In *Proceedings of SAFEPROCESS'03 (Bridge-Session),*
Washington, DC, June, 2003

MODEL REFINEMENT FOR MONITORING – REFUTATION VS. TRADITIONAL PARAMETER ESTIMATION

Andreas Doblander, Bernhard Rinner, Ulrich Weiss

Institute for Technical Informatics,
Graz University of Technology, AUSTRIA

Abstract:
Model-based monitoring and diagnosis systems must be able to express and reason with incomplete knowledge. However, it is desired to refine the imprecision in the underlying model when more measurements from the supervised system are available. Imprecision is often specified by intervals of model parameters. In this paper we compare the two refinement methods refutation and parameter estimation in the context of monitoring. Refutation removes parts of the parameter intervals that are provable inconsistent with the measurements. Parameter estimation, on the other hand, searches for exact parameter values that best match the measurements. This comparison is supported by various experiments with the refutation-based refinement implemented in the MOSES monitoring system and the MATLAB system identification toolbox.

Keywords: model–based monitoring; parameter estimation; interval models; system identification;

1. INTRODUCTION

In case of an unexpected fault knowledge about the supervised system is per definition incomplete. Model-based monitoring and diagnosis systems must, therefore, be able to express and reason with incomplete knowledge. However, it is desired to refine the imprecision in the underlying model when more measurements from the supervised system are available. Imprecision is often specified by intervals of model parameters.

This paper focuses on the refinement process of imprecise models, i.e., we compare the two refinement methods *refutation* and *parameter estimation* in the context of monitoring. Both methods start with a parameterized differential equation as system model. Refutation removes parts of the initial parameter intervals that are provable inconsistent with the measurements. Parameter estimation, on the other hand, searches for exact parameter values that best match the measurements.

Refutation-based refinement expands ideas from semi-quantitative system identification (Kay *et al.*, 2000) and its application to monitoring (Rinner and Kuipers, 1999). The refutation-based refinement method used in this comparison has been implemented in the monitoring system MOSES (Rinner and Weiss, 2002*a*; Rinner and Weiss, 2003). Parameter estimation is a traditional refinement method (Ljung, 1999); various estimation techniques have been implemented in the SYSTEM IDENTIFICATION TOOLBOX from MATLAB (The Mathworks, Inc., 2001).

The remainder of this paper covers these parameter refinement methods in more detail. Section 2

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

introduces the refutation refinement method. Section 3 compares the refinement procedures of both methods and Section 4 presents experimental results using MOSES and the SYSTEM IDENTIFICATION TOOLBOX for MATLAB. A summary and a discussion about related work conclude this paper.

2. SUBSPACE REFUTATION

Both refinement methods compared in this paper are based on a parameterized differential equation model. This section briefly summarizes the imprecise modeling and the refutation of subspaces used in MOSES.

In general, a technical system is modeled as a linear system

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

where $\mathbf{x}(t)$ is the state vector, $\mathbf{u}(t)$ is the input vector, $\mathbf{p}$ is the parameter vector, $\mathbf{y}(t)$ is the output vector, and $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ are matrices with appropriate dimensions and functions of $\mathbf{p}$. In an imprecise model $\mathbf{p}$ is a vector of intervals instead of exact numerical values, i.e., $\tilde{\mathbf{p}} = [(\underline{p}_1, \overline{p}_1), (\underline{p}_2, \overline{p}_2), \ldots, (\underline{p}_K, \overline{p}_K)]^T$. The system is then imprecisely specified as:

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

The model imprecision is now represented by the K-dimensional uncertainty space specified by $\tilde{\mathbf{p}}$. In order to refute parts of this uncertainty space we must first divide it into smaller regions (partitions) $\tilde{\mathbf{q}} = \left[(\underline{q}_1, \overline{q}_1), (\underline{q}_2, \overline{q}_2), \ldots, (\underline{q}_K, \overline{q}_K)\right]^T$ with $\tilde{\mathbf{q}} \subseteq \tilde{\mathbf{p}}$. Each partition defines a *subspace*, and for each subspace m a model can be stated as follows:

$$\begin{aligned}\dot{\tilde{\mathbf{x}}}^{(m)}(t) &= \mathbf{A}(\tilde{\mathbf{q}}^{(m)})\tilde{\mathbf{x}}^{(m)}(t) + \mathbf{B}(\tilde{\mathbf{q}}^{(m)})\mathbf{u}(t) \\ \tilde{\mathbf{y}}^{(m)}(t) &= \mathbf{C}(\tilde{\mathbf{q}}^{(m)})\tilde{\mathbf{x}}^{(m)}(t) + \mathbf{D}(\tilde{\mathbf{q}}^{(m)})\mathbf{u}(t)\end{aligned} \quad (3)$$

If the monotonicity of $\mathbf{x}$ and $\mathbf{y}$ with regard to $\mathbf{p}$ is given (Rinner and Weiss, 2002*b*), the trajectories of all subspaces can be derived by computing the integral only at special (extremal) points of the subspaces. Thus, the trajectories of the imprecise model can be computed by standard numerical methods starting at the extremal points of the subspaces. The number of extremal points is exponential in the number of uncertain parameters, and therefore the computational effort for computing the trajectories strongly increases with the number of uncertain parameters.

Each subspace model can be checked for consistency with the measurement by checking whether the sample at time t (superimposed by a fixed noise interval) lies within the computed trajectories of the subspace model at t. When there is no overlap the subspace model is inconsistent and refuted from further processing. The corresponding partition is then also refuted and the model's imprecision has been refined.

3. COMPARISON OF THE TWO REFINEMENT METHODS

A brief comparison of the refutation approach and traditional parameter estimation is summarized in Tab. 1. The most important requirements for the two approaches as well as their main features are informally described in this section.

3.1 Refinement by Refutation

At all extremal points of each subspace trajectories are computed using a standard Runge–Kutta solver. All these subspace model behaviors are checked for consistency with the measured data, and inconsistent models are refuted. The above steps are repeated for all data samples. As more subspace models get inconsistent and, therefore, are refuted the parameter space gets smaller.

Note that for refutation the only requirement for measurement noise is that its amplitude must be bounded. No special distribution has to be assumed. Measurement noise can, therefore, be modeled as an interval parameter.

Currently, only linear models with time-invariant parameters are supported. Although, slight nonlinearities may be covered by wider parameter intervals. Since refutation is an incremental method, it can be applied online. However, depending on the number of uncertain parameters and the number of partitions the refinement can be a very time and memory consuming task.

Refutation keeps the implicit correspondence between model parameters to physical parameters. It does not result in parameter values outside the initial interval and, therefore, does not violate the modeler's interpretation of the parameters.

3.2 Traditional Parameter Estimation

The starting point for the standard parameter estimation is also a parameterized model structure. Contrary to refutation, many different types of models can be used for parameter estimation. A lot of different methods have been developed for various application domains. For the purpose of this work mainly methods for parameterized

	Refutation	Parameter Estimation
Requirements		
model structure	known a priori	not required
model	linear systems	linear systems
parameters	intervals, time-invariant	exact values, time-variant possible
measurement noise	bounded amplitude noise	filtered white noise
initial values	noncritical	critical for convergence
Features		
mode of operation	online	online
discontinuous changes	possible	possible
refined model	reduced parameter intervals	exact parameter values
derived system behavior	guaranteed bounds	best matching single behavior
computational load	strongly dependent on uncertain parameters	moderate
uninformative data	insensitive	may converge to wrong model
parameter interpretation	maintains physical interpretation	interpretation may get lost

Table 1. Summary of important requirements and features of the refinement methods refutation and traditional parameter estimation.

state–space representations are used, i.e., only the prediction error method (PEM) is applied. In short, a residual, the prediction error, is calculated from the predicted and the measured value. This residual is a function of the parameters. Minimizing the norm of the prediction errors with respect to the parameters yields the parameter's values that best fit the data.

As an iterative method PEM needs initial parameter values to start the iteration. Unfortunately, the choice of these initial values is critical for the convergence. The iteration may get stuck at a local minimum (Ljung, 1999). There are also some requirements for the measurement noise. To ensure correct estimates, it has to be filtered white noise. The measured data has to be informative about interesting frequency ranges. Parameter estimation results in exact values for parameters. Therefore, only a best matching single behavior of the analyzed system is derived. There are no guaranteed bounds on the system behavior as in refutation.

Traditional parameter estimation methods are applied for time–invariant and linear models. Recursive formulations of standard algorithms are for online operation and can cope with time–variant parameters. For the computation of the estimates efficient algorithms are available. The computational load is kept within acceptable limits. As a widely used implementation of parameter estimation methods the system identification toolbox (Sitb) for Matlab (Ljung, 2001) is used in this work.

Parameter estimation may result in completely different parameter values than their initial values. This strongly complicates the physical interpretation of the parameters which is important in monitoring and diagnosis applications.

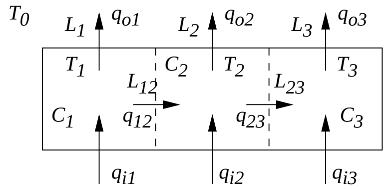

Fig. 1. The heating system. L_1, L_2, L_3, L_{12}, L_{23}, C_1, C_2 and C_3 are the system parameters. T_1, T_2 and T_3 are the element temperatures and T_0 is the environment temperature. The q_{jk} are heat flows.

4. EXPERIMENTAL RESULTS

In order to compare the performance of the refinement methods several experiments have been conducted. Data from simulated processes as well as a real technical system are used as input to Moses and Sitb.

Basis for all experiments is a laboratory scale heating system which consists of three heating/cooling elements mounted on a thermal conductive plate. A process control computer (B&R 2003) controls the individual elements and transfers the measured data to the monitoring system.

For our experiments the heating system is modeled as an imprecise linear differential equation (Fig. 1) (Rinner and Weiss, 2002*a*), and only the heating source q_{i2} and environment temperature T_0 are considered as inputs. The outer heating elements (q_{i1} and q_{i3}) remain switched off. In overall there are eight uncertain parameters in the model. These are the thermal conductivities L_i and the thermal masses C_i, i.e., $p_1, \ldots, p_3$ correspond to $L_1, \ldots, L_3$, p_4 and p_5 correspond to L_{12} and L_{23}, and $p_6, \ldots, p_8$ correspond to $C_1, \ldots, C_3$.

The structure of this model is given as (cp. Eq. 1)

$$\mathbf{A} = \begin{pmatrix} -\frac{p_1+p_4}{p_6} & \frac{p_4}{p_6} & 0 \\ \frac{p_4}{p_7} & -\frac{p_2+p_4+p_5}{p_7} & \frac{p_5}{p_7} \\ 0 & \frac{p_5}{p_8} & \frac{p_3+p_5}{p_8} \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} 0 & \frac{p_1}{p_6} \\ \frac{1}{p_7} & \frac{p_2}{p_7} \\ 0 & \frac{p_3}{p_8} \end{pmatrix}, \mathbf{C} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4)$$

and the empty matrix **D**.

The input vector consists of two elements. The binary controller signal for heating element q_{i2} and the environment temperature T_0. Transitions of the binary signal can be seen as mode changes. In case of the heating system the controller signal corresponds to on and off states of the heater.

In our experiments we have used three different data sets as input to the refinement methods. First, data is simulated from a linear SIMULINK model corresponding to Eq. 4. In this case all parameters are exactly known. Second, the SIMULINK model is slightly modified by introducing a moderate nonlinear saturation effect, i.e., the parameter vector slightly depends on the system state. Both simulated data sets are superimposed by an uniformly distributed noise of variance 0.2. Finally, data from the real heating system is used. In this case we have only a very imprecise knowledge about the parameter values, and we do not know the properties of the measurement noise.

All experiments have been performed using MOSES implemented on a standard PC running LINUX and the SITB in version 5.0.1 under WINDOWS.

4.1 Simulated linear Process

In this experiment the refinement capabilities are compared using simulated data derived from a linear model with known parameter values. The results of the parameter refinement with the SITB and MOSES are shown in Tab. 2 and Tab. 3, respectively.

In this case parameter estimation excels over refutation. Even with data superimposed by rather large noise the estimated values are very close to the real parameter values.

As an additional experiment, the model resulting from parameter estimation is used to monitor the underlying process, i.e., the exact model and an interval for the measurement noise are used as model in MOSES. As expected the estimated values (cf. Tab. 2) allow the linear behavior to be tracked without any false alarms.

	Real Value	Initial Value	Estimate
p_1	0.12	0.1	0.1191
p_2	0.1534	0.2	0.1553
p_3	0.12	0.1	0.1191
p_4	0.65	0.8	0.6452
p_5	0.649	0.8	0.6452
p_6	50	40	49.7288
p_7	59	70	59.5756
p_8	50	40	49.7288

Table 2. Refinement of linear model parameters using the prediction error method (PEM) from the SITB.

	Real Value	Initial Interval	Refined Interval
p_1	0.12	(0.1, 0.2)	(0.1, 0.15)
p_2	0.1534	(0.1, 0.2)	(0.15, 0.2)
p_3	0.12	(0.1, 0.2)	(0.1, 0.15)
p_4	0.65	(0.5, 0.8)	(0.5, 0.8)
p_5	0.649	(0.5, 0.8)	(0.5, 0.8)
p_6	50	(33, 100)	(33, 100)
p_7	59	(33, 100)	(33, 100)
p_8	50	(33, 100)	(33, 100)

Table 3. Refinement of linear model parameters by MOSES.

4.2 Simulated Process with a nonlinear Effect

In this experiment the comparison is based on simulated data from a nonlinear system model with known parameters. The results of the parameter refinement with the SITB and MOSES are shown in Tab. 4 and Tab. 5, respectively.

Noise and the nonlinear behavior can cause problems for parameter estimation. As can be seen in Tab. 4 the estimated parameters are very different from the real ones. However, the resulting behavior describes data from the process with acceptable accuracy. But with diagnosis in mind getting an accurate simulation of a process is not the only desired result. Moreover, it is also preferable to retain a mapping of physical system components to model parameters. For example parameter p_8 corresponds to a thermal mass. A negative value for this parameter makes a physical interpretation difficult.

In Tab. 5 the results of MOSES are presented. With such heavy noise the refinement by refutation is moderate. Although, MOSES does not converge to wrong parameters. More significant nonlinear effects, however, result in the need for wider parameter intervals to be able to track system behaviors. Such a case is undesirable because fault detection time increases with growing parameter intervals.

Monitoring the given process in MOSES using exact values computed by the SITB results in an early false alarm. As can be verified by Fig. 2, the exact model leads to a false alarm at time $t = 62s$. Even tough a noise level of 1.5 (instead of 0.2 as in the data creating process) was specified for the monitoring algorithm. Note that although

	Real Value	Initial Value	Estimate
p_1	0.12	0.1	0.8984
p_2	0.1534	0.2	0.1674
p_3	0.12	0.1	-0.6360
p_4	0.65	0.8	4.8738
p_5	0.649	0.8	-3.4399
p_6	50	40	377.0969
p_7	59	70	63.3121
p_8	50	40	-261.9042

Table 4. Refinement of a model with saturation effect computed using SITB.

	Real Value	Initial Interval	Refined Interval
p_1	0.12	(0.1, 0.2)	(0.10, 0.15)
p_2	0.1534	(0.1, 0.2)	(0.15, 0.20)
p_3	0.12	(0.1, 0.2)	(0.10, 0.15)
p_4	0.65	(0.5, 0.8)	(0.5, 0.8)
p_5	0.649	(0.5, 0.8)	(0.5, 0.8)
p_6	50	(33, 100)	(33, 100)
p_7	59	(50, 100)	(50, 100)
p_8	50	(33, 100)	(33, 100)

Table 5. Refinement of a model with saturation effect computed by MOSES.

exact model parameter values are used bounding envelopes are derived by MOSES because the noise interval is treated as an additional parameter.

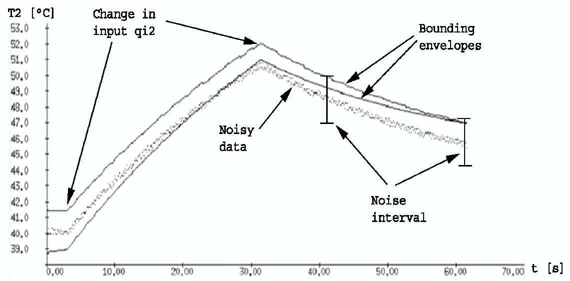

Fig. 2. Monitoring a (simulated) linear process with slight nonlinearities. The system parameters are computed by the SITB.

4.3 Real-world Heating System

In this experiment data from the real-world heating system is used. Since our knowledge about the parameter values is very limited we start the refutation-based refinement process with very large intervals because we want to be sure that the "real" values lie within the initial intervals. Using data from the fault free process, MOSES is now able to dramatically refine the intervals. This refinement is shown in Tab. 6. As a measure for the model imprecision the uncertainty space is stated as the product of the interval widths of all parameters. The uncertainty space is reduced by several orders of magnitude. Note that the achieved refinement is much better compared to the previous experiments because there is less noise in the data.

With the SITB, on the other hand, only exact parameter values are computed. Tab. 7 presents

	Initial Interval	Resulting Interval
p_1 (L_1)	(0.01, 1)	(0.11, 0.13)
p_2 (L_2)	(0.01,1)	(0.14, 0.18)
p_3 (L_3)	(0.01, 1)	(0.11, 0.13)
p_4 (L_{12})	(0.01, 10)	(0.62, 0.8)
p_5 (L_{23})	(0.01, 10)	(0.62, 0.8)
p_6 (C_1)	(5, 200)	(48, 63)
p_7 (C_2)	(5, 200)	(57, 66)
p_8 (C_3)	(5, 200)	(48, 63)
Uncertainty Space	3.72E+05	1.94E-02

Table 6. Refinement with data from the heating system computed by MOSES.

	Initial Value	Estimate
p_1 (L_1)	0.1	0.1342
p_2 (L_2)	0.2	0.1013
p_3 (L_3)	0.1	0.1342
p_4 (L_{12})	0.8	0.6435
p_5 (L_{23})	0.8	0.6435
p_6 (C_1)	40	68.0733
p_7 (C_2)	70	74.3107
p_8 (C_3)	40	68.0733

Table 7. Refinement of the heating system model computed using SITB.

these results computed with the PEM routine from the SITB. Like previous results this experiment shows the strength of MOSES compared to the SITB.

Again, using the model from the SITB for monitoring the process leads to an early false alarm. This situation is depicted in Fig. 3. As in the previous case there is not a single predicted behavior but two bounding envelopes due to measurement noise.

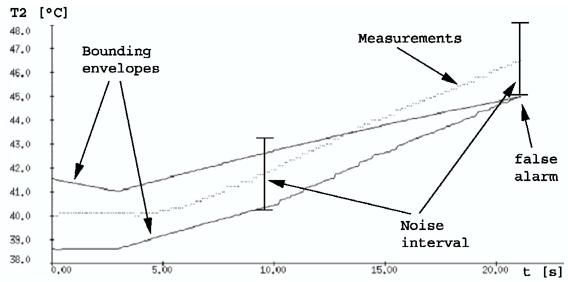

Fig. 3. Monitoring the heating system using exact parameters computed by the SITB.

5. DISCUSSION

In this paper we have compared the two refinement methods refutation and parameter estimation. Especially in monitoring and diagnosis, refinement is an important feature since the initial knowledge about the supervised system is imprecise. It is desired to (automatically) reduce the model imprecision by exploiting the measurements from the healthy system. This refinement can be applied either offline prior to monitoring or online.

Both methods refine or estimate the parameters of a linear differential equation model. Refutation narrows the initial parameter intervals, keeps the correspondence of the model parameters to the physical parameters, and is insensitive to uninformative data. Parameter estimation on the other hand searches for best matching parameter values and can cope with time-variant parameters. The convergence of parameter estimation is sensitive to the initial parameter values, and the important correspondence between model and physical parameters is more complicated and requires more complex (nonlinear) models (Frank *et al.*, 2000). The result of the parameter estimation also depends on the norm used in the prediction error method (PEM). In this comparison we only applied the standard quadratic norm. The computational load for refutation may be very high, especially when the number of uncertain parameters and partitions is large.

Related work has been done by (Bradley *et al.*, 2001). In their framework PRET nonlinear models are automatically constructed using experiments on a physical system. PRET focuses on system identification and not on monitoring. Petridis and Kehagis (Petridis and Kehagias, 1998) have developed a parameter estimation algorithm for dynamic, nonlinear systems. They partition the uncertainty space into a number of different models with exact parameter values. The trajectories of these models are computed simultaneously, and the model with the smallest deviation to the measurements is selected. The deviation is computed based on a stochastic representation and results in probabilities for the individual models. Finally, there are several examples for applying interval models and/or parameter estimation for monitoring technical system, e.g., (Tornil *et al.*, 2000), (Armengol *et al.*, 2000), (Narasimhan *et al.*, 2002).

The presented empirical comparison can be deepend and extended in various ways. Thus, future work may include (i) an evaluation of different norms for the predictive error method (PEM), (ii) a formal investigation of the theoretical properties of both approaches, and (iii) different case studies.

REFERENCES

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

Bradley, Elizabeth, Matt Easley and Reinhard Stolle (2001). Reasoning about nonlinear system identification. *Artificial Intelligence* **133**, 139–188.

Frank, P.M., E. Alorca Garcia and B. Köppen-Seliger (2000). Modelling for fault detection and isolation versus modelling for control. *Mathematics and Computers in Simulation* **53**, 259–271.

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

Ljung, Lennart (1999). *System Identification: Theory for the user*. Information and System Sciences. second ed.. Prentice Hall.

Ljung, Lennart (2001). System identification toolbox – user's guide.

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.

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

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 Partitining. In: *Proceedings of the Thirteenth International 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.

Rinner, Bernhard and Ulrich Weiss (2003). Online Monitoring of Hybrid Systems using Imprecise Models. In: *Proceedings of the 5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS 2003)*. Washington, DC, U.S.A.

The Mathworks, Inc. (2001). http://www.mathworks.com.

Tornil, Sebastian, Teresa Excobet and Vicenc Puig (2000). Fault Detection Using Interval Models. In: *Proceedings of the 4th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2000)* (A.M. Edelmayer, Ed.). Budapest, Hungary. pp. 1180–1185.