Qualitative modelling of a power stage for diagnosis

Dirk Förstner
Robert Bosch GmbH
Abteilung FV/FLI, Postfach 106050
D-70049 Stuttgart, Germany
Foerstner@fli.sh.bosch.de

Jan Lunze
Technische Universität Hamburg-Harburg
AB Regelungstechnik, Eissendorfer Str.40
D-21071 Hamburg, Germany
Lunze@tu-harburg.de

Abstract

The paper deals with the representation of continuous-variable systems whose state can only be measured through a quantiser. The model provides a discrete-event description which is suitable for consistency-based diagnosis. The paper describes which information about the quantised system the model includes and how the discrete-event model can be abstracted from a quantitative model. The results are applied to the fault diagnosis of an electronic power stage which supplies an electromagnetic valve of a car engine and consists of digital and analogue components.

Introduction

Model-based approaches for process monitoring and diagnosis are based on explicit knowledge about a dynamical system. Due to this, the development of a diagnostic system can be separated into the construction of a general diagnostic algorithm and the modelling of the given system. Different approaches have been elaborated in the fields of control engineering or artificial intelligence, cf for surveys (Frank 1996), (Hamscher, Console, and de Kleer 1992), (Isermann 1984), (Lunze 1995), and (Patton, Frank, and Clark 1989). This paper focuses on the modelling step. A qualitative modelling approach is proposed, which should be applied to continuous-variable dynamical systems under the following practical circumstances:

- Under strict real-time restrictions continuous signals often cannot be processed fast enough (the cycle time of the power stage example described in this paper is between 10 and 100 milliseconds). Therefore, qualitative considerations are necessary which abstract from detailed information.
- The model has to be used to find faults that change the behaviour of the given system qualitatively (eg Figure 5). Then, qualitative considerations have sufficient information to discriminate correct and faulty behaviours.

The qualitative view of this paper is illustrated by Figure 1. The input and output signals of the quantitative system are only measured through quantisers. A quantiser generates an event e each time the signal changes its qualitative value. Instead of continuous signals $\boldsymbol{u}(t)$ and $\boldsymbol{x}(t)$ only discrete-event sequences $\boldsymbol{E}$ are obtained. The continuous-variable system together with the quantisers is called the "quantised system".

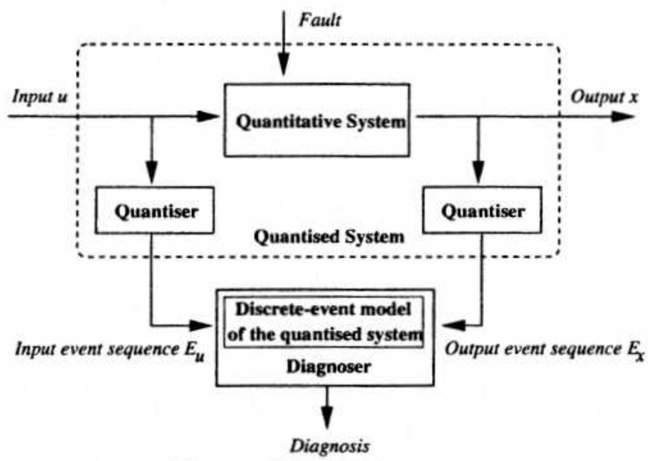

Fig. 1: Quantised system

The qualitative modelling problem treated in this paper concerns two tasks:

- First, a discrete-event model suitable to represent a quantised system for diagnostic tasks is defined. By using the ideas of (Lunze, Nixdorf, and Schröder 1999), (Raisch et al. 1998), and (Sampath et al. 1996) a nondeterministic automaton is used as qualitative model. The automaton captures the discrete-event dynamics of the quantised system by representing the occurrence of an event as state transition of the automaton.
- Second, an abstraction algorithm for the construction of the qualitative model is described. The main result of the paper concerns the relation between the abstracted discrete-event representation by means of an automaton and the quantised continuous-variable system.

The modelling problem considered here has been dealt with in the literature on qualitative modelling by, for example, (Antsaklis, Stiver, and Lemmon 1996), (Lunze 1995), (Lunze, Nixdorf, and Schröder 1999), (Raisch et al. 1998). The main difference to this work results from the pure discrete–event point of view adapted in this paper. That is, the time points considered are determined by the quantiser (and not by a sampler as in the discrete–time systems literature). While approaches as (Dousson, Gaborit, and Ghallab 1993) capture cuts of the discrete–event behaviours in so–called situations, the automaton proposed in this paper represents event sequences in a recursive manner.

The diagnostic algorithm is derived from the results of (Lunze 1999). It is used here mainly to demonstrate the usefulness of our qualitative model.

The paper is organised as follows. The quantised system is introduced in the next section. The following section defines the automaton for the representation of the discrete–event behaviour of quantised systems and specifies the modelling requirements. Then, the construction of the model by abstraction is presented and proved to meet the modelling requirement (Theorem 1). The subsequent section proposes a consistency-based diagnostic algorithm that uses the discrete–event model. Finally, the approach is demonstrated for the diagnosis of an industrial application example.

The quantised system

The continuous–variable system

This paper concerns dynamical systems that can be described by the differential equation

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), f), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0. \tag{1}$$

where the behaviour of the state vector $\boldsymbol{x} \in \mathbb{R}^n$ is determined by the input vector $\boldsymbol{u} \in \mathbb{R}^m$ and the system function $\boldsymbol{f}$. The dynamic behaviour of the system depends on the fault $f \in \mathcal{F} = \{f_0, \ldots, f_{n_F}\}$ where f_0 denotes the faultless system.
Because the main ideas can be demonstrated for autonomous systems, the system

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), f), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{2}$$

without inputs will be considered in the following. All signals of the systems are assumed to be observable which is reflected in the fact that no difference is made between state signals $\boldsymbol{x}$ and output signals $\boldsymbol{y} = \boldsymbol{x}$.

Spatial and temporal quantisation

Spatial quantisation. The state quantiser (Figure 1) can be represented by a partition of the state space $\mathbb{R}^n$ into the sets $\mathcal{Q}_x(i) \subset \mathbb{R}^n$, $i \in \mathcal{N}_u = \{0, 1, \ldots, N\}$. The qualitative value of the state $\boldsymbol{x}(t)$ at time t is given by the index i of the set $\mathcal{Q}_x(i)$ to which the state belongs:

$$[\boldsymbol{x}(t)] = i \iff \boldsymbol{x}(t) \in \mathcal{Q}_x(i). \tag{3}$$

Thus, only the quantised state $[\boldsymbol{x}]$ is assumed to be measurable whereas the measurability of the quantitative state $\boldsymbol{x}$ is not necessary. In case a signal is not observable at all, a single partition may cover the whole value range of this signal. The approach is then suitable for systems whose output signals $\boldsymbol{y}$ include only a subset of the input signals $\boldsymbol{u}$ as well.

Temporal quantisation. An event $e_{ji} \in \mathcal{E}$ takes place at time t_e if the qualitative value of the state $\boldsymbol{x}$ changes from the value i to j:

$$e_{ji} := (i, j), \quad i \neq j \tag{4}$$

$$[\boldsymbol{x}(t_e - \delta)] = i\,,\ [\boldsymbol{x}(t_e + \delta)] = j\,, \quad \delta \to 0. \tag{5}$$

At the event time t_e the trajectory $\boldsymbol{x}(t)$ passes the border $\delta\mathcal{Q}_x(i, j)$ between the partitions $\mathcal{Q}_x(i)$ and $\mathcal{Q}_x(j)$. In an event sequence $\boldsymbol{E}(0 \ldots H) = (e_0, e_1, e_2, \ldots, e_H)$ the events are numbered where e_k denotes the k–th event and $H = |\boldsymbol{E}|$ is the number of events that the system generates within a given time horizon. If the quantised system is considered in the time interval $[0, T_h]$, the continuous–variable system follows the trajectory $\boldsymbol{x}_{[0,T_h]}$. The relation between a continuous state trajectory $\boldsymbol{x}_{[0,T_h]}$ and the event sequence $\boldsymbol{E}$ is given by the quantiser. This fact is reflected in the equation

$$\boldsymbol{E}(0 \ldots H) = \text{Quant}\,(\boldsymbol{x}_{[0,T_h]}). \tag{6}$$

Behaviour of the quantised system

For a given time interval $[0, T_h]$, a fault f and an initial event e_0 the quantised autonomous system behaviour is defined as a set of event sequences

$$\mathcal{S}(e_0, f) = \{\boldsymbol{E} = \text{Quant}\,(\boldsymbol{x}_{[0,T_h]}) \mid \ \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), f), \ \boldsymbol{x}_0 \in \delta\mathcal{Q}_x(e_0)\}. \tag{7}$$

The qualitative system behaviour $\mathcal{S}(e_0, f)$ denotes the event sequences that are generated by the quantisation of possible state trajectories. Only the initial event e_0 is assumed to be known. Therefore, the initial condition $\boldsymbol{x}_0$ in equation (1) is uncertain and only constrained to the set of points on the border $\delta\mathcal{Q}_x(e_0)$ between two partitions. Due to this uncertainty a set of trajectories has to be considered, hence a set of different event sequences. For that reason the qualitative system behaviour is nondeterministic in general.
The function $p_{\mathcal{S}}(e, k, f) \in \{0, 1\}$ describes whether an event e may occur at step k:

$$p_{\mathcal{S}}(e, k, f) = \begin{cases} 1 & \text{the event } e \text{ may occur at step } k \\ 0 & \text{else} \end{cases} \tag{8}$$

The value of $p_{\mathcal{S}}(e, k, f)$ is related to the quantised system behaviour $\mathcal{S}(e_0, f)$ by:

$$p_{\mathcal{S}}(e, k, f) = 1 \quad \Leftrightarrow \quad \exists \boldsymbol{E} \in \mathcal{S}(e_0, f) \ : \ e_k = e. \tag{9}$$

The aim of the following modelling step is to represent the quantised behaviour (7) in an efficient and compact manner. A collection of all possible event sequences would not meet this constraint, neither the use of the system equation (2) together with the inequality representing the quantiser.

Discrete-event modelling of quantised systems

Representation of the quantised systems by a nondeterministic automaton

The quantised system is represented by a nondeterministic automaton

$$N = (\mathcal{Z}, \mathcal{F}, L, z_0) \tag{10}$$

where $\mathcal{Z}$ denotes the set of model states, $\mathcal{F}$ the set of faults, L the state transition relation and z_0 the initial model state. A nondeterministic model has been chosen due to the nondeterministic behaviour of the quantised system that is caused by the state quantisation.
In (Lunze 1994) and (Lunze, Nixdorf, and Schröder 1999) the automaton state $z \in \mathcal{Z}$ denotes the quantised state $[\boldsymbol{x}]$ or alternatively an event e. In the following this approach is generalised. Each automaton state represents an event sequence $\boldsymbol{E}(1 \dots d)$ which consists of a length-d sequence of events ("model depth" d). Each event sequence (that is each model state) is interpreted as the sequence of last events that occurred in the quantised system. This idea has been described by (Raisch et al. 1998) for time–discrete sequences of qualitative states and is used here for sequences of events.
Initially, only the single event e_0 is given. Afterwards a sequence of events can be observed. In order to trace the behaviour also before d events occurred, automaton states are added to $\mathcal{Z}$ that represent event sequences of a length less than d. The set of model states is thus defined as:

$$\mathcal{Z} = \{z_1, \dots, z_M\} = \bigcup_{h=1 \dots d} \{(e_1, \dots, e_h) \mid e_i \in \mathcal{E}\}. \tag{11}$$

In contrast to approaches with model depth $d = 1$ where each automaton state represents only the last event, the extension to $d > 1$ yields the possibility to reduce the set of spurious solutions that qualitative models generally produce.
The dynamical behaviour of the nondeterministic automaton is described by the transition relation

$$L : \mathcal{Z} \times \mathcal{Z} \times \mathcal{F} \;\rightarrow\; \{0, 1\}. \tag{12}$$

For $L(z', z, f) = 1$ the automaton can step from model state z to z' if the fault f is present. If the model performs a state transition it generates an event e. This event is stored as the last element of the new state's event sequence. In the following, this last element of a sequence is symbolised by $e = \text{lastevent}(z)$.

Behaviour of the model

The behaviour of the automaton is given by the set of all possible model state sequences

$$\mathcal{M}_z(e_0, f) = \{ (z_0, z_1, \dots, z_H) \mid L(z_{i+1}, z_i, f) = 1, \; z_0 = (e_0)\}. \tag{13}$$

To compare the model behaviour with the behaviour of the quantised system (7) the set of model state sequences has to be mapped on a set of event sequences:

$$\begin{aligned} \mathcal{M}(e_0, f) = \{ \boldsymbol{E} = (e_0, e_1, \dots, e_H) \mid & \\ \exists (z_0, z_1, \dots, z_H) \in \mathcal{M}_z(e_0, f) : & \\ e_i = \text{lastevent}(z_i) \} & \end{aligned} \tag{14}$$

The description whether a model state z occurs at step k in the model behaviour and its relation to the set (13) is defined in the same way as in (8) and (9):

$$p_{\mathcal{M}}(z, k, f) = \begin{cases} 1 & z \text{ may occur at step } k \\ 0 & \text{else} \end{cases} \tag{15}$$

$$p_{\mathcal{M}}(z, k, f) = 1 \;\Leftrightarrow\; \exists (z_0, z_1, \dots, z_k) \in \mathcal{M}_z(e_0, f) : \; z_k = z. \tag{16}$$

The automaton possesses the Markov property as its state captures all information of the past needed to predict the future events. Therefore, the behaviour set $\mathcal{M}_z(e_0, f)$ and the value of the function $p_{\mathcal{M}}(z, k, f)$ can be determined recursively if the initial event e_0 or the initial model state $z_0 = (e_0)$ is known, that is, for

$$p_{\mathcal{M}}(z, 0, f) = 1 \quad \Leftrightarrow \quad z = z_0 = (e_0). \tag{17}$$

The value of $p_{\mathcal{M}}(z, k+1, f)$ only depends on the preceding value $p_{\mathcal{M}}(z, k, f)$ and the state transition relation L of the Markov model:

$$p_{\mathcal{M}}(z, k+1, f) = 1 \;\Leftrightarrow\; \sum_{\tilde{z} \in \mathcal{Z}} L(z, \tilde{z}, f) \cdot p_{\mathcal{M}}(\tilde{z}, k, f) \; > 0 \tag{18}$$

If $p_{\mathcal{M}}(z, k, f)$ is known it can be determined whether an event e can occur at step k.

$$p_{\mathcal{M}}(e, k, f) = 1 \quad \Leftrightarrow \quad \exists z : \; e = \text{lastevent}(z), \; p_{\mathcal{M}}(z, k, f) = 1. \tag{19}$$

In this way, possible event sequences of any length can be produced by the automaton.

Modelling aim

Now it is possible to investigate the relation between $p_{\mathcal{M}}(e, k, f)$ and $p_{\mathcal{S}}(e, k, f)$. Although the continuous–variable state of the system obeys the Markov property, the quantised state does not (Lunze 1994). In contrast, the automaton possesses the Markov property. Due to this, an equality of the quantised system behaviour and the model behaviour is not possible in general. Therefore, the modelling aim is given by

$$\boxed{\mathcal{M}(e_0, f) \supseteq \mathcal{S}(e_0, f)} \tag{20}$$

which demands that all possible event sequences of the quantised system are included in the model behaviour. For most quantised systems the equality sign is not achievable. Instead, spurious solutions $\mathcal{M}(e_0, f) / \mathcal{S}(e_0, f)$ are generated by the model. These solutions can not occur in the quantised system.

Determination of the model by abstraction

Abstraction algorithm

The main question concerning the construction of the automaton model is how to determine the state transition relation L in order to satisfy the modelling aim (20). We propose the following model construction that is based on the abstraction of the quantitative system:

Given: System (2), Quantisers (6), Model depth d.

1. Initialise the transition relation L to zero:

For all tripels (z', z, f) with $z \in \mathcal{Z}$, $z' \in \mathcal{Z}$, $f \in \mathcal{F}$:

$$L(z', z, f) := 0. \tag{21}$$

2. Determine possible transitions:

For all tripels (z, e', f) with $z \in \mathcal{Z}$, $e' \in \mathcal{E}$, $f \in \mathcal{F}$:

2.1. Construct a succeeding model state z' by appending the new event e' to the event sequence represented by the model state z:

$$z = (e_1, \ldots, e_h) = (e_1, \hat{\boldsymbol{E}}). \tag{22}$$

$$\begin{array}{l} \text{if } h = d: \; z' := (\hat{\boldsymbol{E}}, e') \\ \text{if } h < d: \; z' := (e_1, \hat{\boldsymbol{E}}, e'). \end{array} \tag{23}$$

2.2. Check whether the transition from z to z' is possible under the given dynamics:

$$\begin{array}{ll} \text{if} & \exists \boldsymbol{x}_0, T_h : \\ & \text{Quant}\,(\boldsymbol{x}_{[0,T_h]} \mid \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), f), \\ & \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0) \\ & = (e_1, \ldots, e_h, e') \\ \text{then} & L(z', z, f) := 1. \end{array} \tag{24}$$

Result: Model transition relation $L(z', z, f)$.

The algorithm starts with a model transition relation $L \equiv 0$ (21). Then it determines cases where $L(z', z, f) = 1$. To determine such transitions, at first the set of compatible successors for each modelstate $z \in \mathcal{Z}$ is determined (step 2.1). The succeeding state z' keeps the $d-1$ last events of the preceding state z, where d denotes the model depth. If there are not yet $d-1$ events represented by the preceding state, the new state has to take all events of the old one and append a new event. This new event e' represents the event generated by the model if the transition is performed. Finally, the central part 2.2 of the abstraction algorithm determines whether the given quantised system allows a transition from model state z to z'. The existence condition within relation (24) is checked as follows: An initial quantitative state $\boldsymbol{x}_0$ has to be found. The quantitative state trajectory starting from this initial state is quantised and results in an event sequence. This event sequence has to be equal to the sequence $(e_1, \ldots, e_h, e')$ which represents the considered model state transition. As the first event generated by the quantised system has to be e_1, the search for a suitable $\boldsymbol{x}_0$ has only to be performed within the set of states $\delta Q_x(e_1)$. The time horizon T_h has to be sufficiently long so that $d+1$ events are generated. Note that only a short sequence of $d+1$ events has to be considered. The quantitative behaviour needs not to be traced for more events. Nevertheless, the automaton is able to produce event sequences of any given length by recursive evalution of equation (18). If the event sequence $(e_1, \ldots, e_h, e')$ can be generated the model transition relation is set to $L = 1$ for the concerned case. Otherwise it keeps the value zero.

A search for the initial state $\boldsymbol{x}_0$, hence for the event sequence, is needed to check the existence condition of (24). This search can be realized numerically. For that reason a quantitative description (2) of the continuous-variable system must be available. Automating software has been build for this task. It performs multiple simulations and evaluates the resulting first $d+1$ events of each simulated trajectory. Thus, the model transition relation can be set up by a systematic exploration of the quantitative signal space.

Example

The model construction principle is illustrated by the example in Figure 2 which shows a set of trajectories from a stable oscillator in the phase portrait. Starting from the event e_{87} which denotes the transition from qualitative state 7 to 8 (marked by the arrow), different event sequences are possible:

$$\begin{array}{rcl} \mathcal{S}(e_{87}) & = & \{\, (e_{87}, e_{58}) \,,\, (e_{87}, e_{58}, e_{65}, e_{56}) \,, \\ & & (e_{87}, e_{98}, e_{69}, e_{56}) \,\} \end{array}$$

Three representative trajectories are highlighted.

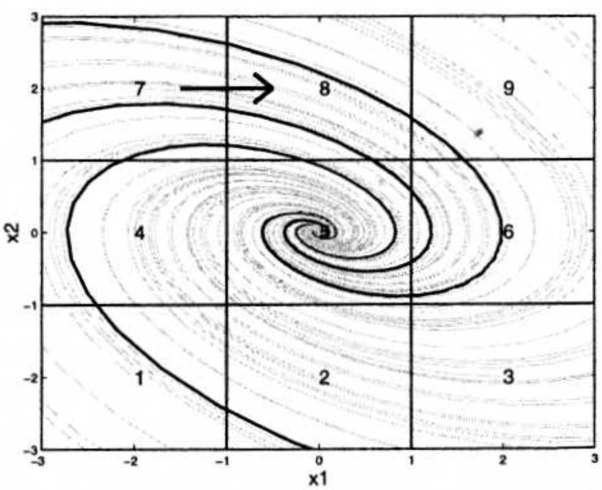

Fig. 2: Trajectories of an oscillator within a quantised state space

To determine the transition relation L for model depth $d = 2$ only two preceding events have to be investigated whether a succeeding event may occur or not. For example, the event e_{56} can take place if either (e_{98}, e_{69}) or (e_{58}, e_{65}) occurred before. More cases that statisfy the condition for a state transition are for example

$$\begin{aligned} 1 &= L((e_{87}, e_{98}), (e_{87})) = L((e_{87}, e_{58}), (e_{87})) \\ &= L((e_{98}, e_{69}), (e_{87}, e_{98})) = L((e_{69}, e_{56}), (e_{98}, e_{69})) \\ &= L((e_{69}, e_{36}), (e_{98}, e_{69})) = \dots \end{aligned}$$

Starting from the initial event $e_0 = e_{87}$ the model generates all event sequences of $\mathcal{S}(e_{87})$ but also spurious solutions, eg $\boldsymbol{E}_{spur.} = (e_{87}, e_{98}, e_{69}, e_{36}, e_{23}, e_{12}, e_{41}, e_{54})$.

Validity of the abstraction algorithm

The automaton's transition relation L is determined by considering short sequences of $d + 1$ succeeding events. However, the resulting automaton can generate sequences of any length by recursively applying equation (18). Given an initial event e_0, the model generates a set of event sequences $\mathcal{M}(e_0, f)$, whereas the qualitative system behaviour covers the set of event sequences $\mathcal{S}(e_0, f)$. The modelling aim (20) demands that $\mathcal{M}(e_0, f) \supseteq \mathcal{S}(e_0, f)$ holds.

Theorem 1: The automaton constructed by algorithm (21...24) satisfies the modelling aim (20).

Proof. It has to be shown that the relation

$$\forall e \in \mathcal{E} : \; p_{\mathcal{S}}(e, k, f) = 1 \quad \Rightarrow \quad p_{\mathcal{M}}(e, k, f) = 1 \quad (25)$$

is valid. If this relation is proved for $0 \dots k$, the model aim holds for horizons $H \leq k$. The proof for (25) will be given by induction.
For $k = 0$ the relation (25) holds as it is assumed that the initial event e_0 is known and, hence, for the model state $z_0 = (e_0)$ $p_{\mathcal{S}}(e_0, 0, f) = 1$ and $p_{\mathcal{M}}(z_0, 0, f) = 1$ imply relation (25).
Assume now that relation (25) is valid for $0 \dots k$ and that for a given event e the equation $p_{\mathcal{S}}(e, k+1, f) = 1$ holds. Due to (9) an event sequence $\boldsymbol{E}(0 \dots k+1) \in \mathcal{S}(e_0, f)$ exists with $e_{k+1} = e$. This fact is equal to the existence of a shorter event sequence $\boldsymbol{E}(0 \dots k)$ and a transition possiblity for e_{k+1}:

$$\begin{aligned} & p_{\mathcal{S}}(e, k+1, f) = 1 \\ \Leftrightarrow \quad & \exists \boldsymbol{E}(0 \dots k+1) \in \mathcal{S}(e_0, f) \; : \; e_{k+1} = e \\ \Leftrightarrow \quad & \exists \boldsymbol{E}(0 \dots k) \in \mathcal{S}(e_0, f) \; \wedge \qquad (26) \\ & \quad p_{\mathcal{S}}(e_{k+1} = e \mid \boldsymbol{E}(0 \dots k)) = 1 \qquad (27) \end{aligned}$$

The first part (26) allows to perform the induction step from $k+1$ to the already proved case k, while the second part (27) involves the specific properties of the abstraction rule as follows:
Equation (26) implies that a shortened event sequence is an element of the model behaviour because the model aim is satisfied for $0 \dots k$. Then, a model state sequence must exist that represents the concerned event sequence. The last model state of this sequence represents the last d occurred events:

$$\begin{aligned} & \boldsymbol{E}(0 \dots k) \in \mathcal{S}(e_0, f) \\ \Rightarrow \quad & \boldsymbol{E}(0 \dots k) \in \mathcal{M}(e_0, f) \\ \Rightarrow \quad & \exists (z_0, z_1, \dots, z_k) \in \mathcal{M}_z(e_0, f) : e_i = \text{lastevent}(z_i) \\ \Rightarrow \quad & p_{\mathcal{M}}(z_k = \boldsymbol{E}(\bar{o}, \dots, k)) = 1 \qquad (28) \end{aligned}$$

with the abbreviation $\bar{o} = \max(0, k - d + 1)$.
Second, the model transition relation can be concluded from (27) as this conditional expression involves that the condition in (24) holds:

$$\begin{aligned} & p_{\mathcal{S}}(e_{k+1} = e \mid (\boldsymbol{E}(0 \dots k)) = 1 \\ \Rightarrow \quad & p_{\mathcal{S}}(e_{k+1} = e \mid \boldsymbol{E}(\bar{o} \dots k)) = 1 \qquad (29) \\ \Rightarrow \quad & L(z', z, f) = 1 \quad \text{with } z = \boldsymbol{E}(\bar{o}, \dots, k), \qquad (30) \\ & z' = \boldsymbol{E}(\hat{o} \dots k+1), \ \text{lastevent}(z') = e \end{aligned}$$

with $\hat{o} = \max(0, k - d + 2)$. Because $p_{\mathcal{M}}(z, k, f) = 1$ and $L(z', z, f) = 1$ are valid for the considered z and z'

$$p_{\mathcal{M}}(z', k+1, f) = 1, \quad \text{lastevent}(z') = e \qquad (31)$$

hold due to equation (18). This result has to be mapped onto the model transition relation by relation (19):

$$\begin{aligned} & p_{\mathcal{M}}(z', k+1, f) = 1 \\ \Rightarrow \quad & p_{\mathcal{M}}(e = \text{lastevent}(z'), k+1, f) = 1. \end{aligned} \qquad (32)$$

Thus, (25) is proved for $k+1$ and by induction for any k. Consequently, the model aim (20) is satisfied for the given abstraction which proves the Theorem 1.

Consistency-based diagnosis

The diagnostic system has to answer the question

> Can the quantised system generate an observed event sequence $\boldsymbol{E}(0 \dots H)$?

It has to determine if the relation $\boldsymbol{E}(0 \dots H) \in \mathcal{S}(e_0, f)$ holds or due to (20) if $\boldsymbol{E}(0 \dots H) \in \mathcal{M}(e_0, f)$. The diagnostic result for horizon H is denoted by $p_N(f, H)$:

$$p_N(f, H) = \begin{cases} 1 & \text{if } \boldsymbol{E}(0 \dots H) \in \mathcal{M}(e_0, f) \\ 0 & \text{else.} \end{cases} \qquad (33)$$

To determine p_N, it is tested whether changes in the signal state are consistent with the state transition relation L of the automaton. As this test can be performed for one observed event after the other, the algorithm can be used recursively and on-line:

Diagnostic Algorithm:

Given: A sequence of events $\boldsymbol{E}(0 \dots H)$
An automaton for each fault mode $f \in \mathcal{F}$.

1. The diagnosis starts with no information about the occurrence of a fault, that is, with

$$p_N(f, 0) = 1 \quad \text{for all } f \in \mathcal{F}. \tag{34}$$

2. After each occurrence of a new event the diagnostic algorithm determines $p_N(f,k)$ recursively for given $p_N(f, k-1)$ as follows:

$$p_N(f,k) = \begin{cases} 1 & \text{if} \quad L(z_k,\, z_{k-1},\, f) = 1, \\ & \qquad p_N(f, k-1) = 1 \\ 0 & \text{else.} \end{cases} \tag{35}$$

Result: $p_N(f, H)$ for each fault $f \in \mathcal{F}$.

In the recursion step k, the algorithm uses the newly observed event e_k, the former events $e_{k-d}, \dots, e_{k-1}$ represented by z_{k-1} and the result $p_N(f, k-1)$ of the former step. The model state z_k is constructed by $(e_{k-d+1}, \dots, e_k)$.
If the fault f could be excluded for a shorter time horizon, $p_N(f, k-1) = 0$ holds and $p_N(f,k) = 0$ follows. Otherwise, the result concerning f depends on the possibility that the automaton performs the state change $z_{k-1} \rightarrow z_k$. This possibility is described by the state transition relation L.

Example: Modelling and diagnosis of a power stage

The presented qualitative modelling approach was applied to an industrial application example and tested by using the model for consistency-based diagnosis. The investigated electronic power stage is an hybrid system with discrete components (parts of the CPU and a logical control unit) and analogue components (resistors, inductances, capacities, diodes etc). It supplies an electromagnetic valve of a car engine (Figure 3).

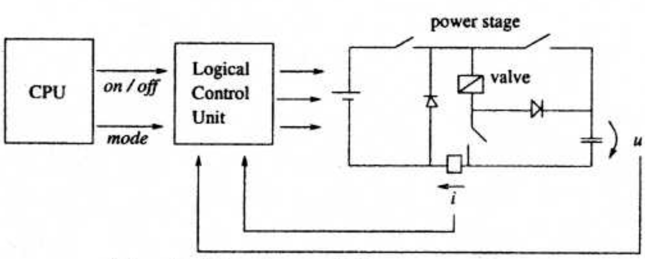

Fig. 3: Structure of the power stage

The power stage shows a typical behaviour which is depicted in Figure 5 for two digital signals and the analogue current signal (upper plots, fat lines). This behaviour is periodically repeated with a cycle time of about 10 to 100 milliseconds.

The spatial quantisation is given by a set of boundaries that define intervals for each signal. These intervals are selected in such a way that the qualitative behaviour captures the main characteristics of the nominal system. The grey boxes in Figure 5 illustrate the evolution of the qualitative signal values.
An event is generated each time the qualitative value of a signal changes. The event times are marked by vertical bars in the bottom plots of Figure 5. As can be seen, the qualitative behaviour differs in normal and faulty cases. This holds true also if the time-scale is abstracted and only pure ("logical") event sequences are observed by the quantiser (6).

The automaton model (10) has been constructed for the normal case and six fault modes (shortcircuits, open circuits etc) by abstracting a given quantitative model of the power stage according to the presented algorithm. An autonomous view could be adopted for this task by including the CPU to the concerned system. This is equal to a specification of the originally arbitrary input signals. The values of the two digital signals are naturally partitioned into the set $\{0, 1\}$ while the analogue values of the current signal are divided into five subsets. A fourth signal, an internal voltage which is not shown in the signal plots, is partitioned into six intervals. The model depth is set to $d = 1$.
Figure 4 illustrates the automaton-graph for the normal case. The cyclic nature of the process is visible as well as the nondeterministic behaviour (from model state 17 transitions are possible to model state 7 or 9).

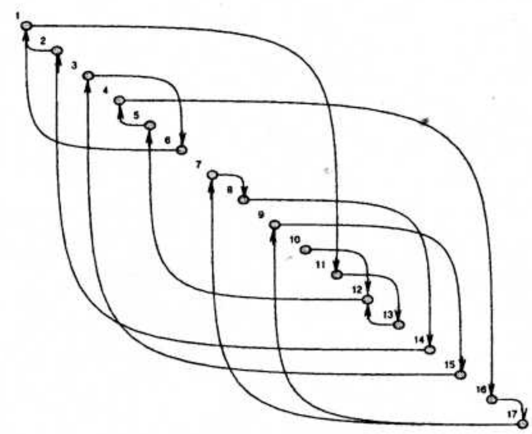

Fig. 4: Automaton for normal mode

The cartesian product of qualitative values generally results in a large number of possible model states. In our example the set of model states could be

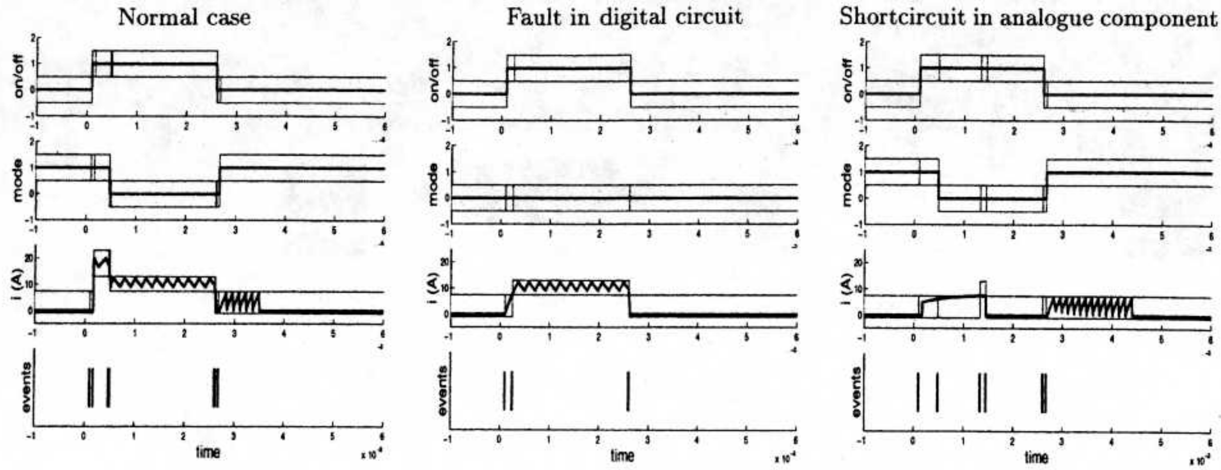

Fig. 5: Quantitative and qualitative behaviour for different fault modes

reduced significantly as the behaviour is limited to few trajectories that do not pass every possible combination of qualitative values. Without this reduction the automaton graph would consist of about 700 nodes.

The consistency–based diagnostic system obtains event sequences which are abstracted from measurements of the power stage. The diagnostic process then checks each event whether it is consistent with any model that was assumed to be valid before the event occurred. Figure 6 visualises two diagnostic runs. It shows the evolution in time of the given signal values and the diagnostic results. The grey boxes denote for each model (numbered from 0 for the normal case and from 1 to 6 for the faults) that the event sequence has been consistent with the model so far.
In the first scenario a normal behaviour was given. After the first event the faults f_3, f_4 and f_5 had to be rejected, after the occurrence of more events only the normal case f_0 could be assumed to be valid. The second scenario concerned a shortcircuit in the analogue circuit. Some faults could be excluded after the first events. Then the event sequence also became inconsistent with the normal model, that is a fault was detected. As only the model for fault f_2 could justify the observed event sequence, the fault could also be discriminated and located at the analogue circuit of the power stage.

Conclusions

The paper presents a discrete-event representation of continuous–variable systems with signals that can only be observed through quantisers. A nondeterministic automaton is applied as discrete–event model.
The main contribution of this paper concerns the determination of the discrete–event model by abstraction of a given quantitative description of the system. This abstraction of a precise system description aims to simplify the diagnostic system.
The paper states that a model that is set up according to the presented abstraction algorithm (21...24) yields a description of the quantised system which satisfies the model requirement (20). The model is then suitable to be used within a model–based diagnosis framework. The approach is demonstrated for the qualitative modelling and consistency–based diagnosis of a power stage. Faults can be detected and isolated by reasoning about the observed event sequences and the information provided by the automaton.

Future work will extend the approach to systems whose behaviour is not precisely known. Uncertainties have to be considered within the abstraction algorithm as parameter tolerances of the system. Alternatively, the discrete–event model can be set up by an identification procedure based on experimental data. Then, an explicit system description is not needed.
Furthermore, the modelling and diagnosis can be improved by including information about the time intervals between two succeeding events in the model, cf the semi-Markov approach of (Lunze 1999).
For compositional modelling a net of automata can be set up. Each automaton of the net represents a single component. The links between the automata are defined by the connections of the concerned system components.

References

Antsaklis, P.; Stiver, J.A.; and Lemmon, M. 1996. A logical DES approach to the design of hybrid control systems. *Mathl. Comput. Modelling* **23**, No. 11/12: 55-76.

Dousson, C.; Gaborit, P.; and Ghallab, M. 1993. Situation Recognition: Representation and Algorithms. *13th International Joint Conference on Artificial Intelligence (IJCAI'93)*, Chambéry, France, pp.166–172.

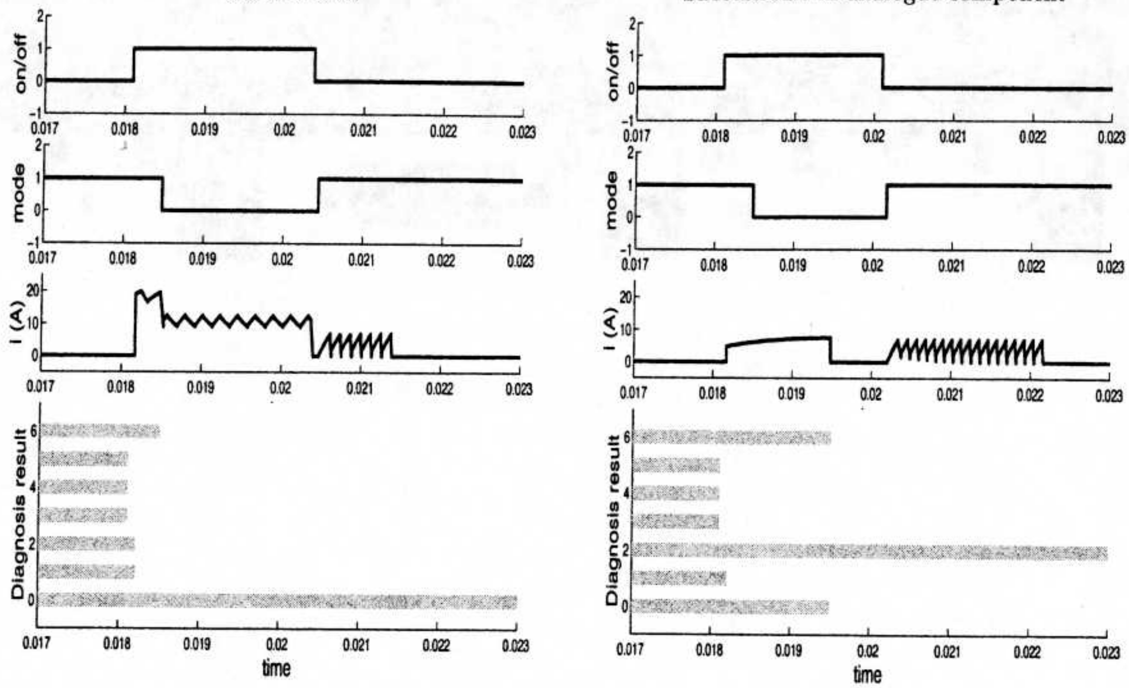

Fig. 6: Two runs of the diagnostic system

Frank, P.M. 1996. Analytical and Qualitative Model–based Fault Diagnosis – A Survey and Some New Results. *European Journal of Control* **2**: 6–28.

Hamscher, W.; Console, L.; and de Kleer, J. eds. 1992. *Readings in Model-Based Diagnosis.* Morgan Kaufmann.

Isermann, R. 1984. Process fault detection based on modeling and estimation methods - a survey. *automatica*, **20**, No. 4: 387–404.

Lichtenberg, G. 1998. Theorie und Anwendung der qualitativen Modellierung zeitdiskreter dynamischer Systeme durch nichtdeterministische Automaten. *Fortschritt-Bericht Reihe 8, 686*, VDI-Verlag.

Lunze, J. 1994. Qualitative modelling of linear dynamical systems with quantized state measurements. *automatica* **30**: 417–431.

Lunze, J. 1995. *Künstliche Intelligenz für Ingenieure*, Band 2. Oldenbourg–Verlag, München.

Lunze, J. 1999. Discrete–event modelling and diagnosis of quantised dynamical systems. *International Workshop on Principles of Diagnosis (DX-99)*, Loch Awe, Scotland.

Lunze, J.; Nixdorf, B.; and Schröder, J. 1999. On the nondeterminism of discrete–event representations of continuous–variable systems. *automatica* **35**, No. 3.

Patton R.; Frank, P.M.; and Clark, R. eds. 1989. *Fault Diagnosis in Dynamic Systems.* Prentice Hall.

Raisch, J.; Klein, E.; O'Young, S.D.; Meder, Ch.; and Itigin, A. 1998. Approximating automata and discrete control for continuous systems – two examples from process control. in *Hybrid Systems V*, Springer-Verlag.

Sampath, M.; Sengupta, R.; Lafortune, S.; Sinnamohideen, K.; and Teneketzis, D. 1996. Failure diagnosis using discrete-event models. *IEEE Trans. Control Systems Technology* **4**, No. 2: 105–124.

Weld, D.S.; de Kleer, J. eds. 1989. *Readings in qualitative reasoning about physical systems.* Morgan Kaufmann.