In *OeGAI Journal, 21(3):5-12*
October, 2002

The model-based online monitoring system MOSES

Bernhard Rinner and Ulrich Weiss
Institut für Technische Informatik
Technische Universität Graz
Inffeldgasse 16
A-8010 Graz, AUSTRIA
[rinner,uweiss]@iti.tu-graz.ac.at

Abstract

Model-based monitoring relies on a comparison between the predicted behavior of a model and the measured behavior of a supervised system. Especially in technical environments, a monitoring system must be able to reason with incomplete knowledge about the supervised system, to process noisy and erroneous observations and to react within a limited time.

We present MOSES, a model-based monitoring system that combines and extends methods from artificial intelligence and control engineering. MOSES relies on models with known structure and parameters that may be imprecisely specified by numerical intervals. It uses numerical techniques to derive the trajectories of these imprecise models and reduces the model's uncertainty by refutation. MOSES checks the consistency between measurements and model's prediction by exploiting qualitative information from residuals.

MOSES expresses and reasons with incomplete knowledge, processes incrementally measurements to enable online operation, and reduces the model's uncertainty during the monitoring procedure. It has been implemented on a Linux PC and evaluated by online monitoring of a complex heating system.

keywords: model-based monitoring; imprecise models; uncertainty space partitioning; parameter estimation

1 Introduction

The primary objective of monitoring is to detect abnormal behaviors of a supervised system as soon as possible to avoid shutdown or damage. Due to the increased complexity of many supervised systems monitoring is becoming more and more important. Especially technical systems such as robots, space crafts or automobiles provide a vast number of challenges for a monitoring system [Rin02]. In such environments, the monitoring system must be able to reason with incomplete knowledge about the supervised system, to process noisy and erroneous observations and to react within predefined time windows.

A particularly important and widely-applied approach is *model-based monitoring* [HCdK90] [CP99] which relies on a comparison between the predicted behavior of a model and the observed behavior of a supervised system. A discrepancy between the predicted and the observed behavior indicates a fault in the supervised system. Model-based monitoring techniques have been investigated and developed within the artificial intelligence *(DX Diagnosis)* and the control engineering *(FDI Fault Detection and Isolation)* communities over the last years.

Model-based monitoring makes use of mathematical models of the supervised system. However, a perfectly accurate and complete model of a technical system is never available. There is always a mismatch between the technical system and its mathematical model even if there are no faults present. For model-based monitoring it is, therefore, important how to express and reason with incomplete knowledge.

We present an alternate approach to model-based monitoring of technical systems (called MOSES, for MOnitoring using uncertainty Space partitioning for tEchnical Systems) that combines and extends techniques from DX and FDI. Our approach [RW02b] [RW02a] is based on imprecise models where the structure of the models is known and the parameters may be imprecisely given as numerical intervals. These parameter intervals span the uncertainty space of the model. From an imprecise model based on intervals only bounds on the trajectory, i.e., *envelopes*, can be derived. When new measurements become available MOSES checks the consistency of this new information with the model's prediction and refutes inconsistent parts from the uncertainty space of the model. This keeps the predicted envelopes small and improves the fault recognition capabilities.

Our approach bridges and extends methodologies from the FDI and the DX communities in the following way:

(1) Modeling the incomplete knowledge about the supervised system is based on differential equations which are augmented by numerical intervals of parameters. In general, reasoning with intervals is complex and tedious. However, by focusing only on individual points on the surface of the model's uncertainty space we can use standard numerical methods, i.e., Runge-Kutta integration, for deriving the envelopes.

(2) Measurements and prediction are checked for consistency by exploiting qualitative information from the residuals. Discrepancies are only reported when there is no overlap between the measurements and the predicted envelopes.

(3) By refuting parts of the uncertainty space that are provably inconsistent, MOSES also performs model refinement during monitoring. This technique originates from semi-quantitative system identification [KRK00] [RK99] and is extended to online monitoring.

By exploiting measurements as soon as possible for online monitoring, by refining the uncertainty space conservatively through refutation, and by applying standard numerical techniques for deriving the trajectories of imprecise models, our approach combines and complements techniques from FDI and DX and helps make them more widely and robustly available.

The remainder of this paper is organized as follows. Section 2 presents the basic monitoring approach of MOSES. Section 3 sketches the implementation of MOSES and introduces some experimental results. Section 4 discusses related work and Section 5 concludes this paper with a discussion and an outlook for future work.

2 Monitoring by Refuting Subspace Models

2.1 Overview

Monitoring methods based on imprecise models can reason with incomplete knowledge in the model as well as with noisy measurements. A main drawback of this approach, however, is that the envelopes may diverge very rapidly which delays or even inhibits a fault recognition. We have revised this interval approach to model-based monitoring with the primary goal to keep the resulting envelopes as small as possible.

In our approach, we exploit the measurements from the supervised system as soon as possible to refine the uncertainty in the model and the derived envelopes. 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 the model prediction and the observation has been recognized and a fault has been detected.

2.2 Imprecise Modeling and Subspace Partitioning

In general, a technical system can be modeled as differential equation of order n

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}(t)) \\ \mathbf{y}(t) &= \mathbf{g}(\mathbf{x}(t), \mathbf{p}(t))\end{aligned} \tag{1}$$

where $\mathbf{x}(t)$ is the state vector, $\mathbf{u}(t)$ is the input vector, $\mathbf{p}(t)$ is the parameter vector, $\mathbf{y}(t)$ is the output vector, and $\mathbf{g}$ and $\mathbf{f}$ are vector functions. 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 $\tilde{\mathbf{p}}(t) = [(\underline{p}_1(t), \overline{p}_1(t)), (\underline{p}_2(t), \overline{p}_2(t)), \cdots, (\underline{p}_K(t), \overline{p}_K(t))]^T$, where K is the number of uncertain parameters. A model with uncertain parameters, i.e., an *imprecise model*, can therefore be described as:

$$\begin{aligned}\dot{\tilde{\mathbf{x}}}(t) &= \mathbf{f}(\tilde{\mathbf{x}}(t), \mathbf{u}(t), \tilde{\mathbf{p}}(t)) \\ \tilde{\mathbf{y}}(t) &= \mathbf{g}(\tilde{\mathbf{x}}(t), \tilde{\mathbf{p}}(t))\end{aligned} \tag{2}$$

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

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

with $\tilde{\mathbf{q}}(t) \subseteq \tilde{\mathbf{p}}(t)$. Thus, a partition divides the uncertainty space into smaller regions. A model based on a partition of the uncertainty space is referred to as *subspace model*. The system equation of a subspace model m is formally defined as:

$$\begin{aligned}\dot{\tilde{\mathbf{x}}}^{(m)}(t) &= \mathbf{f}(\tilde{\mathbf{x}}^{(m)}(t), \mathbf{u}(t), \tilde{\mathbf{q}}^{(m)}(t)) \\ \tilde{\mathbf{y}}^{(m)}(t) &= \mathbf{g}(\tilde{\mathbf{x}}^{(m)}(t), \tilde{\mathbf{q}}^{(m)}(t)).\end{aligned} \tag{4}$$

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.[1] 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 $\tilde{\mathbf{q}}(\mathbf{t})$ and can be represented as set $Q^{(m)}(t) = \{\tilde{\mathbf{q}}_i^{(m)}(t)\}$ with $i = 1, \ldots, 2^K$. Thus, an uncertainty space of dimension K results in 2^K corner points.

The consistency of a subspace model m can be simply checked by computing the state of each corner point and checking whether the measurement lies within the "state space" of all corner points [RW02a]. Note that this is an incremental procedure because the new "state space" at t can be computed from the state space at $t-1$. Since this technique is based on the calculation of an exact state (at corner points), we can use standard numerical methods for computing the solution of differential equations. Note that subspaces are only refuted when they are genuinely inconsistent with the measurements.

Figure 1 depicts the consistency check which is realized by computing the residuals for each state at the corner points of the subspace m. Then the consistency can be simply determined by comparing the signs of the maximum and minimum values of the residual.

An additional source of uncertainty in technical systems is measurement noise. In MOSES noise can be simply handled as additional uncertain parameters, i.e., the measurements are superimposed

[1] 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 [BB94]. However, the number of trajectories to be computed is still infinite, even if it is of a lower order.

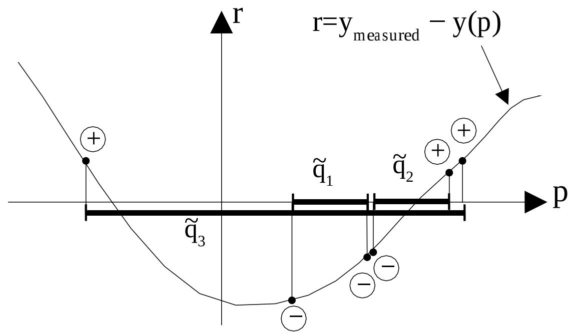

Figure 1: Consistency checking demonstrated with an one-dimensional uncertainty space and three subspaces $\tilde{q}_1$, $\tilde{q}_2$, and $\tilde{q}_3$. The residuals at the corner points of subspace $\tilde{q}_1$ are both negative, therefore, the model with the subspace $\tilde{q}_1$ is inconsistent with the measurement. In subspace $\tilde{q}_2$, the residuals at the corner points have different signs. Thus, $\tilde{q}_2$ is consistent. For the parameter range of subspace $\tilde{q}_3$ the monotonicity assumption is violated. In this case, checking the residuals' signs at the corner points is not feasible.

by a fixed (noise) interval. The residual is then computed using this interval. Note that uncertain parameters representing measurement noise may not be divided during dynamic partitioning.

As depicted in Figure 1 the consistency check does hold only as long as the monotonicity of the residual with regard to the parameters is given. However, simply to assume monotonicity would result in a strong limitation to the modeling of technical systems. MOSES does not rely on monotonic models. It uses a special check to test for monotonicity before applying the consistency check. Only when the check confirms monotonicity the consistency check is applied [RW02a].

3 Implementation and Experimental Results

This section briefly sketches the implementation and some experimental results of MOSES. The basic monitoring approach described in the previous section has already been extended to (hybrid) systems with discrete transitions including a check for the monotonicity of the state values and a generalized consistency check [RW02c]. In this section, however, we briefly sketch the implementation and some experimental results of MOSES.

3.1 MOSES Implementation

The monitoring system MOSES has been completely implemented on a standard PC running Linux. The software – including a graphical user interface (Figure 2 – has been implemented in C/C++. MOSES communicates with the technical system via a standard process control computer. The performance of MOSES is evaluated using both a simulated as well as a "real" supervised system. The evaluation is performed in offline and online operation as well.

3.2 Fault Detection Performance

We demonstrate the performance of our monitoring algorithm on a "real" technical system which is comprised of three heating/cooling components mounted on a thermal conductive plate. A process control computer (B&R 2003) controls the three heating/cooling components. The measured samples as well as the control actions issued are transferred to MOSES via a RS 232 interface.

Our model of the heating system includes the temperature T_i, the heat flow q_i, the thermal conductivity L_i and the thermal mass C_i of the three components; it is a third order differential equation with a total of 8 uncertain parameters.

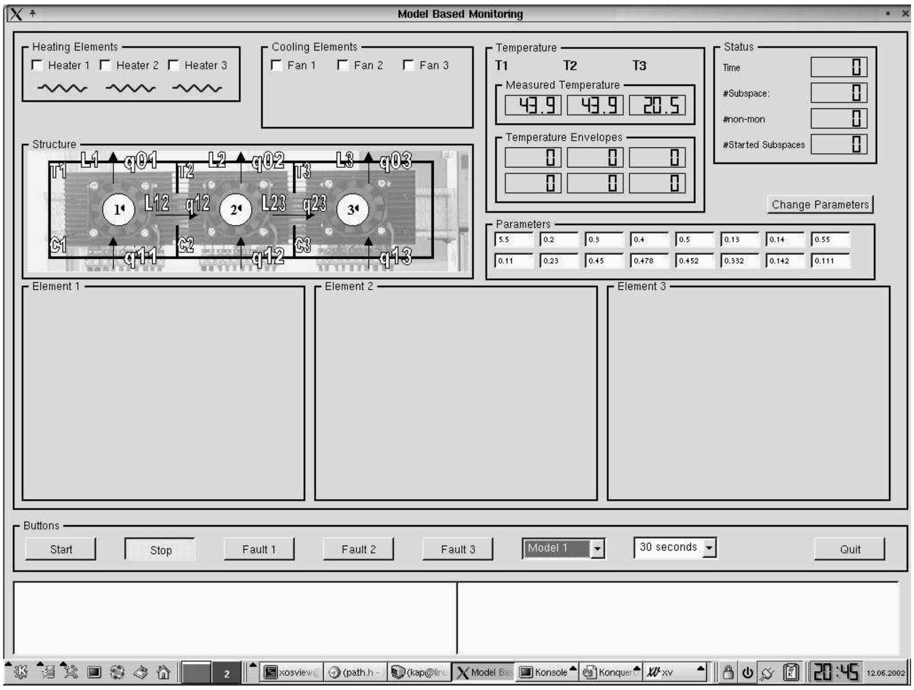

Figure 2: User interface of MOSES.

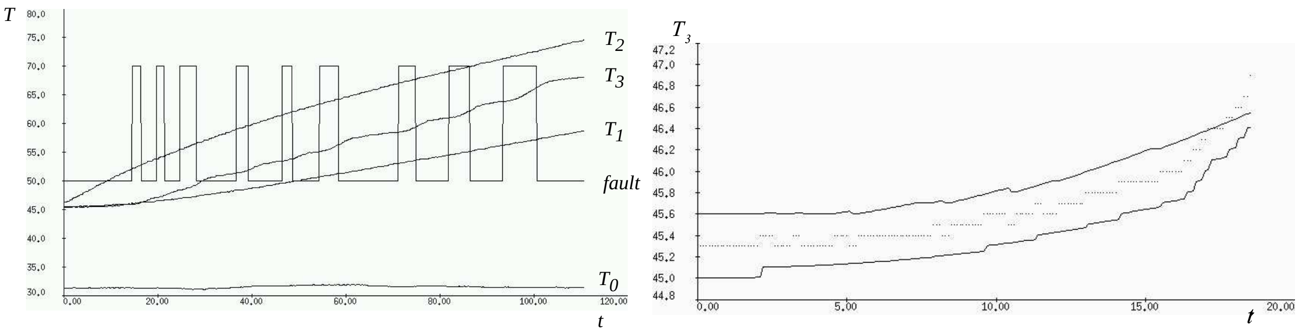

Figure 3: Scenario of an intermittent fault (left) and fault detection by observing temperature sensor T3 (right). The sensor readings of all four temperature sensors as well as the induced fault are plotted in the left graph. The derived trajectories and the sensor reading for temperature sensor T3 are plotted in the right graph.

We demonstrate the fault detection performance using an intermittent fault scenario in component 3 of the heating system (Figure 3 (left)). At $t = 0$ the heating element of component 2 is switched on; all other actuators remain turned off. Starting at $t = 14.6s$, the heating element of component 3 is switched on and off several times. Figure 3 (right) depicts the situation of detecting this fault scenario by observing only the temperature sensor of component 3. At $t = 18.4s$, the sensor value exceeds the trajectory derived from the imprecise model. Note, that the envelopes are kept quite small for all the time.

Table 1 presents the time required by our monitoring system for detecting the intermittent fault in the heating system. This table summarizes the results from experiments where the uncertainty space of the model and the number of observed variables have been varied. Model #1 has the largest uncertainty space and model #4 has the smallest uncertainty space. The parameter intervals of these models are presented in Table 2. Note that observing only T1 is not sufficient in order to detect the fault within the observation period of $110.7s$.

observ. variables	*fault recognition [s]*			
	model #1	model #2	model #3	model #4
T1, T2, T3	3.8	3.8	3.8	3.8
T1, T2	> 96.1	74.1	53.1	47.8
T1	> 96.1	> 96.1	> 96.1	> 96.1

Table 1: Time required to detect the intermittent fault using MOSES.

Space No.	L_1	L_2	L_{12}	C_1	C_2	*size*
#1	$[0.11, 0.13]$	$[0.13, 0.18]$	$[0.62, 0.86]$	$[48, 68]$	$[49, 66]$	162
#2	$[0.11, 0.13]$	$[0.15, 0.18]$	$[0.62, 0.76]$	$[48, 60]$	$[59, 66]$	14
#3	$[0.12, 0.13]$	$[0.15, 0.18]$	$[0.62, 0.76]$	$[51, 57]$	$[60, 65]$	2.5
#4	$[0.12, 0.13]$	$[0.15, 0.18]$	$[0.62, 0.76]$	$[51, 54]$	$[61, 65]$	1

Table 2: Achieved parameter refinement of our heating system. The size of the uncertainty space is specified as the product of the range of the parameter intervals to the size of model #4.

3.3 Refining Imprecise Models

Partitioning results in more but smaller subspace models. If a subspace model is inconsistent with the measurements, the subspace model is refuted and excluded from further investigation. This can result in a smaller uncertainty space after refutation and also in smaller bounds on the uncertain parameters, if the measurement and the system is fault-free. This can also be seen as a form of *parameter estimation*.

Refinement of uncertain parameters has also been applied to estimate the parameters of our heating system. We started the refinement with very large parameter intervals. In this experiment, the initial parameters were given as $L_1 = [0.01, 1]$, $L_2 = [0.01, 1]$, $L_{12} = [0.01, 10]$, $C_1 = [5, 200]$ and $C_2 = [5, 200]$. The inputs are estimated as $q_{i1} = q_{i3} = 1.24W$ and $q_{i2} = 32.8W$.

We have executed a monitoring process on MOSES using this large subspace model and measurements of T_1, T_2 and T_3 from the (healthy) heating system. The achieved refinement on the parameters are summarized in Table 2. The different refinements depend on the maximum number of subspace models allowed during monitoring. Note that this corresponds to a reduction of the uncertainty space of 8 orders of magnitude. The size of the uncertainty space can be defined as the product of the range of all uncertain parameter intervals.

4 Related Work

Although, MOSES represents uncertainty by parameter intervals, it is not using interval models. MOSES simulates the model at the corner points of the uncertainty space, and therefore, with exact parameter values.

Model-based monitoring using uncertainty space partitioning is related to the interval identification algorithm of Schaich et al. [SKKC99]. In their approach the consistency check is only performed at the qualitative level. Thus, valuable detection time is lost, as long as the fault is only manifested in a quantitative value. Additionally, the monotonicity of the states with regard to the parameters is not checked, and, therefore, the derived envelopes may not be complete. Petridis an Kehagias [PK98] have also developed an algorithm with subspace partitioning. Their partitioning is only performed in advance and the consistency check is based on probabilities depending on the measurement noise and Markovian time-varying parameters. Hence, they can not refute subspaces because the probabilities will never reach zero. Not mentioned in the paper is that their algorithm converges to more than one partition.

Bonarini and Bontempi [BB94] have developed a simulation approach for linear systems with uncertain initial states and a monotonicity check. They have also described a technique to simulate models with uncertain parameters by introducing additional states which represent the parameter values. Unfortunately, this leads to non-linear system equations, where the monotonicity check is not always sufficient. Also related to our work is Armengol et al. [ATMVdlR00, AVTMS01]. The simulation is based on modal interval arithmetic, which produces *overbounded* and *underbounded* envelopes of a technical system. To minimize the rate of false and missed alarms, the uncertainty space is only partitioned at *critical* measurements (which lie between the underbounded and overbounded envelopes). In comparison, our approach "leads" to exact envelopes for linear systems, and, therefore, the problem of false and missed alarms in the above mentioned sense does not exist.

Other work in monitoring [HM00, MRS98, Bog95] uses multiple models for fault detection. These models represent known faults of the supervised system.

5 Discussion

In this paper, we have presented a model-based monitoring approach based on refining imprecise models of the supervised system. The fundamental assumption of this approach is the monotonicity of the state values with regard to the range of the parameters. In order to evaluate this assumption we have implemented a special check for monotonicity. The uncertainty space of the imprecise model is partitioned into smaller subspace models. When new measurements become available inconsistent subspace models are refuted resulting in a smaller uncertainty space. When all subspace models have been refuted, a fault has been detected.

This approach can also be seen as *system identification*, because refuting subspace models reduces the uncertainty space, resulting in smaller bounding intervals on the parameters. Smaller intervals on the parameters result in a faster fault recognition time. With dynamic uncertainty partitioning, it is possible to partition parameter intervals online at the monitoring process. So, MOSES can adapt the uncertain parameters to the real system at the beginning of the monitoring process, and continue to detect faults with smaller uncertainty space.

However, this approach is in contrast to traditional system identification where the model space is specified by a parameterized differential equation. Identification selects numerical parameter values so that simulation of the model best matches the measurements. By using refutation instead of search our method is able to derive *guaranteed* bounds on the trajectories.

There are several directions for future research. First, MOSES is currently able to monitor hybrid system models when the transition between modes are known, e.g., by signals indicating a transition. When the time of the transition is not known an additional source of uncertainty is introduced – the time uncertainty between trajectory and measurements of the new mode. Second, MOSES uses currently imprecise linear system models. Non-linear models are more expressive, however, they significantly complicate the determination of the monotonicity of state variables. The computation of the state at corner point is also not sufficient for computing the envelopes. Future research should, therefore, be focused to special classes of non-linear system. Finally, MOSES can be viewed as a method for tracking hypotheses and detecting discrepancies in the context of diagnosis. To develop a complete fault diagnosis system for dynamic systems, MOSES could be combined with existing methods for automated model building and proposing hypotheses given weak information between observations and predictions [dKW87, Ng91].

Acknowledgments

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

References

[ATMVdlR00] Joaquim Armengol, Louise Travé-Massuyès, Josep Vehi, and Josep Lluis de la Rosa. A Survey on Interval Model Simulators and their Properties related to Fault Detection. *Annual Reviews in Control*, 24:31–39, 2000.

[AVTMS01] Joaquim Armengol, Josep Vehi, Louise Travé-Massuyès, and Miguel Angel Sainz. Application of Multiple Sliding Time Windows to Fault Detection Based on Interval Models. In *Proceedings of the 12th International Workshop on Principles of Diagnosis (DX-01)*, pages 9–16, Sansicario, Italy, 2001.

[BB94] Andrea Bonarini and Gianluca Bontempi. A Qualitative Simulation Approach for Fuzzy Dynamical Models. *ACM Transactions on Modeling and Computer Simulation*, 4(4):285–313, 1994.

[Bog95] S. Bogh. Multiple Hypothesis-Testing Approach to FDI for the Industrial Actuator Benchmark. *Control Eng. Practice*, 3(12):1763–1768, 1995.

[CP99] Jie Chen and Ron J. Patton. *Robust Model-Based Fault Diagnosis for Dynamic Systems.* Kluwer, 1999.

[dKW87] Johan de Kleer and Brian C. Williams. Diagnosing Multiple Faults. *Artificial Intelligence*, 32:97–130, 1987.

[HCdK90] Walter Hamscher, Luca Console, and Johan de Kleer, editors. *Readings in Model-based Diagnosis.* Morgan Kaufmann, 1990.

[HM00] Peter D. Hanlon and Peter S. Maybeck. Multiple-Model Adaptive Estimation using a Residual Correlation Kalman Filter Bank. *IEEE Transactions on Aerospace and Electronic Systems*, 36(2):393–406, April 2000.

[KRK00] Herbert Kay, Bernhard Rinner, and Benjamin Kuipers. Semi-Quantitative System Identification. *Artificial Intelligence*, 119(1-2):103–140, May 2000.

[MRS98] Raman Mehra, Constantino Rago, and Sanjeev Seereeram. Autonomous Failure Detection, Identification and Fault-tolerant Estimation with Aerospace Applications. *Proceedings of the IEEE Aerospace Applications Conference*, 2:133–138, 1998.

[Ng91] Hwee Tou Ng. Model-Based, Multiple-Fault Diagnosis of Dynamic, Continuous Physical Devices. *IEEE Expert*, 6(6):38–43, December 1991.

[PK98] V. Petridis and Ath. Kehagias. A Multi-model Algorithm for Parameter Estimation of Time-varying Nonlinear Systems. *Automatica*, 34(4):469–475, 1998.

[Rin02] Bernhard Rinner. Überwachung und Diagnose (special issue). *Telematik*, 8(2), 2002.

[RK99] Bernhard Rinner and Benjamin Kuipers. Monitoring Piecewise Continuous Behaviors by Refining Semi-Quantitative Trackers. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 1080–1086, Stockholm, Sweden, August 1999. Morgan Kaufmann.

[RW02a] Bernhard Rinner and Ulrich Weiss. Model-based Monitoring of Piecewise Continuous Behaviors using Dynamic Uncertainty Space Partitioning. In *Proceedings of the 13th Workshop on Principles of Diagnosis (DX'02)*, pages 146–150, Semmering, Austria, May 2002.

[RW02b] Bernhard Rinner and Ulrich Weiss. Model-based Monitoring using Uncertainty Space Partitioning. In *Proceedings of the 21st IASTED International Conference on Modelling, Identification and Control (MIC 2002)*, pages 174–179, Innsbruck, Austria, February 2002.

[RW02c] Bernhard Rinner and Ulrich Weiss. Online Monitoring by Dynamically Refining Imprecise Models. Technical Report TR 02/01, Institute for Technical Informatics, Graz University of Technology, 2002. submitted to *IEEE Transactions on Systems, Man and Cybernetics.*

[SKKC99] David Schaich, Rudibert King, Uwe Keller, and Mike Chantler. Interval Identification - a Modelling and Design Technique for Dynamic Systems. In *Proceedings of the 13th International Workshop on Qualitative Reasoning*, pages 6–9, June 1999.