Model-based Monitoring of Piecewise Continuous Behaviors using Dynamic Uncertainty Space Partitioning[1]

Bernhard Rinner[2] and **Ulrich Weiss**[2]

Abstract. Monitoring gains importance for many technical systems such as robots, production lines or anti lock brakes. A monitoring system for technical systems must be able to deal with incomplete knowledge of the supervised system, to process noisy observations and to react within predefined time windows. This paper presents a new approach to monitoring technical systems based on imprecise models. Our approach repeatedly partitions the uncertainty space of an imprecise model 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 further focuses on the extension of our basic approach to monitoring systems that exhibit both continuous and discrete behaviors. Our monitoring system has been implemented using COTS components and has been demonstrated in online monitoring of a non-trivial heating system.

Keywords: fault detection; hybrid systems; imprecise models; residual generation

1 INTRODUCTION

The primary objective of a monitoring system is to detect abnormal behaviors of a supervised system as soon as possible to avoid shutdown or damage. Technical systems such as robots, production lines or anti lock brakes provide a vast number of challenges for a monitoring system, i.e., it must be able to deal with incomplete knowledge about the supervised system, to process noisy observations and to react within predefined time windows.

A particularly important and widely-applied approach is *model-based monitoring* [6, 5] which relies on a comparison of the predicted behavior of a model with the observed behavior of the supervised system. Our approach using dynamic uncertainty space partitioning [12] is based on imprecise models where the structure of the models is known and the parameters may be imprecisely given as numeric intervals. These parameter intervals span the uncertainty space of the model. From an imprecise model based on intervals only bounds on the trajectory (envelopes) can be derived. Dynamic uncertainty space partitioning keeps the envelopes small by exploiting the measurements from the supervised system as soon as possible. Whenever new measurements arrive residuals are generated at the "corner points" of the uncertainty space and checked for consistency by comparing their signs. This results in a fast fault detection [12].

The fundamental assumption of dynamic uncertainty space partitioning is that the model's state values are monotonic within the range of the uncertainty space. Discontinuous transitions in the system's model may introduce non-monotonic behaviors in the state values and, therefore, violate our assumption for the consistency check. In order to preserve a conservative monitoring approach for hybrid systems, we have to extend our consistency check by a *monotonicity check*. Whenever the monotonicity of the state values is given the consistency check can be performed potentially resulting in a refutation of the imprecise model. If the monotonicity is not known the consistency check is simply ignored and no model is refuted.

The remainder of this paper is organized as follows. Section 2 describes the technical details of uncertainty space partitioning and the consistency check. Section 3 discusses the necessary extensions of our approach to monitoring systems which exhibit both continuous and discrete behaviors. Section 4 presents experimental results of our monitoring approach in a real-world system with several changes of a input value. A discussion and a summary of related work conclude this paper.

2 MONITORING BASED ON UNCERTAINTY SPACE PARTITIONING

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 subspace are refuted, a discrepancy between model prediction and observation has been recognized and a fault has been detected.

2.2 Subspace Partitioning and Consistency Checking

In general, a technical system can be modeled as

$$\begin{aligned} \mathbf{x}_t &= \mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{p}_{t-1}) \\ \mathbf{y}_t &= \mathbf{g}(\mathbf{x}_t, \mathbf{p}_t) \end{aligned} \quad (1)$$

[1] This work has been supported by the Austrian Science Fund under grant number P14233-INF. The authors are in alphabetical order.
[2] Institute for Technical Informatics, Graz University of Technology, AUSTRIA; email: [rinner,uweiss]@iti.tu-graz.ac.at.

where $\mathbf{x}_t$ is the state vector at discrete 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 $\mathbf{g}$ 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:

$$\begin{aligned} \tilde{\mathbf{x}}_t &= \mathbf{f}(\tilde{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}, \tilde{\mathbf{p}}_{t-1}) \\ \tilde{\mathbf{y}}_t &= \mathbf{g}(\tilde{\mathbf{x}}_t, \tilde{\mathbf{p}}_t) \end{aligned} \tag{2}$$

Equation 2 is the starting point of our approach. It defines an imprecise model of the supervised system with K uncertain parameters. Thus, this model has a K-dimensional uncertainty space. In order to divide this uncertainty space we have to define a *partition* $\tilde{\mathbf{q}}_t = [(\underline{q}_{1,t}, \overline{q}_{1,t}), (\underline{q}_{2,t}, \overline{q}_{2,t}), \cdots, (\underline{q}_{K,t}, \overline{q}_{K,t})]^T$ with $\tilde{\mathbf{q}}_t \subseteq \tilde{\mathbf{p}}_t$. A complete partitioning of the uncertainty space at any time t into M partitions must satisfy the following condition $\bigcup_m \tilde{\mathbf{q}}_t^{(m)} = \tilde{\mathbf{p}}_t$ where $m = 1, \ldots, M$. A model based on a partition of the uncertainty space is referred to as *subspace model*. From the definition of a partition, we can finally define the state of a subspace model m:

$$\begin{aligned} \tilde{\mathbf{x}}_t^{(m)} &= \mathbf{f}(\tilde{\mathbf{x}}_{t-1}^{(m)}, \mathbf{u}_{t-1}, \tilde{\mathbf{q}}_{t-1}^{(m)}) \\ \tilde{\mathbf{y}}_t^{(m)} &= \mathbf{g}(\tilde{\mathbf{x}}_t^{(m)}, \tilde{\mathbf{q}}_t^{(m)}). \end{aligned} \tag{3}$$

With the monotonicity assumption of $\mathbf{f}$ and $\mathbf{g}$ 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 of the *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 $Q_t^{(m)} = \{\tilde{\mathbf{q}}_{t,i}^{(m)}\}$ with $i = 1, \ldots, 2^K$. Thus, an uncertainty space of dimension K results in 2^K corner points. The states at the corner points can be represented as set

$$\begin{aligned} \mathbf{X}_t^{(m)} &= \{\mathbf{x}_{t,i}^{(m)} : \mathbf{x}_{t,i}^{(m)} = \mathbf{f}(\mathbf{x}_{t-1,i}^{(m)}, \mathbf{u}_{t-1}, \mathbf{q}_{t-1,i}^{(m)})\} \\ \mathbf{Y}_t^{(m)} &= \{\mathbf{y}_{t,i}^{(m)} : \mathbf{y}_{t,1}^{(m)} = \mathbf{g}(\mathbf{x}_{t,i}^{(m)}, \mathbf{q}_{t,i}^{(m)})\} \end{aligned} \tag{4}$$

where $\mathbf{q}_{t,i}^{(m)}$ is an exact parameter vector at time t from the subspace m and at corner $i = 1, \ldots, 2^K$ of this subspace. Note, that $\mathbf{x}_{t,i}^{(m)}$ are state vectors, and also $\mathbf{y}_{t,i}^{(m)}$ are output vectors with exact values. Note that this approach assumes that the parameters of the system are constant, and are not varying in time. This assumption will be discussed later.

This representation of an uncertain state is directly exploited by our consistency check for a given subspace m. First, a *residual* is calculated for each state at a corner point using the measurements at time t, i.e., $\mathbf{r}_{t,i}^{(m)} = \mathbf{y}_{t,measured} - \mathbf{y}_{t,i}^{(m)}$, where $\mathbf{r}_{t,i}^{(m)}$ has the same dimension J as $\mathbf{y}_{t,measured}$ and $\mathbf{y}_{t,i}^{(m)}$. Then, the minimum and maximum values of the residual are determined as

$$r_{t,min,j}^{(m)} = \min_i \{r_{t,i,j}^{(m)}\} \tag{5}$$

$$r_{t,max,j}^{(m)} = \max_i \{r_{t,i,j}^{(m)}\} \tag{6}$$

with $i = 1, \ldots, 2^K$, and $j = 1, \ldots, J$. Finally, subspace model m is checked for consistency simply by comparing the signs of $r_{t,min,j}^{(m)}$ and $r_{t,max,j}^{(m)}$. The subspace model m is consistent with the measurements, iff

$$\mathrm{sgn}(r_{t,min,j}^{(m)}) \neq \mathrm{sgn}(r_{t,max,j}^{(m)}) \tag{7}$$

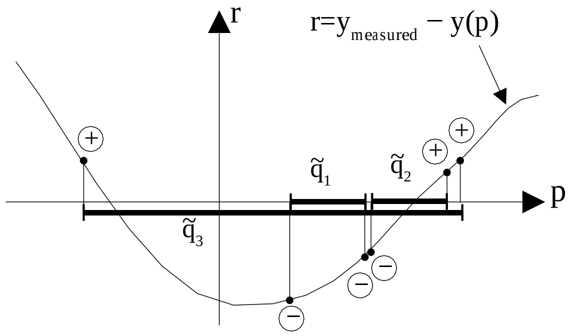

Figure 1. Consistency check with one uncertain parameter p and three subspaces $\tilde{q}_1$, $\tilde{q}_2$, and $\tilde{q}_3$. The residuals at the corner points of subspace $\tilde{q}_1$ are both negative, therefore, the model with the subspace $\tilde{q}_1$ is inconsistent with the measurement. In 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.

holds for all elements $j = 1, \ldots, J$.

Informally, Equation 7 checks whether the zero vector lies within the "residual subspace" (see Figure 1). If this equation is violated, the subspace model m is refuted. This simple consistency check holds also if not all elements of $\mathbf{y}$ are included in the measurements. In this case, a comparison with the missing elements is simply ignored. 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.

Due to the uncertainty in the parameters this method may result in diverging envelopes. This deviation of the predicted value to the "correct" value over time is referred to as *accumulation uncertainty*. In order to keep this deviation small we have also introduced a *dynamic* partitioning of the subspace models. During monitoring consistent subspaces are further partitioned resulting in smaller subspace models that potentially describe the supervised system more precisely [12].

3 MONITORING PIECEWISE CONTINUOUS BEHAVIORS

3.1 Monotonicity at Transitions

In order to extend our approach to monitoring piecewise continuous behaviors and discrete transitions, we must have a closer look at our monotonicity assumption. Remember that the result of our consistency check is only valid if the state values within the subspace are monotonic.

In general the monotonicity of the state values with regard to the parameters is not guaranteed by the monotonicity of the system equations $\mathbf{f}$ and $\mathbf{g}$. The monotonicity is only given when the following assumptions also hold:

1. the system input $\mathbf{u}$ does not change, and
2. the initial values of a subspace model are the same over its complete uncertainty space.

Both assumptions are important for monitoring discrete and continuous behaviors. The first assumption is especially relevant for transitions because they are often triggered by stepwise changes of the system input (e.g., caused by operator actions). Such transitions violate, therefore, the first assumption. The second assumption is a simple consequence of the integration of the given differential equation:

$$\mathbf{x}(t) = \mathbf{x}_{t_0} + \int_{t_0}^{t} \dot{\mathbf{x}}(\tau) d\tau \tag{8}$$

If the initial states $\mathbf{x}_{t_0}$ are different at some corners in the subspace model, the state values $\mathbf{x}_t$ may not be monotonic (even if $\dot{\mathbf{x}}$ is monotonic). However, monotonicity is guaranteed after some time.

As discussed above discontinuous transitions may result in a non-monotonicity of the state values with regard to the parameters (for a limited period of time), which in turn leads to an incorrect consistency check. Thus, to maintain a correct (and conservative) monitoring technique we must extend the consistency check by a check for monotonicity. If the monotonicity is not guaranteed the consistency check is simply ignored and this subspace can not be refuted. At some time after the transition the subspace may become monotonic again and the consistency check can be applied again.

3.2 Checking for Monotonicity

The monotonicity of the state values for an individual subspace is checked by the following method.

We define a matrix $\mathbf{B}(t, \mathbf{x}, \mathbf{p})$ with the elements

$$b_{ij}(t, \mathbf{x}, \mathbf{p}) = \frac{\partial \dot{x}_i(t, \mathbf{x}, \mathbf{p})}{\partial p_j}, \tag{9}$$

where t is the time, $\mathbf{x}$ the state vector, and $\mathbf{p}$ the parameter vector with its elements p_j. We also define the matrix $\mathbf{C}(t, \mathbf{x}, \mathbf{p})$ with the elements

$$c_{ij}(t, \mathbf{x}, \mathbf{p}) = \frac{dx_i(t, \mathbf{p})}{dp_j}. \tag{10}$$

The matrix $\mathbf{C}(t, \mathbf{x}, \mathbf{p})$ is calculated by

$$\frac{d\mathbf{C}(t, \mathbf{x}, \mathbf{p})}{dt} = \mathbf{A}(t, \mathbf{x}, \mathbf{p})\mathbf{C}(t, \mathbf{x}, \mathbf{p}) + \mathbf{B}(t, \mathbf{x}, \mathbf{p}), \tag{11}$$

where $\mathbf{C}(0, \mathbf{x}_0, \mathbf{p}) = \mathbf{0}$ (the empty matrix), and the matrix $\mathbf{A}(t, \mathbf{x}, \mathbf{p})$ is defined as

$$a_{ij}(t, \mathbf{x}, \mathbf{p}) = \frac{\partial \dot{x}_i(t, \mathbf{x}, \mathbf{p})}{\partial x_j}. \tag{12}$$

The elements $c_{ij}(t, \mathbf{x}, \mathbf{p})$ give us the trend of the state value $x_i(t, \mathbf{p})$ with regard to the parameter p_j. This is exploited by our *monotonicity check*: The state values of a subspace model are monotonic, iff

$$\text{sgn}(c_{ij,min}) = \text{sgn}(c_{ij,max}) \tag{13}$$

holds for all state values $i = 1, \ldots, I$ and all directions of the uncertainty space $j = 1, \ldots, K$. $c_{ij,min}$ are the appropriate values of $c_{ij}(t, \mathbf{x}, \mathbf{p})$ at the corner min, and $c_{ij,max}$ are the values of $c_{ij}(t, \mathbf{x}, \mathbf{p})$ at the corner max of that subspace model (as described with Equations 5 and 6).

Figure 2 depicts the monotonicity check. In general, the information at the corner points is not sufficient to decide on monotonicity. However, assuming the monotonicity of the functions $\mathbf{f}$ and $\mathbf{g}$ with regard to the parameter, the monotonicity check becomes sufficient.

The calculation of the monotonicity check implies a numerical solution of the differential equation (Equation 12). However, since we use also a differential description of the system ($\mathbf{f} = \dot{\mathbf{x}}$), the monotonicity check does not significantly increase the computational load. Note that matrix $\mathbf{A}$ is constant for linear systems.

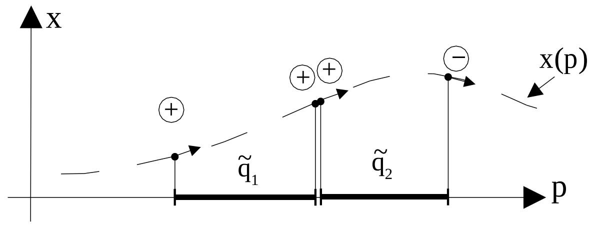

Figure 2. Monotonicity check with one state value and one parameter. To check the subspace model for monotonicity, the gradients of the state values with regard to the parameters are calculated at the corner points. In this example, the subspace $\tilde{q}_1$ is monotone and the subspace $\tilde{q}_2$ violates the monotonicity check.

4 THE MONOTONICITY CHECK IN A REAL-WORLD SYSTEM

We now examine the monotonicity behavior 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 the monitoring system via a RS 232 interface.

Our model which includes the three components with heating elements is given as

$$\begin{aligned} \dot{T}_1 &= \tfrac{1}{C_1}(q_{i1} - L_1(T_1 - T_0) - L_{12}(T_1 - T_2)) \\ \dot{T}_2 &= \tfrac{1}{C_2}(q_{i2} + L_{12}(T_1 - T_2) - L_2(T_2 - T_0) \\ &\quad - L_{23}(T_2 - T_3)) \\ \dot{T}_3 &= \tfrac{1}{C_3}(q_{i3} + L_{23}(T_2 - T_3) - L_3(T_3 - T_0)) \end{aligned} \tag{14}$$

where T_i is the temperature of the three components, C_i is the mass of the components, q_i is the heat flow into the components, L_i the thermal conductivity between the component i and the environment, L_{ij} the thermal conductivity between the component i and j, and T_0 the temperature of the environment. We can reduce the complexity of this model by exploiting the symmetric construction of the heating system ($L_3 = L_1$, $L_{23} = L_{12}$, $C_3 = C_1$) resulting in a total of five uncertain parameters.

The state vector is given as $\mathbf{x} = (T_1, T_2, T_3)^T$, the input vector as $\mathbf{u} = (q_{i1}, q_{i2}, q_{i3}, T_0)^T$, and the output vector as $\mathbf{y} = (T_1 + n_1, T_2 + n_2, T_3 + n_3)^T$, where n_i is the noise of each temperature sensor. The noise parameters are also included in the uncertainty space resulting in a total of eight uncertain parameters. Note that noise parameters are not dynamically partitioned into smaller intervals and they are not considered by the monotonicity check.

We have measured the input values with $q_{off} = 1.24W$ and $q_{on} = 34.8W$ (heating element is either turned off or turned on). With an initial refinement step, we get the parameter intervals as $L_1 = [0.12, 0.13]$, $L_2 = [0.15, 0.18]$, $L_{12} = [0.62, 0.73]$, $C_1 = [51, 54]$, $C_2 = [61, 65]$. The refinement step is performed in a single continuous behavior segment [12].

To examine the non-monotonic behavior in the system, we observe the system after a transition, and count the subspace models,

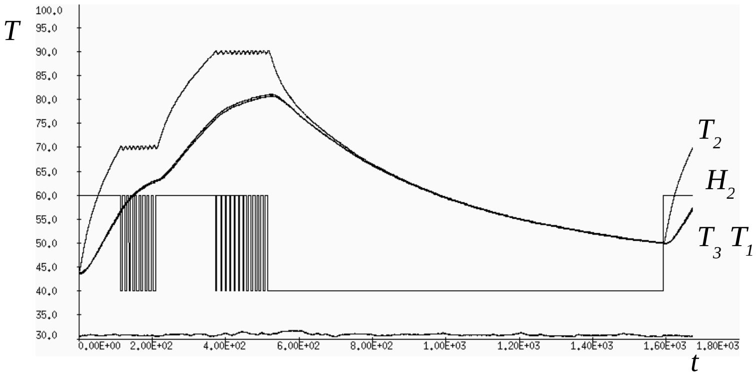

Figure 3. Measurements from the heating system used for monotonicity checking. The input H_2 is generated by the process control computer and sent to the monitoring system.

which are marked as non-monotonic. Over time, this gives us a picture, how the transition produce non-monotonicity in the state values. We choose the following scenario:

Control state 1: Heat T_2 until T_2 reaches 70. Then go to state 2.
Control state 2: Heat T_2, if $T_2 < 70$. If $t_{state2} \geq 100sec$, go to state 3.
Control state 3: Heat T_2, if $T_2 < 90$. If $t_{state3} \geq 100sec \wedge T_2 \geq 90$, go to state 4.
Control state 4: Do not heat. If $T_2 \leq 50$, go to state 1.

Figure 3 plots the resulting measurements for this scenario. The heating flag H_2 (generated by the PCC) is used, to get a discrete change of an input. To implement the heating element characteristic, we assume an additional mass C_h and a thermal conductivity L_{2h} between component 2 and the heating mass:

$$\dot{T}_{h2} = \frac{1}{C_h}(32.82H_2 - L_{2h}(T_{h2} - T_2)) \quad (15)$$

$$q_{i2} = 1.24 + L_{2h}(T_{h2} - T_2) \quad (16)$$

To demonstrate the non-monotonic effect after a stepwise change of an input, we check the monotonicity of all subspace models, and count non-monotonic subspace models, i.e., which violate Equation 13. Figure 4 shows a part of the scenario, where the temperature of component 2 is hold at 90 degree (control state 3). For this plot, we have started with 128 subspace models, and no dynamic partitioning is introduced. Due to the discrete controller the heating is turned on and off several times. At each transition about 40 subspace models are non-monotonic. An interesting observation in this figure is, that the non-monotonic subspaces disappear quickly, if the heating flag is turned off only for a short time.

Figure 5 shows the number of the non-monotonic subspace models after control state 3. The peak here is about 30 subspace models. It shows, that non-monotonic subspace models are also existing for a "longer" time period (here about 400 seconds) after the last discrete change of an input.

Non-monotonic subspace models are not refuted, and, therefore, do not make any contribution to decrease the uncertainty space. Although the number of non-monotonic subspace models are quite high (about 50 percent of the current subspace models) for some times. it has not a significantly influence to the refutation. The reason is, however, that such peaks does not hold for long time, so the consistency check soon becomes valid again. At this example the number of consistent subspace models at the end of the scenario is about 20.

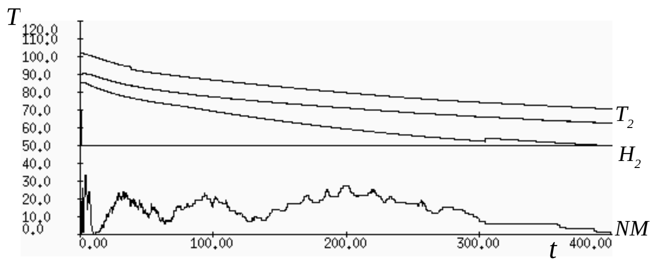

Figure 5. The non-monotonicity after the switching period. Drawn are (as same as in figure 4) the measurement and the envelopes of T_2, the heating flag H_2 and the number of non-monotonic subspace models NM. Some subspace models are non-monotonic after the heating period.

5 DISCUSSION

In this paper, we have presented a model-based monitoring approach based on uncertainty space partitioning. The fundamental assumption of this approach is the monotonicity of the state values with regard to the range of the parameters. In systems which exhibit both discrete and continuous behaviors the monotonicity can not be guaranteed only by the monotonicity of the vector functions. Thus, in order to apply our basic approach to monitor hybrid systems, we have introduced a monotonicity check for the state values.

Note the difference of monitoring based on pre-calculated envelopes with our approach. With pre-calculated envelopes, the envelopes remain constant over the complete monitoring process. In our approach, the envelopes may become smaller than the initial ones due to the refutation of inconsistent subspaces during monitoring. This results in an earlier detection of faults. However, there is a significant increase in the computational load of subspace partitioning.

Our approach is based on computing the envelopes of differential equations. For complex models, the overall runtime of our monitoring algorithm is dominated by solving the differential equations, especially when a high-precise method such as Runge-Kutta is used. The computational complexity of our algorithm for a single time-step can be estimated as

$$\mathcal{O}(M2^K(\rho + \mu)) \quad (17)$$

where M is the number of partitions, K is the number of uncertainty parameters, ρ is the time of the Runge-Kutta algorithm, and μ is the time of the matrix multiplication according to Equation 11. The time ρ strongly depends on the dynamic properties of the system, and for high dynamic systems, the assumption $\rho > \mu$ holds.

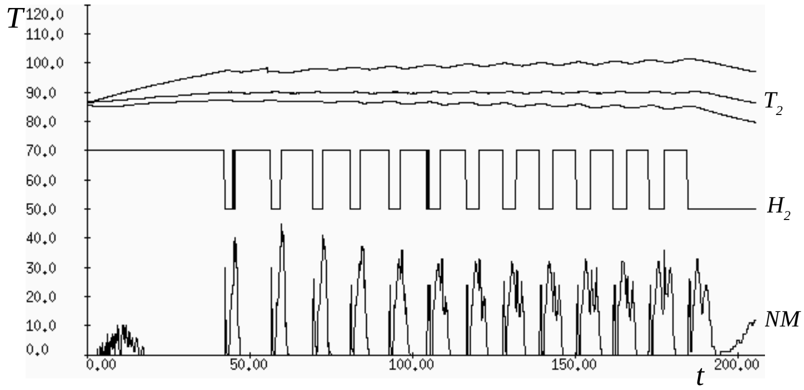

Figure 4. First overview of the monotonicity of the technical system. Drawn are the measured T_2 with its envelopes, H_2 is the heating flag for the second component, and NM is the number of the non-monotone subspace models. The discrete change of the input makes a directly effect to the monotonicity of the state values.

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. Measurement noise can also be handled by introducing additional uncertainty parameters into the model.

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.

Model-based monitoring using uncertainty space partitioning is related to the interval identification algorithm of Schaich et al. [13]. 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. Petridis an Kehagias [10] have also developed an algorithm with subspace partitioning. The partitioning is only performed in advance and the consistency check is based on probabilities depending on the noise in the system. Other work in monitoring [7, 9, 3] uses multiple models for fault detection. These models represent known faults of the supervised system. From the viewpoint of system identification, our approach is closely related to semi-quantitative system identification [8]. Identification of both approaches are grounded on the refutation of subspace models that are known to be inconsistent with the measurements. Semi-quantitative system identification performs refinement at the qualitative and interval level. Semi-quantitative system identification has also been applied to model-based monitoring [11]. Bonarini and Bontempi [4] have developed a quite similar approach to our consistency check. However, they have focused on uncertainty initial state values, which are given as intervals. Also related to our work is Armengol et al. [1, 2]. The simulation is based on modal interval arithmetics, 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 are between the underbounded and overbounded envelopes). In comparison to our approach, we simulate at each corner of the uncertainty space, which leads to exact envelopes (no false and missed alarms, according to observability) for linear systems.

Directions for future work include (i) the incorporation of (unknown) discontinuous transitions in our monitoring approach, (ii) further investigations on the monotonicity properties after a discontinuous transition, especially in the context of non-linear systems, and (iii) the improvement of the dynamic uncertainty space partitioning.

REFERENCES

[1] Joaquim Armengol, Louise Trave-Massuyes, 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).

[2] Joaquim Armengol, Josep Vehi, Louise Trave-Massuyes, and Miguel Angel Sainz, ‘Application of Multiple Sliding Time Windows to Fault Detection Based on Interval Models’, *12th International Workshop on Principles of Diagnosis (DX-01)*, 9–16, (2001).

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

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

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

[6] *Readings in Model-based Diagnosis*, eds., Walter Hamscher, Luca Console, and Johan de Kleer, Morgan Kaufmann, 1992.

[7] 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, (2000).

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

[9] 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).

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

[11] 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)*, pp. 1080–1086, Stockholm, Sweden, (August 1999). Morgan Kaufmann.

[12] 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)*, (February 2002).

[13] David Schaich, Rudibert King, Uwe Keller, and Mike Chantler, ‘Interval Identification - a Modelling and Design Technique for Dynamic Systems’, in *QR99*, 13^{th} *International Workshop on Qualitative Reasoning*, (June 6-9 1999).