-0072.

References:

[1] Cameron, I. T., Raman, R. Process Systems Risk Management. Vol. 6 of Process Systems Engineering. *Elsevier Academic Press* San Diego, CA, 2005

[2] E. Németh, R. Lakner, I.T. Cameron and K.M. Hangos, Fault diagnosis based on hazard identification results, *In Preprints of the 7th IFAC Symposium onFault Detection, Supervision and Safety of Technical Processes,* Barcelona, Spain, 2009 June 30 - July 3., pp. 1515-1520.

[3] B. J. Seligmann, E. Németh, K. Hangos and Ian T. Cameron, A blended hazard identification methodology to support process diagnosis, *Journal of Loss Prevention in the Process Industries*, **25**, pp. 746-759.

[4] A. Tóth, K. M. Hangos and Á. Werner-Stark, HAZID information based operational procedure diagnosis method, *In 12th International PhD Workshop on Systems and Control*, Veszprém, 2012. aug. 27. Veszprem (ISBN 978-615-5044-71-7), pp. 1-6 (on CD)

[5] Venkatasubramanian, V., Zhao, J. S., Viswanathan, S., Intelligent systems for HAZOP analysis of complex process plants. *Computers and Chemical Engineering* **24** (9-10), 2000, pp. 2291-2302, doi: 10.1016/S0098-1354(00)005731

[6] Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri, S. N. A review of process fault detection and diagnosis Part I: Qualitative model-based methods. *Computers and Chemical Engineering 27 (3)*, 2003, pp. 293-311, doi: 10.1016/S0098-1354(02)00160-6

[7] SA, 2003. AS IEC 61882-2003: Hazard and operability studies (HAZOP studies) Application Guide. Standards Australia. aS61882.