In *Proceedings of SAFEPROCESS'03*,
Washington, DC, June, 2003

ONLINE MONITORING OF HYBRID SYSTEMS USING IMPRECISE MODELS

Bernhard Rinner and Ulrich Weiss

Institute for Technical Informatics,
Graz University of Technology, AUSTRIA

Abstract: This paper presents a model-based monitoring system which is based on imprecise models where the structure is known and the parameters may be imprecisely specified by numerical intervals. This monitoring approach is applied to hybrid systems and is now able (i) to follow a known sequence of imprecisely modeled modes, (ii) to detect unknown transitions and (iii) to refine the time uncertainty of the transitions as well as the imprecision of mode models. The implemented system is demonstrated by online monitoring of a non-trivial heating system.

Keywords: model-based monitoring; hybrid systems; imprecise models; parameter estimation

1. INTRODUCTION

The primary objective of a monitoring system is to detect abnormal behaviors of a supervised system as soon as possible to avoid possible shutdown or damage. A particularly important and widely-applied approach is *model-based monitoring* which relies on a comparison of the predicted behavior of a model with the observed behavior of the supervised system. The monitoring system MOSES (Rinner and Weiss, 2002*a*; Rinner and Weiss, 2002*b*) is based on imprecise models in order to express and reason with incomplete knowledge about the supervised system. To keep the overall uncertainty during monitoring small MOSES repeatedly partitions the uncertainty space of the imprecise models and checks the derived model's state for consistency with the measurements. Inconsistent partitions are then refuted resulting in a smaller uncertainty space and a faster failure detection.

This paper extends the online monitoring system MOSES to *hybrid systems* (Branicky, 1995). In a hybrid system, isolated regions of rapid change are abstracted to instantaneous discontinuities separating regions of continuous behaviors. The continuous segment of the system's behavior is referred to as *mode of operation* and a discontinuous change is referred to as *transitions.* The basic idea of the extension to hybrid systems is to map the time uncertainty about the transition into an uncertainty of the new mode's initial state. MOSES is now able (i) to follow a known sequence of imprecisely modeled modes, (ii) to detect unknown transitions and (iii) to refine the time uncertainty of the transitions as well as the imprecision of mode models.

The remainder of this paper is organized as follows. Section 2 briefly introduces the monitoring approach. Section 3 presents the extensions of MOSES to monitoring hybrid systems. Section 4 demonstrates the performance of MOSES for hybrid systems by monitoring a complex heating system. A discussion and a summary of related work conclude this paper.

⋆ This work has been partially supported by the Austrian Science Fund under grant number P14233-INF.

2. MONITORING CONTINUOUS MODES

2.1 Overview

In MOSES a continuous mode is represented by a differential equation model. Imprecision is introduced by specifying model parameters as numerical intervals which span the uncertainty space of the model. Only bounds on the trajectory, i.e., *envelopes*, can be derived from this imprecise model and a possibly imprecisely specified initial state. The key step in our approach is to partition the uncertainty space of the model into several subspaces. The trajectories derived from each subspace are then checked for consistency with the measurements. Each inconsistent subspace is refuted and excluded from further investigations. Partitioning and consistency checking are continued resulting in a smaller uncertainty space of the model. When all subspaces are refuted, a discrepancy between model prediction and observation has been recognized and a fault has been detected.

2.2 Imprecise Modeling and Subspace Partitioning

In more detail, a technical system is modeled as a linear differential equation of order n

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}(\mathbf{p})\mathbf{x}(t) + \mathbf{B}(\mathbf{p})\mathbf{u}(t)\\ \mathbf{y}(t) &= \mathbf{C}(\mathbf{p})\mathbf{x}(t) + \mathbf{D}(\mathbf{p})\mathbf{u}(t)\end{aligned} \tag{1}$$

where $\mathbf{x}(t)$ is the state vector at time t, $\mathbf{u}(t)$ is the input vector at time t, $\mathbf{p}(t)$ is the parameter vector at time t, $\mathbf{y}(t)$ is the output vector at time t, and $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ are matrixes with appropriate dimensions and functions of $\mathbf{p}$. In an exact model, $\mathbf{p}(t)$ is a vector of real numbers. However, in a model with uncertain parameters, $\mathbf{p}(t)$ is replaced by a vector of intervals $\tilde{\mathbf{p}}(t) = [(\underline{p}_1, \overline{p}_1), (\underline{p}_2, \overline{p}_2), \cdots, (\underline{p}_K, \overline{p}_K)]^T$, where K is the number of uncertain parameters. [1] A model with uncertain parameters, i.e., an *imprecise model*, can therefore be described as:

$$\begin{aligned}\dot{\tilde{\mathbf{x}}}(t) &= \mathbf{A}(\tilde{\mathbf{p}})\tilde{\mathbf{x}}(t) + \mathbf{B}(\tilde{\mathbf{p}})\mathbf{u}(t)\\ \tilde{\mathbf{y}}(t) &= \mathbf{C}(\tilde{\mathbf{p}})\tilde{\mathbf{x}}(t) + \mathbf{D}(\tilde{\mathbf{p}})\mathbf{u}(t)\end{aligned} \tag{2}$$

Equation 2 is the starting point of our approach. It defines an imprecise model of the supervised system with K uncertain parameters. Thus, this model has a K-dimensional uncertainty space. A *partition* is defined as

$$\tilde{\mathbf{q}}(t) = [(\underline{q}_1, \overline{q}_1), (\underline{q}_2, \overline{q}_2), \cdots, (\underline{q}_K, \overline{q}_K)]^T \tag{3}$$

with $\tilde{\mathbf{q}} \subseteq \tilde{\mathbf{p}}$. Thus, a partition divides the uncertainty space into smaller regions. A complete partitioning into M partitions must satisfy the following condition: $\bigcup_m \tilde{\mathbf{q}}^{(m)} = \tilde{\mathbf{p}}$ where $m = 1, \ldots, M$. A model based on a partition of the uncertainty space is referred to as *subspace model*. The state of a subspace model m is formally defined as:

$$\begin{aligned}\dot{\tilde{\mathbf{x}}}^{(m)}(t) &= \mathbf{A}(\tilde{\mathbf{p}}^{(m)})\tilde{\mathbf{x}}^{(m)}(t) + \mathbf{B}(\tilde{\mathbf{p}}^{(m)})\mathbf{u}(t)\\ \tilde{\mathbf{y}}^{(m)}(t) &= \mathbf{C}(\tilde{\mathbf{p}}^{(m)})\tilde{\mathbf{x}}^{(m)}(t) + \mathbf{D}(\tilde{\mathbf{p}}^{(m)})\mathbf{u}(t)\end{aligned} \tag{4}$$

To apply imprecise models in MOSES, we must compute their trajectories. A simple but intractable method to derive the trajectories is to repeat a numerical integration starting from any point within the uncertainty space. [2] However, if we assume monotonicity it is sufficient to focus only on few points of the uncertainty space.

2.3 Consistency Checking

With the monotonicity assumption of $\mathbf{x}(t)$ and $\mathbf{y}(t)$ with regard to the parameters $\mathbf{p}(t)$ over the range of the intervals, the (uncertain) state of a subspace model can be represented by the (exact) state at extremal points, i.e., *corner points*, of a subspace. The corner points of a subspace are defined as all combinations of upper and lower bounds of a partition $\tilde{\mathbf{q}}(\mathbf{t})$. Note that an uncertainty space of dimension K results in 2^K corner points.

In order to test the consistency between prediction and observation, MOSES checks whether the measurements at time t lie within the trajectory space derived from the corner points (Rinner and Weiss, 2002*a*). Since the trajectory space is derived from exact corner states, standard numerical methods for computing the solution of differential equations can be used. In MOSES the measurements may also be bound by numerical intervals in order to account for measurement noise. This requires an extension of the consistency check which is described in (Rinner and Weiss, 2002*a*).

2.4 Dynamic Partitioning

A large number of subspace models may be detected as inconsistent during monitoring. To increase the fault detection performance of MOSES the uncertainty space of consistent subspace models can be partitioned dynamically according to Equation 3. This dynamic partitioning results in smaller subspace models that potentially describe the supervised system more precisely. There is

[1] In our approach we assume that the parameters do not vary over time.

[2] Note that for deriving the trajectories of imprecise models it is also sufficient to focus on points belonging to the external surface of the uncertainty space (Bonarini and Bontempi, 1994). However, the number of trajectories to be computed is still infinite, even if it is of a lower order.

clearly a trade-off between the number of (active) subspace models and the computational load in MOSES. Dynamic partitioning allows to adjust on-line the computational load as well as the degree of uncertainty of individual subspace.

3. MONITORING MULTIPLE MODES

In order to extend MOSES to hybrid systems the monitoring process must be able to follow a sequence of modes. MOSES must, therefore, identify a mode change, perform the transition by instantaneous changing the input and/or the model, and continue monitoring the new mode. It is important to know when and what kind of transitions occurs. Since we assume to know the (nominal) sequence of modes the supervised system exhibits the remaining problem is to determine the time of the transition. In general, there are three different cases to consider.

(1) The exact time instant of the transition is known.
(2) Only bounds on the time instant of the transition can be specified.
(3) No information about the transition time is known a priori.

Information about the time instant of the transition may be delivered by an auxiliary signal such as an input to an actuator or by identifying abrupt changes in the observations (Basseville and Nikiforov, 1993). Note that MOSES currently requires to detect all mode changes. If a mode change is missed, e.g., due to an autonomous transition, MOSES may deliver a false alarm.

A failure in the supervised system may be manifested by a deviation within the mode or a transition to an unknown mode. In both cases MOSES may eventually detect the failure by identifying a discrepancy in the current or preceeding modes when the degree of model uncertainty and measurement noise is sufficiently small.

3.1 Transitions at known time instants

Transitions with known exact time instants can be directly handled by MOSES. At the transition, MOSES is initiated with the imprecise model of the new mode, and the final state of the previous mode is mapped to the initial state of the new mode. Since the final state is imprecisely specified this mapping will result in an imprecise initial state of the new mode.

3.2 Transitions with time bounds

When the time instant of the transition is not exactly known, MOSES has to manage an additional uncertain parameter – the *time uncertainty* of the transition. Introducing a new uncertain parameter at each transition would result in an ever increasing uncertainty space. In order to overcome this problem we transform the time uncertainty of the transition into an (additional) uncertainty of the initial state of the new mode. This keeps the dimension of the uncertainty space constant.

This transformation is achieved by adding the time uncertainty to the set of uncertain parameters as long as the monitoring process is within the time uncertainty of the transition. More formally, the monitoring process of MOSES is extended by the following steps. Note that the initial time bound is given by the interval $[t_1, t_2]$.

(1) At t_1 the time uncertainty is added as an additional uncertain parameter $p_{K+1} = [t_1, t_2]$.
(2) As long as the monitoring process is not in the new mode monitoring is continued with $K + 1$ uncertain parameters.
(3) When the new mode has been genuinely reached – either by detecting a discrepancy or by exceeding t_2 – the resulting state space $\mathbf{x}(\mathbf{p})$ is mapped to the initial state space of the new mode, and the time uncertainty is removed from the set of uncertain parameters.

3.3 Transitions without time information

If no information about the time of the transition is available in advance, MOSES may eventually detect the inconsistency of the old mode's prediction with the observed data. This time serves then as an initial upper bound of the transition. In order to compute a lower bound we have to backtrack, i.e., we have to compute the state of the new mode starting from previous sampling points and check for consistency with the observed data. This backtracking is continued, i.e., starting from earlier sampling points, until the predicted envelopes are inconsistent with the data. Then a lower bound of the time uncertainty has been identified, and MOSES can continue monitoring by mapping the time uncertainty to an initial state uncertainty of the new mode (cp. Section 3.2).

Fig. 1 illustrates the computation of the bounds on the time uncertainty. The key steps can be summarized as follows:

(1) **initialize**
Set the immediate predecessor sampling time after the discrepancy has been detected as the new starting time t_i of the backtracking.
(2) **determine upper bound**
Start backtracking at t_i and compute the state of the new mode until the time point the discrepancy has originally detected. Check

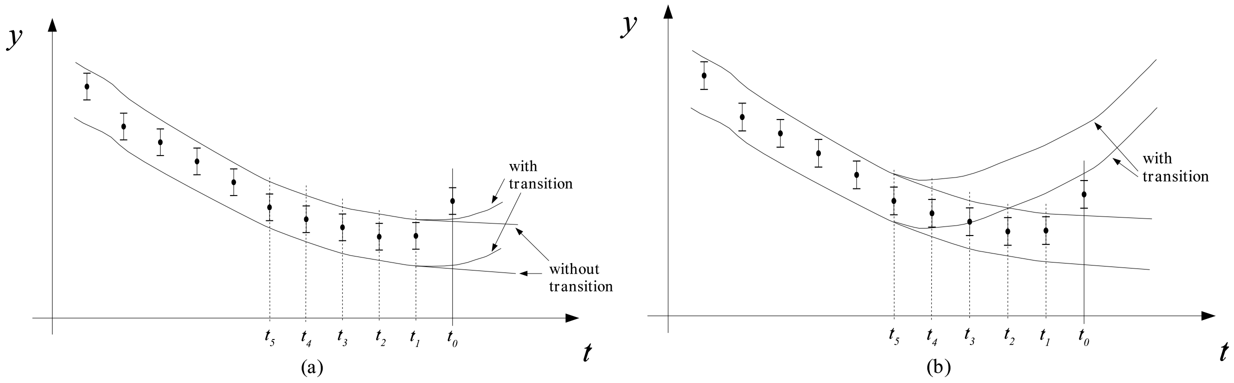

Fig. 1. Computing time bounds of a transition. In both graphs, the measurements including a noise interval as well as the predicted envelops of the previous and new modes are plotted. The transition is detected at t_0 because the measurement lies outside the predicted envelopes of the previous mode. As depicted in graph (a) backtracking is initially started at t_1 (inconsistent). Backtracking terminates at t_5 (also inconsistent) as depicted in graph (b) resulting in a time uncertainty of (t_5, t_1).

the consistency of the derived envelopes with the data samples after t_i. If the data is consistent, t_i is an upper bound of the time uncertainty then goto (3). Otherwise, decrease t_i ($t_i \leftarrow t_{i-1}$) and start backtracking at the new time point.

(3) **determine lower bound**
Same backtracking procedure as in (2). However, if the data is *inconsistent*, t_i is a lower bound of the time uncertainty then terminate. Otherwise, decrease t_i ($t_i \leftarrow t_{i-1}$) and start backtracking at the new time point.

Note that for the computation of the bounds a history of the input values and the state values are needed. In order to restrict the memory requirements (and the computation time) backtracking is only applied for a limited time period.

3.4 Refining the time uncertainty

Time uncertainty on a transition affects the entire correspondence between prediction and observation in the following mode, resulting in propagating uncertainty. Thus, keeping the time uncertainty as small as possible is important for the overall failure recognition performance of MOSES.

When more observations become available in the new mode the initial time uncertainty of the transition may be refined. For this refinement the same algorithm as described in the previous section can be applied. Since there are more data samples available for the consistency check more backtracking steps may result in an inconsistency resulting in a smaller time uncertainty.

3.5 Refuting subspace models

MOSES partitions the uncertainty space and independently computes the trajectories of each subspace model. If a subspace model becomes inconsistent, the subspace is refuted and excluded from further investigation. However, if a transition with time uncertainty is given then simply refuting any inconsistent subspace model is false. This is because a subspace model detected as inconsistent within the time uncertainty of a transition may be consistent in the new mode.

In order to avoid premature refutation of subspace models we have to store all inconsistent subspaces until a discrepancy has been detected and the time uncertainty of a possible transition has been computed. All subspace models that have been detected as inconsistent within the time uncertainty must be included in the new mode. To store a subspace model it is sufficient to save the last state of that subspace model before the inconsistency has been detected.

4. EXPERIMENTAL RESULTS

We demonstrate the performance of MOSES on a “real” technical system which is comprised of three heating components mounted on a thermal conductive plate. A process control computer (B&R 2003) controls the heating elements and transfers the measured samples as well as the control actions issued to MOSES via a RS 232 interface. MOSES has been completely implemented on a standard PC running Linux.

We model the heating system as a linear differential equation (Rinner and Weiss, 2002*a*). The temperatures T_i of the components represent the state vector of this system. Two temperatures T_1 and T_2 are measured. The noise interval of the sensors are given as 0.2. The individual thermal masses and the thermal conductivities across the components are imprecisely specified resulting in a total of 8 uncertain parameters.

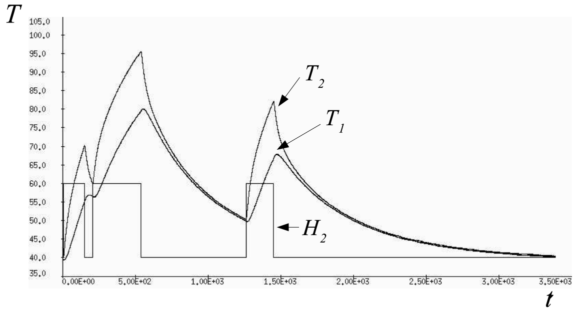

Fig. 2. Scenario #1 with a total of six transitions. The temperatures T_1 and T_2 as well as the control signal H_2 are depicted in the graph. H_2 is not transferred to MOSES.

Transition time (H_2)	*new mode*	*lower bound*	*upper bound*	*ISS*
5.5	heat	5.0	8.4	5.057
149.9	idle	148.1	154.0	6.838
206.3	heat	205.9	209.9	6.413
538.5	idle	536.6	543.3	7.819
1260.5	heat	1260.1	1263.8	5.336
1448.9	idle	1446.9	1453.2	7.278

Table 1. Computed bounds on the time uncertainty and the initial state space (ISS) of the new mode for scenario #1.

In our experiments we distinguish two modes, *heat* and *idle*, whether the central heating element is switched on or off. We use three different scenarios in our experiments. No a priori time information about the transitions is available is made available to MOSES in all scenarios. The sampling period is given as 0.1 sec.

4.1 Scenario #1

Scenario #1 corresponds to a simple sequence of six transitions between heat and idle modes (Fig. 2). MOSES is able to follow all mode changes of this scenario, i.e., it detects the transitions, computes the time uncertainties and maps the time uncertainty to the initial state space (ISS) which is defined as the product of the interval size of all state variables.

Tab. 1 presents the computed bounds on the transition time and the size of the ISS. The real transition time (left column) lies always between the computed bounds. Note that the ISS does not increase with the number of transitions. Fig. 3 depicts the situation at the second transition. MOSES detects a inconsistency 4.1 sec after the transition.

4.2 Scenario #2

Scenario #2 is similar to the first scenario. However, a failure has been introduced by switching on the heater H_1 of an adjacent component for a short period of time during the second idle mode (Fig. 4). MOSES detects a discrepancy within this idle mode after 4.8 sec the failure has been introduced. It switches then to the known but wrong heat mode. However, after 2 more seconds this mode is also detected as inconsistent and the failure has eventually been detected (Fig. 5).

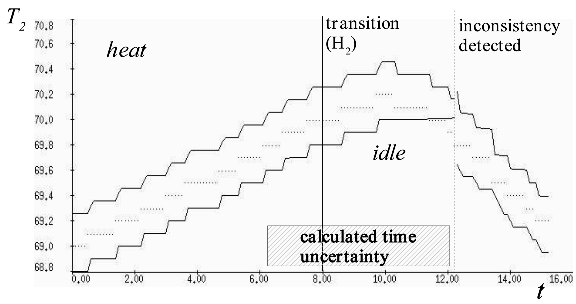

Fig. 3. The transition at $t = 8$ in scenario #1. The envelopes (solid lines) and the measurement (points) are plotted in the graph.

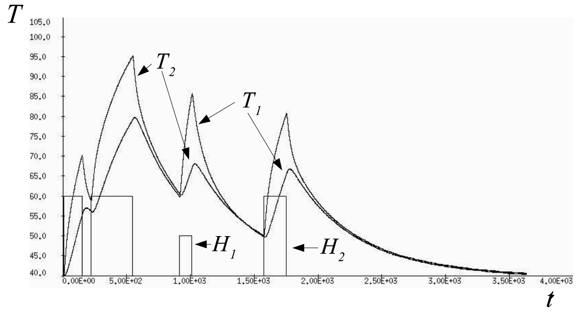

Fig. 4. Scenario #2. A failure is introduced by switching on H_1.

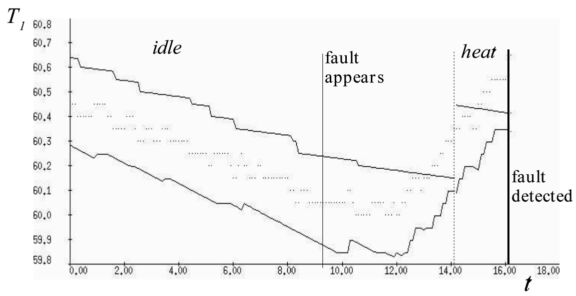

Fig. 5. Recognition of the failure in scenario #2.

4.3 Scenario #3

In this experiment MOSES has been tested whether it is able to follow rapid mode changes. Scenario #3 is generated by a discrete controller on the process control computer. This controller stabilizes the temperature of component 2 to a given set point (Fig. 6).

Tab. 2 shows the result of MOSES following the transitions. The ISS does not increase over time.

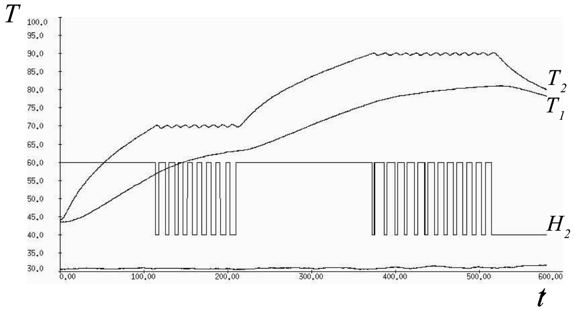

Fig. 6. Scenario #3. H_2 is controlled by the process control computer to stabilize T_2.

Transition time (H_2)	*new mode*	*lower bound*	*upper bound*	*ISS*
1.6	heat	1.2	4.2	3.303
115.4	idle	113.9	119.4	5.66
119.5	heat	119.5	124.4	6.551
127.6	idle	126.1	131.7	4.576
131.8	heat	131.8	137.0	6.364
138.5	idle	137.1	142.8	5.439
143.0	heat	142.9	148.2	6.102
148.6	idle	148.3	153.5	6.346
153.6	heat	153.6	158.8	6.205

Table 2. Time bounds and ISS for scenario #3.

5. DISCUSSION

We have presented an extension of MOSES to monitoring dynamical systems that exhibit both discrete and continuous behaviors. MOSES is now able to monitor hybrid systems by transforming time uncertainties of the transitions into initial state uncertainties of the following modes. This work expands ideas from semi-quantitative system identification (Kay *et al.*, 2000) and semi-quantitative reasoning (Rinner and Kuipers, 1999) to monitoring hybrid systems.

There is a clear tradeoff between model uncertainty including the initial state space and the fault detection performance of MOSES. A smaller uncertainty space results in a faster fault detection and is also important to follow rapid mode changes. Note that the number of uncertain parameters strongly affects the computational load of MOSES. Thus, for online monitoring it is important to keep the number of uncertain parameters small.

Related work includes Biswas et al. (Manders *et al.*, 2000; Narasimhan *et al.*, 2002) who apply numerical and qualitative techniques to monitor hybrid systems. A hybrid observer is used to track a continuous mode and determine whether a mode change has occurred. Model disturbances and measurement noise are accounted for by Kalman filters. Diagnosis of hybrid systems based on probabilistic models are presented in (Hofbaur and Williams, 2002; Benazera *et al.*, 2002) where the mode transitions are represented by extended hidden Markov models. Tornil et al. (Tornil *et al.*, 2000) apply *interval models* to fault detection. The envelopes of these models are derived using interval prediction or interval simulation.

MOSES currently assumes to know the (nominal) sequence of modes in advance. It is natural to include a mode selection functionality during the monitoring process by incorporating ideas from the above mentioned work. Another direction for further work is to expand MOSES to a complete fault diagnosis system.

REFERENCES

Basseville, M. and I. Nikiforov (1993). *Detection of abrupt changes: theory and applications.* Prentice-Hall.

Benazera, Emmanuel, Luise Travé-Massuyès and Philippe Dague (2002). State Tracking of Uncertain Hybrid Concurrent Systems. In: *Proceedings of the 13th Workshop on Principles of Diagnosis (DX'02).* Semmering, Austria. pp. 106–114.

Bonarini, Andrea and Gianluca Bontempi (1994). A Qualitative Simulation Approach for Fuzzy Dynamic Models. *ACM Transactions on Modeling and Computer Simulation* **4**(4), 285–313.

Branicky, Michael (1995). Studies in Hybrid Systems: Modeling, Analysis, and Control. PhD thesis. Massachusetts Institute of Technology.

Hofbaur, Michael and Brian Williams (2002). Hybrid Diagnosis with Unknown Behavioral Modes. In: *Proceedings of the 13th Workshop on Principles of Diagnosis (DX'02).* Semmering, Austria. pp. 97–105.

Kay, Herbert, Bernhard Rinner and Benjamin Kuipers (2000). Semi-Quantitative System Identification. *Artificial Intelligence* **119**(1-2), 103–140.

Manders, Eric-J., Sriram Narasimhan, Gautam Biswas and Pieter J. Mosterman (2000). A Combined Qualitative/Quantitative Approach for Fault Isolation in Continuous Dynamic Systems. In: *Proceedings of SAFEPROCESS'2000.* Budapest, Hungary. pp. 512–517.

Narasimhan, Sriram, Gautam Biswas, Gabor Karsai, Tivadar Szemethy, Tim Bowman, Mark Kay and Kirby Keller (2002). Hybrid Modeling and Diagnosis in the Real World: A Case Study. In: *Proceedings of the 13th Workshop on Principles of Diagnosis (DX'02).* Semmering, Austria. pp. 7–15.

Rinner, Bernhard and Benjamin Kuipers (1999). Monitoring Piecewise Continuous Behaviors by Refining Semi-Quantitative Trackers. In: *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99).* Morgan Kaufmann. Stockholm, Sweden. pp. 1080–1086.

Rinner, Bernhard and Ulrich Weiss (2002*a*). Model-based Monitoring of Piecewise Continuous Behaviors using Dynamic Uncertainty Space Partitioning. In: *Proceedings of the 13th Workshop on Principles of Diagnosis (DX'02).* Semmering, Austria. pp. 146–150.

Rinner, Bernhard and Ulrich Weiss (2002*b*). Online Monitoring by Dynamically Refining Imprecise Models. Technical Report TR 02/01. Graz University of Technology.

Tornil, Sebastian, Teresa Excobet and Vicenc Puig (2000). Fault Detection Using Interval Models. In: *Proceedings of SAFEPROCESS'2000.* Budapest, Hungary. pp. 1180–1185.