FAULT DIAGNOSIS BASED ON LIMIT MEASUREMENTS OF PROCESS VARIABLES.

Heinz A Preisig [*], **Yun Xia Xi** [**] and **Khiang Wee Lim** [**]

[*] *Dept of Chemical Engineering, Norwegian University of Science and Technology (NTNU), N-7491 Trondheim, Norway.*
Heinz.Preisig@chemeng.ntnu.no
[**] *National University of Singapore, Singapore.*
elelimkw@nus.edu.sg

Abstract: Industry uses mostly information on measurements passing limits as inputs to diagnostic systems. Asking the question on what can be achieved by such systems, we aim at an optimal design of diagnostic systems. The approach is, in contrast to the currently available techniques, based on continuous models that are mapped into discrete-event dynamic systems in the form of nondeterministic automata, with faults being constraint to have event-dynamic, that is, they occur and persist. We compute domains in which faults can be isolated or detected and discuss guidelines on how to design an application-optimal fault detection and isolation mechanisms.

Keywords: Fault analysis, fault detection, fault isolation, reliability, safety

1. THE PROBLEM BEING STUDIED

Industrial diagnostic systems use as input mostly measurements indicating passing of limits placed on process variables, such as temperature has exceeded value x or pressure is lower than value y. These measurements indicate the limit that is crossed and also the direction in which the process did cross the limit. Currently industry uses mostly diagnostic systems that are built carefully based on process knowledge and experience of the operators. These systems are invariably quite complex and far from easy to build or maintain. The past has seen a number of efforts to improve the situation, which reflects into a rich literature. A wide selection of different methods are reviewed in Gertler (1988), Frank (1990), Patton et al. (1989), Pouliezos and Stavrakakis (1994), Isermann (1997). One of the most common techniques is based on fault tree analysis. Today, it is almost a traditional technique and its development has a rich history Lapp and Powers (1977), Ulerich and Powers (1988), Vries de (1990). With the evolvement of on-line filtering techniques and their extension to parameter estimation, their mostly high sensitivity of the parameters to faults has been utilised for the construction of diagnostic system Isermann (1984), Isermann (1993), Chow and Willsky (1984). The dawn of discrete-event dynamic added over time another viewpoint that yielded alternative design methods, of which examples are reported in Lin (1994), Bavishi and Chong (1994), Sampath et al. (1995), Sampath et al. (1996), Cassandras and Lafortune (1999). Knowledge-based systems are also very popular as they provide a systematic method to capture people's knowledge about the process, experimental or theoretical, into a easy-to-program structure and neural nets provided a matching modelling technique that did not have to rely on mechanistic process knowledge, which was considered too expensive to develop Hoskins and Himmelblau (1988), Venkatasubramanian and Chan (1989), Venkatasubramanian et al. (1990), Maki and Loparo (1997).

Several years ago, we took an alternative route, which grew out of two efforts: one on computer-aided modelling that provides a systematic and easy method to construct mechanistic

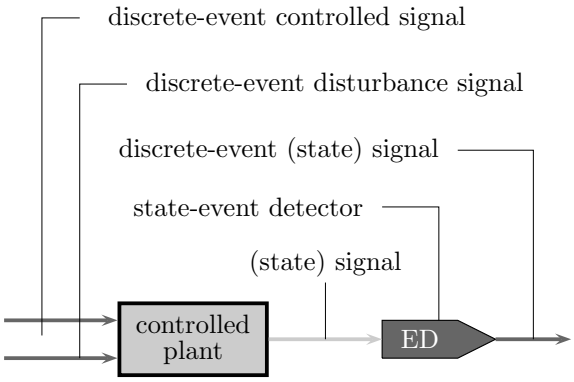

Fig. 1. *The generic plant set-up with a continuous plant, a state-event detection mechanism providing the primitive measures and two discrete-event signal inputs one controlled and the other one not.*

process models, at least to the extent possible, and second the need for safe supervisory control, which was a project that aimed at the design of supervisory systems that are guaranteed complete, meaning including all aspects of the process behaviour on the supervisory level. Quite obviously, mainly the latter induced the question on how to handle faults in such systems, as supervisory control systems are directly linked to diagnostic systems for handling shut-down or recovery. Consequently we posed the question on what can be achieved having only available such limited information as limit crossing and the direction of the crossing; and: can we find design methods for diagnostic systems, a problem that turned out to be at least equally challenging than designing supervisory control systems Philips (2001).

Figure (1) shows the context of plants that we consider, namely a continuous plant, which is affected by fast-switching unobserved disturbances and commands that change the parameters of the controllers in the plant and their setpoints or ask to switch flows or the like.The scheme assumes that the plant is operating in continuous time and that it provides information about the state. Both may raise some questions as the plant's signals are usually sampled and it is not the state that is available, but an estimate of the state being reconstructed using an observer. However, we take the view of a larger time scale in which, for all practical purposes, the plant and its associated measurement and low-level control systems provide essentially continuous state information. The obvious conditions on the relative sampling rate and the delays caused by filtering and reconstruction must be satisfied.

2. WHERE CAN WE DETECT WHAT

A diagnostic system should provide information on faults. Obviously one would like to detect these faults whenever they occur. But then, it is also obvious that this will not be possible for various reasons but mostly because of accuracy of measurements and accuracy of knowing the systems behaviour under different operational modes (normal or various different faults). We shall certainly come back to some of these questions later in this exposition. But first, we shall focus on the ideal world, just to learn on what would be possible and analyse a process model. When we aimed at the design of supervisory control systems, we took a similar approach and derived a method to map continuous plants that are, on the high level, controlled by commands and which generate as output limit crossing signals generated by event detectors. Thus we describe a hybrid system, but choose to map the continuous part controlled by commands and generating discrete event signals as a discrete-event dynamic system Preisig et al. (1997),Philips (2001). In its core, the method maps the continuous state space of the plant into a discrete equivalent, which is a space of hypercubes defined by the limits, which, in turn, define the event detectors operation.

Mathematically, let us choose the plant as being described by:

$$\begin{aligned}\dot{\underline{\mathbf{x}}} &:= \underline{\mathbf{f}}(t, \underline{\mathbf{x}}, \underline{\mathbf{u}}_c, \underline{\mathbf{d}}_c) \\ &\underline{\mathbf{x}} \in \mathbb{R}^n, \quad \underline{\mathbf{d}}_c \in \mathbb{R}^m, \quad \underline{\mathbf{u}}_c \in \mathbb{R}^p\,. \end{aligned} \tag{1}$$

with the two vector signals $\underline{\mathbf{u}}_c$ and $\underline{\mathbf{d}}_c$ being piecewise constant signals for the command input and the persistent fault signals and $\underline{\mathbf{f}}(t, \underline{\mathbf{x}}, \underline{\mathbf{u}}_c, \underline{\mathbf{d}}_c)$ a vector of analytical functions. The plant is observed by a set of event detectors attached to the individual state measurements, or reconstruction in the case they are not directly attachable, each defined by a set of limits:

$$\mathcal{B}_i := \left\{\beta_i^1, \ldots, \beta_i^{b_i}, \ldots \beta_i^{n_i}\right\}$$

then the continuous state space is mapped into a discrete space of hypercubes representing the discrete space:

$$\mathcal{H}(x) := \left\{[x_i]_{\forall i} \mid \beta_i^{b_i - 1} \leq x_i \leq \beta_i^{b_i}\right\}$$

An event is, in this context, defined as a crossing of a face of a hypercube, that is, a boundary is crossed. The dynamics of the discretely-controlled and discretely-observed plant is a non-deterministic automaton Preisig et al. (1997), Philips (2001). The crossing of the boundary may only occur if there exists at least one point on the face having a gradient that points across the boundary. Requesting a reasonable kind behaviour of the system, that is in the simplest case, analytical functions describing the dynamics, the directionality of the gradient in a particular direction changes on a hypersurface defined by the particular component being zero. This hyper surface is defined by the expression:

$$\dot{x}_i := f_i(t, \underline{\mathbf{x}}, \underline{\mathbf{u}}^k, \underline{\mathbf{d}}^l) := 0\,. \tag{2}$$

Assuming, which is not very limiting, that these hypersurfaces are reasonably kind, preferably monotone, one sees quickly that they split the state space into two parts, one in which all boundaries defined for this state variable are crossed in positive direction, and one in which it is crossed in negative direction, which are the measurements we have available.

3. DIAGNOSABILITY

With this result, we know where a particular measurement, the signal indicating limit crossing and its direction, gives us information about the dynamics of the process. Diagnosability tries to distinguish between different plant behaviours, which above we termed modes of operation. The task now is to find the piece of the state space in which such rudimentary process information yields information about the mode of operation. Mathematically spoken we seek a subspace in which the behaviour of the plant operating under *mode a* behaves differently from the plant operating under *mode b* such that it is uniquely observable with the given measurement. All combinations of inputs are defined as instances of the discrete command and disturbance vectors. For the command (control) vector the running index $k \in \mathcal{K} \subset \mathbb{N}^+$ is used and for the disturbances it is the letter $l \in \mathcal{L} \subset \mathbb{N}^+$ both shown as superscript, in distinction to the vector component index which is shown as subscript. Let $\mathcal{N}_i$ be the index set of the states being coupled with state i and $\mathcal{L}_i$ the index set of the disturbances being coupled with state i we first define the space:

$$\mathcal{X}_{i,s}(k,l) := \left\{ \underline{\mathbf{x}} \in \mathcal{V} \mid s := \text{sign}\left(f_i(t, \underline{\mathbf{x}}, \underline{\mathbf{u}}^k, \underline{\mathbf{d}}^l) \right) \right\} \tag{3}$$

in which the i^{th} component of the gradient assumes the sign s and next

$$\begin{aligned} \mathcal{O}_{i,s}(k, \mathcal{A}_r) := & \left(\bigcap_{\forall j \in \mathcal{A}_r} \mathcal{X}_{i,s}(k,j) \right) \cap \\ & \cap \left(\bigcap_{\forall j \in \{\mathcal{L}_i - \mathcal{A}_r\}} \mathcal{X}_{i,-s}(k,j) \right) \\ & \forall i; \forall r\,. \end{aligned} \tag{4}$$

with

$$\begin{aligned} \mathcal{A} := \{\mathcal{A}_r\} &:= \{\mathcal{T}_0, \mathcal{T}_j\} \cup \mathcal{F}\,, \\ \mathcal{F} := \{\mathcal{F}_r\} &:= \{\{\mathcal{T}_{j_1,j_2}\}, \{\mathcal{T}_{j_1,j_2,j_3}\}, \dots\}\,, \\ \mathcal{T}_{j_1,j_2} &:= \{j_1, j_2 | j_1 \neq j_2; j_1, j_2 \in \mathcal{L}_i\}\,, \\ \mathcal{T}_{j_1,j_2,j_3} &:= \{j_1, j_2, j_3 | j_1 \neq j_2 \neq j_3; j_\cdot \in \mathcal{L}_i\}\,, \\ \dots &:= \dots\,, \\ \mathcal{T}_0 &:= \{0\}\,, \\ \mathcal{T}_j &:= \{j \in \mathcal{L}_i\}\,, \end{aligned}$$

with

$$\mathcal{L}_i := \left\{ j \;\middle|\; \frac{\partial f_i(t, \underline{\mathbf{x}}, \underline{\mathbf{u}}^k, \underline{\mathbf{d}}^l)}{\partial {d_j}^l} \neq 0 \right\}. \tag{5}$$

being the set of persistent fault inputs that are coupled with state i. The index set $\mathcal{T}_0$ represents the no-fault case, whilst the sets $\mathcal{T}_j$ each having only one element are for the jthfaults. The other test sets $\mathcal{T}_{\{j_1,\dots\}}$ are for groups of faults. The non-empty subspaces $\mathcal{O}_{i,s}(k, \mathcal{A}_r)$ have the properties that gradient information in the x_i-direction is sufficient information to diagnose the plant for the cases listed in Table 1, which is what we were seeking.

		s detected	-s detected
$\exists$	$\mathcal{O}_{i,s}(k, \mathcal{T}_0) \neq 0$	no fault	$\{j\} \subset \mathcal{L}_i$
$\exists$	$\mathcal{O}_{i,s}(k, \mathcal{T}_j) \neq 0$	fault j	not fault j
$\exists$	$\mathcal{O}_{i,s}(k, \mathcal{F}_r) \neq 0$	$\{j\} \subset \mathcal{F}_r$	$\{j\} \subset \mathcal{L}_i - \mathcal{F}_r$

Table 1. *Different types of overlapping subspaces for the i^{th}component*

Now that tells us on what we can achieve, our main result. How about, though, the design of diagnostic units operating on such measurements?

4. DESIGN ISSUES

If the automaton to be constructed serves the purpose of fault detection & and fault isolation, then the boundaries that make the state-event detector, are to be place into the subspaces $\mathcal{O}_{i,s}(k, \mathcal{A}_r)$ for each component. The computation of the automaton is solved, as the procedure for the computation of the non-deterministic automaton is given (Preisig et al. (1997), Philips (2001)). Remains the question on how precisely to place the boundaries into these subspaces. The question cannot be answered in a deterministic manner, as it is a true design issue. Why? The automaton operates in a square world, with the size of the hypercubes defining a type of resolution for the detection. If we make the resolution high, the detection will operate on this high resolution and if it is low it will be correspondingly operating on the low resolution. It is the designer's choice and must be based on the dynamics of the process and the significance of the fault to be detected. It involves such questions as how quickly should it

be detected and how much of faulty behaviour can be tolerated under various conditions. De facto, the designer gets the information on which part of the state space he is to approximate with the automaton and he is left with the decision on the fidelity of the approximation. The automaton, as it is described in Preisig et al. (1997) and Philips (2001) does not provide any timing information but first results on computing minimal and maximal transition times are now available Preisig et al. (2002), though for monotone plants only or for parts of the state space that exhibit monotone behaviour. Work on non-monotone plants is currently in progress.

There are two additional obvious issues to be mentioned. Firstly, the design of a diagnostic system is based on a process model. Consequently the diagnostic system will not be designed to detect faults that are not modelled, but then this is not the case with any of the diagnostic system; none can operate and act on information it does not have. The second limitation is the fact that the measurement is sensitive to noise. It must be assumed that the detection of the event and consequently the detection of the direction is essentially done with certainty thus with a probability that is near to 1. This raises certainly questions on the implementation, but then effects of noise are always present independent on the behaviour of the diagnostic system. We prefer to separate these question, being aware that we deal with dynamics in a certain time scale, which is limited and which allows us to design appropriate state reconstruction and filters.

Combination of faults can easily be handled as they can simply be defined as additional operating modes. The set-up of the initial description is design to handle this complexity. Further, the knowledgeable reader may point out that the dimensionality of the discrete state space grows combinatorially with the number of state variables and the number of boundaries being defined for the individual state variables. Whilst the basic observation is correct, the reality shows that the systems are almost always very sparse and the at the individual functions (equations 2) are very weekly coupled in the state space and the input spaces indeed. Thus since the automaton is based on these local measurements, it is always locally of small dimension: the two sets $\mathcal{N}_i$ and $\mathcal{L}_i$ are usually rather small. Thus the state explosion argument is not applicable.

5. EXAMPLE

The sample plant consists simply of two tanks standing side by side with a feed of fluid that is driven by a pump into the first tank and a pipe connecting the two tanks at the bottom. Once the pump is running, the plant is obviously not stable as it consists of two coupled, pure integrators.

For a given set of parameters, the subspaces are shown in Figure 5 with the levels (volumes or masses) of the two tanks as the respective co-ordinates.Table 2 and the Table 3 list the

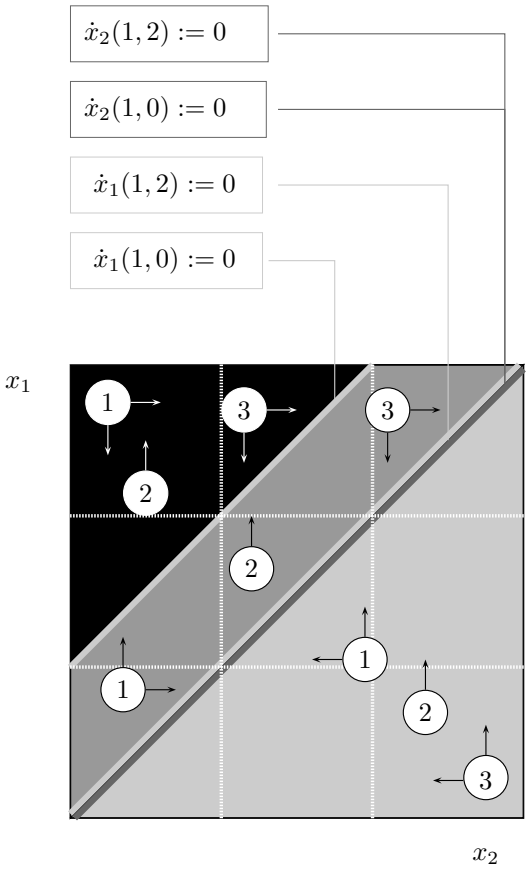

Fig. 2. *Phase diagram of two tank system for case 1-3. (Case 1: pump on, pipe open, pump running; case 2 : pump on, pipe blocked, pump running; case 3: pump on, pipe open, pump failing.)*

component-equilibrium surfaces, here lines, and the different subspaces for the two signs and the two co-ordinates. From the analysis, it is evident that only one co-ordinate (level, volume, mass of tank 1) provides information for diagnoses if one does not use transition time in which case the fault of a blocked pipe could be detected as not generating a transition in the x_2 direction. Otherwise, it is the stripe in the middle in which it is case 3 that shows a negative sign whilst the other two cases would move in positive direction. Moving across a boundary in the negative x_1-direction isolates the fault of the pump not being on.

The second fault that can be isolated using directional information on state component x_1 is detectable in the upper-left triangle in which it is case 2, which is the only one that moves in positive direction for this operation mode,

case	k	l	u_1	d_1	d_2	x_1^0	dyn	x_2^0	dyn
1	1	-	1	0	0	$x_2 + \frac{p_2}{p_1}$	+	x_1	+
2	1	1	1	1	0	$x_1 \rightarrow x_1^+$	−	$x_2^- \leq x_2 \leq x_2^+$	0
3	1	2	1	0	1	x_2	+	x_1	+
4	2	-	0	0	0	x_2	+	x_1	+
5	2	1	0	1	0	$x_1^- \leq x_1 \leq x_1^+$	0	$x_2^- \leq x_2 \leq x_2^+$	0
6	2	2	0	0	1	x_2	+	x_1	+

Table 2. *Component equilibrium surfaces for all cases (+ indicates stable, - unstable and 0 no dynamics for the respective component)*

case	k	l	u_1	d_1	d_2	$\mathcal{X}_{1,-1}(k,l)$	$\mathcal{X}_{1,+1}(k,l)$	$\mathcal{X}_{2,-1}(k,l)$	$\mathcal{X}_{2,+1}(k,l)$
1	1	-	1	0	0	$x_1 > x_2 + 1$	$x_1 < x_2 + 1$	$x_2 > x_1$	$x_2 < x_1$
2	1	1	1	1	0	0	$x_1 > {x_1}^-$	0	0
3	1	2	1	0	1	$x_1 > x_2$	$x_1 < x_2$	$x_2 > x_1$	$x_2 < x_1$
4	2	-	0	0	0	$x_1 > x_2$	$x_1 < x_2$	$x_2 > x_1$	$x_2 < x_1$
5	2	1	0	1	0	0	0	0	0
6	2	2	0	0	1	$x_1 > x_2$	$x_1 < x_2$	$x_2 > x_1$	$x_2 < x_1$

Table 3. *Subspaces for both co-ordinates, each case and both signs*

whilst for the other two operation modes the gradient in this direction is negative. Thus a move across a boundary in this the upper-left triangular subspace in positive x_1 direction indicates the fault "blocked pipe".

6. CONCLUSIONS

Insight gained on modelling the discretely-controlled and discretely-observed continuous plant section of a hybrid system as a non-deterministic automaton gives valuable insight into what can be achieved with simple measurements of state-variable limit crossing and direction of the crossing.

The key is to analyse the flow of the dynamic behaviour in the continuous domain under the different operating modes and seek the parts of the state space in which directional information is sufficient to diagnose faulty behaviours.

The design of diagnostic systems focusing on individual or any combination of faults are only constraint by the ability to model the plant behaviour under any combination of faulty conditions and the accuracy with which the fault is to be detected, both in state space as well as in time.

Since the dynamic equations are sparsely coupled and since we analyse the behaviour component by component, the state dimension remains quite small and the often in this context cited dimension-explosion problem does simply not occur.

References

S Bavishi and K P E Chong. Automated fault diagnosis using a discrete event systems framework. *IEEE Int Symp Intelligent Control*, pages 213–218, 1994.

C G Cassandras and S Lafortune. Discrete event systems: The stae of the art and new directions. *Applied and Computational Control Signals and Circuits*, 1:34–42, 1999.

E Y Chow and A S Willsky. Analytical redundancy and the design of robust faulure detection systems. *IEEE Transaction on Automatic Control*, pages 603–614, 1984.

P M Frank. Fault diagnosis in dynamic systems unsing analytical and knowledge-based redundancy – a survey and some new results. *Automatica*, 26(3):459–474, 1990.

J J Gertler. Survey of model-based failure detection and isolation in complex plants. *IEEE Control System Magazine*, 6(8):3–11, 1988.

J C Hoskins and D M Himmelblau. Artificial neural network models of knowledge representation in chemical engineering. *Computers & Chemical Engineering*, 12:881–890, 1988.

R Isermann. Process fault detection based on modelling and estimation methods – a survey. *Automatica*, 20(4):387–404, 1984.

R Isermann. Fault diagnosis of machines via parameter estimation and knowledge processing. *Automatica*, 29(4):815–835, 1993.

R Isermann. Supervision, fault-detection and fault-diagnosis methods – and introduction. *Control Engineering Practice*, 5(5):639–652, 1997.

S Lapp and G Powers. Computer-aided synthesis of fault trees. *IEEE Transaction Reliability Engineering*, 26:2–13, 1977.

F Lin. Diagnosablility of discrete event systems and its application. *Journal of Discrete Event Dynamic Systems*, 4(2):197–212, 1994.

Y Maki and K A Loparo. A neural network approach to fault detection and diagnosis in

industrial processes. *IEEE Transaction Control System Technology*, 5(6):529–541, 1997.

R Patton, P M Frank, and R Clark. *Fault diagnosis in dynamic systems: theory and applications*. Prentice Hall, Englewood Cliffs, NJ, 1989.

P P H H Philips. *Modelling, Control and Fault Detection of Discretely-Observed Systems*. PhD thesis, TU Eindhoven, Eindhoven, The Netherlands, 2001.

A D Pouliezos and G S Stavrakakis. *Real-time fault monitoring of industrial processes*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

H A Preisig, K W Lim, and Y X Xi. Computation of min max transition times in automata represting discrete-event observed continous, monotone plants. *ASCC 2003*, (TM 9-23):1658, 2002.

H A Preisig, M J H Pijpers, and M Weiss. A discrete modelling procedure for continuous processes based on state- discretization. *MATHMOD 2*, pages 189–194, 1997.

M Sampath, R Sengupta, S Lafortune, K Sinnamohideen, and D Teneketzis. Diagnosablility of discrete event systems. *IEEE Transaction on Automatic Control*, 40(9):1555–1575, 1995.

M Sampath, R Sengupta, S Lafortune, K Sinnamohideen, and D Teneketzis. Failure diagnosis using discrete-event models. *IEEE Transaction Control System Technology*, 4(2):105–124, 1996.

N Ulerich and G Powers. On-line hazard aversion and fault diagnosis in chemical processes: The digraph + fault-tree method. *IEEE Transaction Reliability Engineering*, 37(2):171–177, 1988.

V Venkatasubramanian and K Chan. A neural network methodology for process fault diagnosis. *AIChE Journal*, 35(‘1):1993–2001, 1989.

V Venkatasubramanian, R Vaidyanathan, and Y Yamamoto. Process fault detection and diagnosis using neural networks. *Computers & Chemical Engineering*, 14(7):699–712, 1990.

R Vries de. An automated methodology for generating a fault tree. *IEEE Transaction Reliability Engineering*, 39:76–86, 1990.