ADCHEM 2006

International Symposium on Advanced Control of Chemical Processes
Gramado, Brazil – April 2-5, 2006

FINITE AUTOMATA FROM FIRST-PRINCIPLE MODELS: COMPUTATION OF MIN AND MAX TRANSITION TIMES

Heinz A. Preisig *

* *Dept of Chemical Engineering, NTNU, 7491 Trondheim, Norway*

Abstract: Supervisory control schemes of (complex) plants utilize different forms of automata or related structures such as Petri-nets. Empirical, knowledge-based mapping of the plant's operation into such a structure cannot be complete or correct. These automata can be computed by a model-based approach, which guarantees completeness and correctness within the limits of the given model. The result is a non-deterministic automaton (Philips 2001), which however contains no information about the range of transition time that may be expected. This information would be extremely useful for the design of the derived operational procedures such as supervisory controllers on all levels and fault detection and fault isolation schemes. The problem has been formulated several times in the past, for example (Kowalewsky 1999, Engell 1997). Here a solution to the problem is described, which applies to plants generating a monotone flow field for constant inputs.

Keywords: Discrete-event dynamic systems, timed automaton, fault detection, supervisory control, modelling, hybrid systems

1. CURRENT STATE OF AFFAIRS

The increasing complexity of plants and the request for closer interaction between plants asks for more and increasingly sophisticated automation. Traditionally, process units were controlled separately, but increased interaction and required co-ordination make it necessary that the process is viewed and analysed in its full entity, giving rise to the subject of plant-wide control. On the supervisory level, which also links to the management levels such as planning and sequencing of operations and capacity allocation, the plant is event-driven. Currently used empirical modelling techniques cannot guarantee the completeness or correctness of the description, thus one branch of research focused on the computation of one-step automaton representations for continuous plants that are observed by an event detection mechanism. These problems can now be seen as solved. Algorithms exist for linear plants Preisig 1993, (monotone: Preisig 1996, general: Philips et al 1997, Pijpers 1996) and nonlinear plants (Preisig et al 1997, Bruinsma 1997), which can also handle all important exceptions. Also the state explosion problem, which was seen as one of the major drawbacks of these automaton computations, has been completely removed (Philips 2001, Foerstner 2001).

The computation of the automata models is based on the representation depicted in Figure 1, the first box representing the continuous (or fast sampled time-discrete) plant, the second the event detection mechanism, which assumes knowledge of the state and noise-free data. We term this mechanism *domain observer*[1], thereby indicating that the extended event detection mechanism reconstructs the continuous state from the output, if it is not directly accessible, and generates a

[1] –in deviation to Lunze, who uses the term quantizer. By choosing the term *domain observer*, we want to place emphasis on the required knowledge of the state, as it is the state that is discretised and not the output.

signal as the continuous state changes from one subdomain into another defined through boundaries placed into the state space of the continuous system. The resulting non-deterministic automaton models have been used in a first study of DEDS control *synthesis* methods (Philips 1998b, Philips 1999) and fault detection (Philips 1998a, Ramkumar 1998, Ramkumar 1999b, Ramkumar 1999a, Lunze 2000, Lunze 1999).

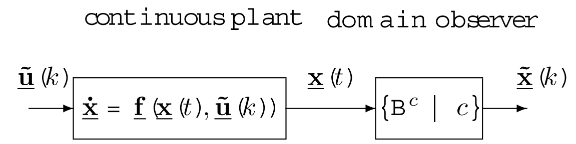

Fig. 1. *Discrete modelling of a discretely observed plant. The tilde quantities represent discrete-event signals*

In both applications it is apparent that knowledge of minimum and maximum transition times would be a very useful piece of information. Thus the problem is formulated, if such information can be obtained from the equations. Here we shall focus on linear plants, though it should be noted that linearity is not limiting, rather limitations on the flow field are imposed, as we shall see below.

2. PROBLEM FORMULATION

Given a linear system with a continuous state $\underline{\mathbf{x}}$, and an input $\tilde{\underline{\mathbf{u}}}$ that, whilst continuous, is changing only at event times and stays constant in between. The derivation may start from a model that is as general as a linear-in-state, time-varying model of the form:

$$\frac{d\underline{\mathbf{x}}(t)}{dt} = \underline{\underline{\mathbf{M}}}(t)\,\underline{\mathbf{x}}(t) + \underline{\mathbf{h}}(t;\tilde{\underline{\mathbf{u}}}), \qquad (1)$$

with $\underline{\mathbf{x}} \in \mathbb{R}^n$, $\tilde{\underline{\mathbf{u}}} \in \mathbb{R}^m$, which for simplicity of algebra we shall reduce to the standard linear, time-constant plant:

$$\frac{d\underline{\mathbf{x}}(t)}{dt} = \underline{\underline{\mathbf{A}}}\,\underline{\mathbf{x}}(t) + \underline{\underline{\mathbf{B}}}\,\tilde{\underline{\mathbf{u}}}(k). \qquad (2)$$

We shall also assume direct knowledge of the state. If the state is not directly accessible, an observer must be added to the plant with the dynamics being fast enough so as to be negligible on the time scale the discrete-event dynamic system operates.

For the automaton representation, we split the continuous state domain into a set of hypercubes by defining a set of ordered boundary values $\beta^c_{d_c}$ with c identifying the state co-ordinate, d_c and the membership of the value in the ordered set of boundary values, $\beta^c_1 < \beta^c_2 < \cdots < \beta^c_{n_c}$ and $[\beta^c_1, \beta^c_{n_c}]$ the validity range of c, defined on the co-ordinate c. For the arbitrary co-ordinate c the boundary set is then:

$$\mathrm{B}^c := \{\beta^c_{d_c} \mid d_c := 1, \ldots, n^c\},$$

with. In practice, these sets are part of the definition of the domain observer. The domain observer assigns membership of the state to an interval dynamically, that is, the boundary point belongs to the interval from where the trajectory enters the boundary (Philips 2001). The hypercubes are conveniently defined in the form of a matrix

$$\underline{\underline{\mathbf{H}}} := \left[[\beta^c_s, \beta^c_{s+1}]\right] := \left[\underline{\mathbf{b}}^c_{-1}, \underline{\mathbf{b}}^c_{+1}\right],$$

with the $\underline{\mathbf{b}}$ vectors being introduced for the elegance of notation later (Equation (3)). Each hypercube has $n!$ faces, each of which is a hyperplane. An event E^S is defined as a crossing of the boundary between two hypercubes, thus a crossing of the actual continuous trajectory through a face S of a hypercube. At this time, the domain observer will emit a signal indicating this event. This definition of an event excludes simultaneous crossing of boundaries; thus, passing through corner points of the hypercubes, defined by the intervals, is not possible. The latter is justified assuming a sequential output line from the domain observer. The computation of the discrete behaviour of the plant as shown in Figure 1 has been reported elsewhere (Preisig 1993, Philips et al 1997, Preisig 1996). Here we wish to compute the minimum and maximum time it takes for the system to move from one transition to the next.

3. WHAT'S THE NEXT POSSIBLE TRANSITION

Having defined the task of computing the minimal and maximal time it takes for event E^B to occur after event E^A, we need first to find what event E^B is possible after E^A has occurred. For this purpose a number of objects are required. Having defined the hypercube representing a discrete state in the continuous state space, and having defined an event as a crossing of the surface of the hypercube, we define a trajectory as

$$\mathrm{X}(\underline{\mathbf{x}}_i) := \{\underline{\mathbf{x}}(t) \mid \forall t, \underline{\mathbf{x}}(t_i) = \underline{\mathbf{x}}_i\},$$

and a bundle of trajectories being

$$\mathrm{T}^A := \{\mathrm{X}(\underline{\mathbf{x}}_i) \mid \mathrm{X}(\underline{\mathbf{x}}_i) \cap A \neq 0\},$$

whereby A is a bounded piece of a hyperplane. With these definitions we can define the surface elements of the hypercube connected by a bundle of trajectories, and thus the *connected events*, by identifying the connecting bundle:

$$\mathrm{T}^{A\ B} := \mathrm{T}^{A} \quad \mathrm{T}^{B};$$

yielding the respective surface pieces:

$$^{A|B} := \mathrm{T}^{A\ B} \quad A,$$
$$^{B|A} := \mathrm{T}^{A\ B} \quad B.$$

The task is thus to find the connecting trajectory bundle. For this purpose, we split the surface of the hypercube into two sets, namely one set where the flow enters F^{in} and a set where the flow exits F^{out}.

At this point, the main assumption is introduced, namely that the flow field is monotone within the extent of the hypercube. At first, this assumption appears rather restrictive. However, one must keep in mind that the flow field is here for a process for which all the inputs are being kept constant. Most natural processes show under these conditions a monotone behaviour. We also exclude the trivial case in which the flow is parallel with a hypercube's surface.

With these conditions, the direction of the flow is:

$$\underline{\mathbf{s}} := \mathrm{sign}(\dot{\underline{\mathbf{x}}}(t)), t < \quad , \qquad (3)$$

and the centre point of the entry surface and the exit surface of the hypercube can be determined:

$$\underline{\mathbf{x}}^{in} := \left[b_i^j\right]_{\forall j}, i := -s_j,$$
$$\underline{\mathbf{x}}^{out} := \left[b_i^j\right]_{\forall j}, i := s_j.$$

These points are the intersection of a set of hyperplanes:

$$\mathrm{P}^{in} := \left\{\mathrm{P}\ (x_i^{in}), \quad i\right\}.$$
$$\mathrm{P}^{out} := \left\{\mathrm{P}\ (x_i^{out}), \quad i\right\}.$$

with the individual hyperplanes:

$$\mathrm{P}\ (x_j) := \{\underline{\mathbf{x}}\ |x_j := b_i^j, i \quad \{-s_j\}\}.$$

Now the different connected pieces of the surfaces can be computed:

$$\mathrm{R}^{A,B} := \mathrm{T}^{A} \quad \mathrm{P}\ (x_j^{out}),$$

and the exit surface piece

$$^{B|A} := \mathrm{R}^{A,B} \quad B. \qquad (4)$$

where $A \quad \mathrm{F}^{in}$ and $B \quad \mathrm{F}^{out}$. If the forward intersection $^{B|A}$ exists, thus the intersection is non-empty, the corresponding next event does exist and the opposite piece of surface on the entry face is the intersection of the trajectory bundle defined by the exit piece $^{A|B}$ [2]:

$$^{A|B} := \mathrm{T}^{\Omega^{B|A}} \quad A.$$

4. TRANSITION TIME

For either of the two models (1, 2) and knowing what next transitions may occur, the transition times can be calculated for any entry point by solving[3] the transcendental equation for

$$x_k^b := \underline{\mathbf{e}}_k^T\ \underline{\mathbf{x}}^b\ (T),$$
$$:= \underline{\mathbf{e}}_k^T\ (\underline{\underline{\mathbf{e}}}^{\int_0^T \underline{\underline{\mathbf{M}}}(t)\,dt}\ (\underline{\mathbf{x}}^a\ (0) + ,$$
$$+ \int_0^T \underline{\underline{\mathbf{e}}}^{-\int_0^t \underline{\underline{\mathbf{M}}}(\tau)\,d\tau}\ \underline{\mathbf{h}}\ (t;\tilde{\underline{\mathbf{u}}})\,dt)),$$

$$x_k^b := \underline{\mathbf{e}}_k^T \left(\underline{\underline{\mathbf{e}}}^{\underline{\underline{\mathbf{A}}}\,T}\left(\underline{\mathbf{x}}^a\ (0) + \int_0^T \underline{\underline{\mathbf{e}}}^{-\underline{\underline{\mathbf{A}}}\,t}\ \underline{\underline{\mathbf{B}}}\ \tilde{\underline{\mathbf{u}}}\,dt\right)\right),$$
$$:= \underline{\mathbf{e}}_k^T \left(\underline{\underline{\mathbf{e}}}^{\underline{\underline{\mathbf{A}}}\,T}\underline{\mathbf{x}}^a\ (0) + \underline{\underline{\mathbf{A}}}^{-1}\left(\underline{\underline{\mathbf{e}}}^{\underline{\underline{\mathbf{A}}}\,T} - \underline{\underline{\mathbf{I}}}\right)\underline{\underline{\mathbf{B}}}\ \tilde{\underline{\mathbf{u}}}\right),$$
$$:= \underline{\mathbf{e}}_k^T\ \left(\underline{\mathbf{g}}\ (T, \underline{\mathbf{x}}^a)\right),$$

where $\underline{\mathbf{x}}^a\ (T) \quad ^a$ and $\underline{\mathbf{x}}^b\ (T) \quad ^b$ and $\underline{\mathbf{e}}_k^T$ the unity vector $[0, ..., x_k, 0, ..., 0], x_k := 1$ selecting the co-ordinate that defines the exit face.

5. THE 3-D SAMPLE SYSTEM

The sample system, being linear and time constant, $:= \{\underline{\underline{\mathbf{A}}}, \underline{\underline{\mathbf{B}}}\}$ being used as an illustration in the continuation is given by the matrices

$$\underline{\underline{\mathbf{A}}} := \begin{pmatrix} 0.8642 & -0.6340 & -0.0672 \\ 15.4736 & -5.3626 & -0.6678 \\ 10.2891 & -2.4301 & -1.5016 \end{pmatrix}, \qquad (5)$$

$$\underline{\underline{\mathbf{B}}} := \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}, \qquad (6)$$

with the input being kept constant at a given value. With the eigenvalues $\underline{} := -[1, -2, -3]$ the system is asymptotically stable.

The Figures 2, 3, 4, 5, 6, 7 show the different pairs of surface elements for the sample system with a zero input. The left-lower front corner being the centre of the entering surface and the right-upper back corner being the centre of the exit surface of the cube.

5.1 *An Alternative View*

An interesting insight is obtained by looking at the problem from a slightly different angle: One

[2] We use here a more detailed notation by indicating the sequence with which the elements of the respective faces are obtained. One may read $B|A$ as (face element B given face element A)

[3] For a reference of solving linear, time-variant ODE's see for example Walter 1960, 1993

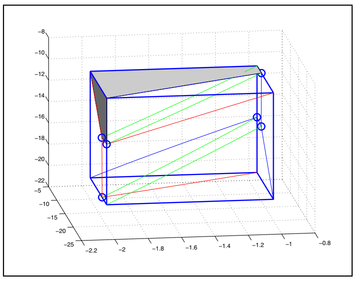

Fig. 2. Front (dark) to attached top (light).

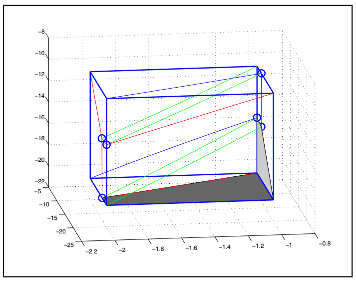

Fig. 3. Bottom (dark) to opposite back (light).

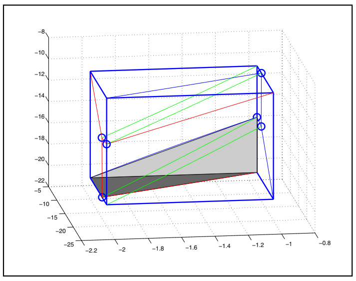

Fig. 4. Bottom (dark) to attached back (light).

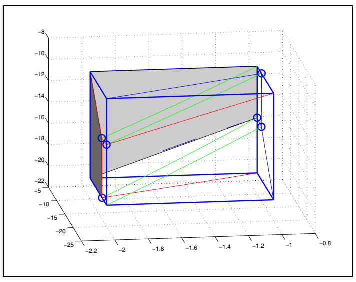

Fig. 5. Front (dark) to attached back (light).

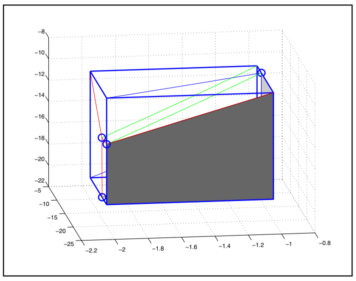

Fig. 6. Front side (dark) to attached back (light).

can view the sectioning of the exit (entry) faces as a projection of the entry (exit) edges onto the opposite side with the dynamic system being the mechanism of projection. Figure 8 shows the projection of the entry edges on the exit surface, which is done forward in time, and Figure 9 the

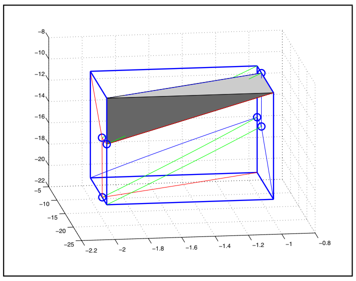

Fig. 7. Front side (dark) to attached top (light).

projection of the exit edges on the entry surface, done backward in time. In the Figure 8 the entry edge is shown in thick lines and the projections in medium lines. In the Figure 9, it is the exit edges in thick lines and the backward projections in medium lines.

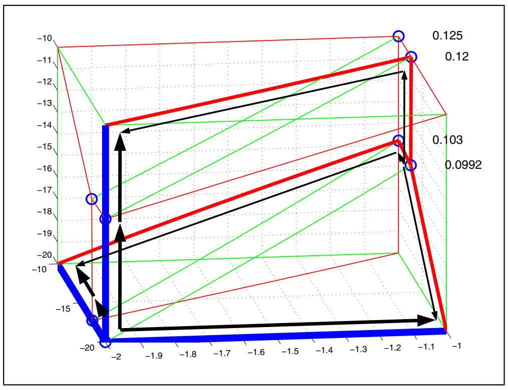

Fig. 8. The view of projecting the entry edges onto the flow-opposite faces.

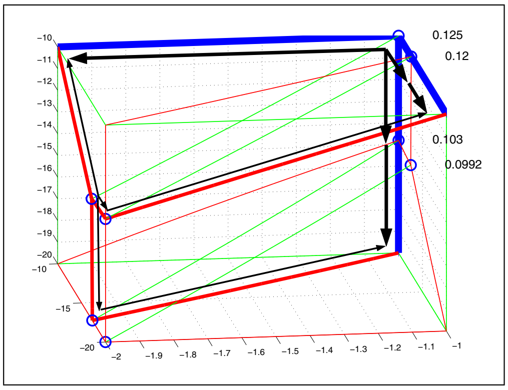

Fig. 9. The backwards projection of the exit edges onto the flow-opposite faces. The arrows indicate the progress of the direction of the begin points as related to the locus of the projected points. The numbers to the left of the marked points indicate the respective transition times.

6. FINDING THE LONGEST AND THE SHORTEST TRAJECTORY IN A MONOTONE FIELD

In a monotone flow field, the computation of the longest and the shortest time is an optimisation problem where the starting point, being element of the entry hypercube surface, is changed such that one finds the minimum and the maximum transition time: In more colloquial terms to find the longest and the shortest trajectory starting on the entry surface of the hypercube.

The optimisation is rather simple if the objective function, namely the transition time changes monotonicly with the adjustable variables, here the position on the entry surface, because in a monotone field the two extremes are associated with opposite corner points of the boundary Gill 1980. It is sufficient to prove monotonic properties of the transition time as a function of the starting point, which is identical of analysing the gradient of the transition time changing with the co-ordinate on the boundary is not changing sign. Let

$$f\left(T, \underline{\mathbf{x}}^a\right) := \underline{\mathbf{s}}\left(\underline{\mathbf{x}}^b(T) - \left(\underline{\mathbf{e}}^{\underline{\underline{\mathbf{A}}}\,T}\underline{\mathbf{x}}^a + \right.\right.$$
$$\left.\left. + \underline{\underline{\mathbf{A}}}^{-1}\left(\underline{\mathbf{e}}^{\underline{\underline{\mathbf{A}}}\,T} - \underline{\underline{\mathbf{I}}}\right)\underline{\underline{\mathbf{B}}}\,\underline{\tilde{\mathbf{u}}}\right)\right),$$

then, since the transition time T cannot be computed analytically, the implicit function theorem is to be used to compute the desired gradient:

$$\frac{dT}{d\underline{\mathbf{x}}^a} := -\frac{f_{\underline{\mathbf{x}}^a}\left(T, \underline{\mathbf{x}}^a\right)}{f_T\left(T, \underline{\mathbf{x}}^a\right)}$$
$$:= \frac{-\,\underline{\mathbf{s}}\,\underline{\mathbf{e}}^{\underline{\underline{\mathbf{A}}}\,T}}{\underline{\mathbf{s}}\left(\underline{\underline{\mathbf{A}}}\,\underline{\mathbf{e}}^{\underline{\underline{\mathbf{A}}}\,T}\underline{\mathbf{x}}^a + \underline{\mathbf{e}}^{\underline{\underline{\mathbf{A}}}\,T}\,\underline{\underline{\mathbf{B}}}\,\underline{\tilde{\mathbf{u}}}\right)}.$$

Monotonic behaviour breaks down as the above gradient passes through a zero in one of its components. At a first glance, the change of sign could be caused by either of the numerator or the denominator. A brief analysis though reveals that it is the denominator that determines the location of the change.

Proof : Consider the boundary Ω^b to initially be close to the starting boundary Ω^a. The transition time can thus be brought arbitrarily close to zero. As the target boundary is moved away, the starting boundary can be moved as well. Again, the difference can be kept arbitrarily small. As long as the gradient does not change, direction, the derivative remains in the same half plain. The sum, or the integral does thus also change in the same direction, which proves the fact that the transition time changes monotonic with the initial location on the starting surface, until the denominator changes sign. The latter is the locus of a derivative in one co-ordinate being zero, which is on a flat plane cutting the space into two monotonic sub-domains. These local equilibrium plains intersect, if we constrain the discussion to asymptotically stable (non-oscillatory) systems, at the global equilibrium point.

□

Alternatively one can prove that the function $T\left(\underline{\mathbf{x}}_a\right)$ is monotone as long as the the right-hand-side of the dynamic model equations does not change sign:

Proof : Given that $\underline{\underline{\mathbf{A}}}\,\underline{\mathbf{x}} + \underline{\underline{\mathbf{B}}}\,\underline{\tilde{\mathbf{u}}}$ does not change sign (asymptotic behaviour), the inverse does not change sign either and the integral with time is monotone and so is the integral of the inverse. The monotone behaviour changes as the sign of the integrand changes.

□

With the accumulated information, it is trivial now to provide the minimal and maximal transition times for each transition. In the cases where the entry face is attached to the exit face, the minimal transition is always zero. The maximal transition is given by the longest trajectory forming the tube running across the hypercube, which is attached to the respective piece of the entry face. Thus only four different maximal transition times occur in the whole, independent of the dimension of the problem. The transition times for the example are shown in Figure 8.

7. CONCLUSIONS

The surface of the hypercube splits into two sections, the entry section and the exit section. If the flow is *not running in parallel with the co-ordinates*, there is only one central entry corner and only one central exit corner. Each of the faces of the hypercube belongs to one of the two surfaces, namely the entry or the exit section. Each face is split into sections whereby each of the entry sections is connected with an exit section, thus defining the reachable pieces of the surface as a function of the entry location.

The computation of the different surface sections is done by finding the forward projection of the centre entry corner onto the exit surface and the backward image of the centre exit point onto the entry surface. The edges of the entry faces project onto the exit surfaces using the dynamics of the process for the projection. The result is the lines subdividing the exit faces. The inverse computation, namely the backward projection of the centre exit point and the exit edges onto the entry surface results the other set of face-sectioning lines.

The minimal and the maximal times for a transition are associated with the centre corner points and the additional two trajectories cutting across the hypercube. Because the objective function, namely the transition time is a monotone function of the location on the entry surface, the maximum and the minimum are associated with transitions from the corner and edge points or to the corner and edge points. Only four trajectories must be computed.

The principle of the computation is not limited to linear systems. Monotonicity is the only condition being used. Note that monotonicity is only requested for the region of the continuous state space being covered by the discrete state space at constant inputs.

8. REFERENCES

Bruinsma U B D M R; (1997); State-Event Discrete Modelling of Nonlinear Plants; *M.Sc Thesis, NR 1984*; Eindhoven University of Technology, Eindhoven, The Netherlands.

Engell S, Kowalewski S, Krogh B; (1997); Discrete events and hybrid systems in process control; *Proc. 5th Intern Conference on Chem Process Control, Tahoe City, (J. E. Kantor, C. E. Garcia, B. Carnahan, Eds.), AIChE Symp Series, No. 316, Vol. 93*; 165-176.

Förstner D; (2001); Aualitative Modellierung für die Prozessdiagnose und deren Anwendung auf Dieseleinspritzsysteme; *PhD Thesis*; Universität Hamburg-Harburg, Hamburg and Bosch AG, Stuttgart; 138.

Gill P E; (1980); Practial Optimisation; *Stronge*; USA.

Kowlewski S, Engell S, Preussig J, Stursberg O; (1999); Verification of logic controllers for continuous plants using timed condition/event-system models; *Automatica*; **35**; 505-518.

Lunze J, Schröder J; (1999); Process diagnosis base on a discrete-event description; *Automatisierungstechnik*; **47**, 780; 358-365.

Lunze J, Schröder J; (2000); State observation and diagnosis of discrete-event systems described by stochastic automata; *Discrete Event Dynamic Systems*; in press.

Philips P P H H; (2001); Modelling, Control and Fault Detection of Discretely-Observed Systems; *PhD Thesis*; TU-Eindhoven, Eindhoven, The Netherlands, ISBN 90-386-1729-1; 157p.

Philips P P H H, Bruinsma U B D M R, Weiss M, Preisig H A; (1997); A Mathematical Approach to Discrete-event Dynamic Modelling of Hybrid Systems; *IFAC Symposium on AI in Real-Time Control*; Kuala Lumpur, Malaysia; 185-190.

Philips P P H H, Ramkumar K, Lim K W, Preisig H A, Weiss M; (1998a); Automation-based fault detection and isolation; *ESCAPE-99, Comp & Chem Eng*; Budapest, Hungary; S215-S218.

Philips P P H H, Weiss M, Preisig H A; (1998b); A design strategie for discrete control of continuous systems; *American Control Conference 98*; San Diego, USA.

Philips P P H H, Weiss M, Preisig H A; (1999); Control based on discrete-event models of continuous systems; *European Control Conference 99*; Karlsruhe, Germany.

Pijpers J H; (1996); Modelling for supervisory control; *M.Sc. Thesis, NR-1956*; Eindhoven University of Technology, Eindhoven, The Netherlands.

Preisig H A; (1993); More on the Synthesis of a Supervisory Controller From First Principles; *12th IFAC World Congress 93*; Sydney, Australia; Vol. V, 275.

Preisig H A; (1996); A Mathematical Approach to Discrete-Event Dynamic Modelling of Hybrid Systems; *Computers & Chemical Engineering*; **20**, Suppl; S1301-S1306.

Preisig H A, Pijpers M J H, Weiss M; (1997); A discrete modelling procedure for continuous processes based on state-discretization; *MATHMOD 2*; Vienna, Austria; 189-194.

Ramkumar K B; (1999a); Fault Detection and Diagnosis in Process Plants Using Finite-State Automaton; *M.Sc. Thesis*; National University of Singapore; 103 p.

Ramkumar K B, Philips P P H H, Ho W K, Preisig H A, Lim K W; (1999b); A real-time realisation of fault-detection and diagnosis using finite-state automation; *Int Fed of Auto Control '99, 14th World Congress*; Bejing, China, Proceedings; P-211.

Ramkumar K, Philips P P H H, Preisig H A, Ho W K, Lim K W; (1998); Structured fault-detection and diagnosis usig finite-state automation; *24th Annual conference of the IEEE Industrial Electronis Society*; Aachen, Germany.

Walter W; (1960,1993); Gewöhnliche Differentialgleichungen, eine Einführung; *Springer*; Germany; 325 S.