IMTC 2003 – Instrumentation and Measurement
Technology Conference
Vail, CO, USA, 20-22 May 2003

Anytime Fuzzy Modeling Approach for Fault Detection Systems

Annamária R. Várkonyi-Kóczy [1,2], Péter Baranyi[2,3], Ron J. Patton[4]
[1]Dept. of Measurement and Information Systems, Budapest University of Technology and Economics
Magyar Tudósok krt. 2., H-1521 Budapest, Hungary
[2]Integrated Intelligent Systems, I[2]S Japanese-Hungarian Laboratory
[3]Dept. of Telecommunication and Telematics, Budapest University of Technology and Economics
Magyar Tudósok krt. 2., H-1521 Budapest, Hungary
[4]Dept. of Electronic Engineering, The University of Hull, Hull, HU6 7RX, UK

Abstract* – Nowadays for solving fault diagnosis, fault isolation, and control problems a possible solution can be the use of model-based approaches, which contain a representation of our knowledge about the fault-free and faulty systems. If the nature and/or the actual circumstances of the problem in hand change the corresponding model should also be changed. Anytime techniques are very flexible in this respect and can advantageously be used when the operation should be performed under changing circumstances. This paper provides the possible anytime extension of the modeling issues of Takagi-Sugeno fuzzy inference operator based approximation which is the essential core of the parallel distributed compensation (PDC) design approach. The PDC based system analysis and the fuzzy observer design are focused on a benchmark problem of fault diagnosis of an actuator in a sugar factory.

***Keywords* - anytime systems, system monitoring, fault diagnosis, multiple-model methods, singular value decomposition, tensor product transformation, parallel distributed compensation**

I. INTRODUCTION

Nowadays practical solutions of engineering problems involve model-integrated computing. This integration means that the available knowledge is represented in a proper form and acts as an active component of the procedure (computer program) to be executed during the operation of the diagnostics, measuring, control, etc. devices. Thus, model based approaches offer a very challenging way to integrate a priori knowledge into the procedure.

In case of linear problems very well established methods are available and they are successfully combined with adaptive techniques to provide optimum performance. Nonlinear techniques, however, are far from this maturity. There is a wide variety of possible models to be applied based on both classical methods [1] and recent advances in handling information [2] but up till recently practically no systematic method was available which could be offered to solve a larger family of nonlinear engineering problems. The efforts based on fuzzy and neural network (NN) approaches seem to result in a real breakthrough in this respect (see e.g. [3]). These techniques can be applied even in cases when no analytical knowledge is available about the system, the information is uncertain or inaccurate, or when the available mathematical form is too complex to be used. Although, the use of fuzzy and NN models is limited by their exponentially increasing complexity and we can easily get into trouble when the problem is very complex with probably large number of parameters (especially when only partial and uncertain measured data is available about the system to be modeled) and the task – measurement, signal processing, fault detection and diagnosis, system control, etc. - should be carried out "on-line", i.e. the processing time is limited or crucial.

In computer-based monitoring and diagnostic systems the operations should be performed under prescribed response time conditions. It is an obvious requirement to provide enough computational power but the achievable processing speed is highly influenced by the precedence, timing, and data access conditions of the processing itself. It seems to be unavoidable even in the case of extremely careful design to get into situations where the shortage of necessary data and/or processing time becomes serious. Such situations may result in a critical breakdown of the monitoring and/or diagnostic systems [4]. Hence, to find an optimal trade-off between the modeling accuracy and processing complexity is an emerging issue in many real time-consuming implementations.

The concept of "anytime" processing tries to handle the case of too many abrupt changes and their consequences in larger scale embedded systems [5]. The idea is that if there is a temporal shortage of computational power and/or there is a loss of some data the actual operations should be continued to maintain the overall performance "at lower price", i.e., information processing based on algorithms and/or models of simpler complexity should provide outputs of acceptable quality to continue the operation of the complete system. The accuracy of the processing will be temporarily lower but possibly still enough to produce data for qualitative evaluations and supporting decisions. Consequently "anytime" processing provides short response time and is very flexible with respect to the available input information and computational power.

The processing time in fault diagnostics systems can be crucial in preventing and avoiding system shutdown, breakdown, and even catastrophes. In this regard, this paper offers a possible way to find the approximation trade-off and to cope with the available computational resources and information.

Takagi-Sugeno (TS) [6] and Takagi-Sugeno-Kang (TSK) [7] modeling approaches combined with Singular Value Decomposition (SVD) techniques are serious candidates to overcome the above outlined information lack, computational complexity, and time limitation problems. This latter becomes possible by the anytime extension of the modeling. SVD has successfully been used to reduce the complexity of a large family of systems based on both classical and soft techniques [8]. An important advantage of the SVD reduction technique is that it offers a formal measure to filter out the redundancy (exact reduction) and also the weakly contributing parts (non-exact reduction). This implies that the degree of reduction can be applied according to the maximum acceptable error corresponding to the temporal circumstances. In case of multi-dimensional problems, the SVD technique can be defined in a tensor product form, i.e. Higher Order SVD (HOSVD) or tensor product transformation (TPT) can be applied.

Besides the appropriate model approximation – which is the first step towards fault diagnosis and isolation (FDI), another important modeling issue is how to link the identified model to further analysis and design techniques. Increasing attention is paid on parallel distributed compensation (PDC) analysis and design techniques in FDI systems [9]. In our approach the TPT and PDC designs are executed numerically by computer without analytic and heuristic derivations. If we can determine an immediate link between these approaches then we may be able to design self developing and learning fault detection systems.

In this paper an intelligent and flexible anytime fuzzy modeling approach is introduced which is suitable for the approximate modeling of complex systems opening a way for the fault diagnosis and isolation in the system. In section II. the main idea of anytime processing is discussed. Section III. is devoted to the fuzzy modeling based fault detection and isolation while in Section IV. the above ideas are applied to the fault diagnosis and control of a real sugar factor plant.

II. ANYTIME PROCESSING

Today there is an increasing number of applications where the computing must be carried out on-line, with a guaranteed response time and limited resources. Moreover, the available time and resources are not only limited but can also change during the operation of the system.

Good examples are the modern computer-based diagnostics and monitoring systems, which are able to supervise complex industrial processes and determine appropriate actions in case of failures or deviation from the optimal operational mode. In these systems, for the diagnosis usually the model of the faultless system together with several typical fault models are used and the evaluation of the system model must be carried out on-line, thus the model must not only be correct, but also treatable by the limited resources during limited time. Moreover, if some abnormality occurs in the system's behavior, some kind of fault diagnostic task should also be completed meaning the reallocation of a part of the finite resources from the evaluation of the system model to this task. Also in case of an alarm signal, lower response time may be needed.

In these cases, the so-called anytime algorithms and systems [10] can be used advantageously, which are able to provide guaranteed response time and are flexible in respect to the available input data, time, and computational power. This flexibility makes able these systems to work in changing circumstances without critical breakdowns in the performance. Naturally, while the information processing can be maintained, the complexity must be reduced, thus the results of the computing become less accurate [11].

The algorithms/computing methods, which are suitable for anytime usage have to fulfill some expectations of vital importance:

- Low complexity: for the optimized behavior the results must be produced with as little computing as possible, i.e. the redundant calculations must be omitted.
- Changeable, guaranteed response time/computational need and accuracy: the achievable accuracy of the results and the necessary amount of computing time/resources must be flexibly changeable and usually known in advance.
- Known error: the error, originating from the necessary model "fitting" (e.g. complexity reduction resulting in non-exact solutions) must also be known, to be able to find the optimal or at least still acceptable solution in the given circumstances, and to be able to compute the resultant error of the outputs.

Iterative algorithms are popular tools in anytime systems, because their complexity can be easily and flexibly changed. These algorithms always give some - not accurate - result, and more and more accurate results can be obtained, if the calculations are continued. A further advantageous aspect of iterative algorithms is that we don't have to know the time/resource-need of a certain configuration in advance, the calculations can be continued until the results are needed.

Then by simply stopping the calculations, always in the given conditions achievable most accurate results are obtained. If PSGS fuzzy systems (product-sum-gravity with singleton consequents) are used the iterative evaluation can be applied [12].

Unfortunately, the usability of iterative algorithms is limited. While there is a wide range of problems, which can be solved by iterative algorithms, adequate evaluation methods can not always be found. Not only the existence of some kind of iterative evaluation method can be a problem, even more frequently the accuracy of the results is unknown: we only know, that the algorithm gives more and more accurate results, but we do not know, how much time is needed to achieve a given accuracy or what will be the rate of error if the calculations are stopped at a given point.

Besides the iterative algorithms, a wide-range of other types of computing methods/algorithms can be applied in anytime systems, as well, if a modular architecture (Fig. 1.) is used [13]. The applicability of this technique is more general with a burden of lower flexibility and it needs extra planning and considerations.

Using this latter solution, each module of the system offers several implementations (characterized by different attribute-values) for a certain task. The units within a given module have uniform interface (same set of inputs, outputs, and solve the same problem) but differ in their computational need and accuracy. At a given time, in the knowledge of the temporal conditions (tasks to complete, achievable time/resources, needed accuracy, etc.) an expert system can choose the adequate configuration, i.e. the units, which will be used. This means the optimization of the whole system instead of individual modules, i.e. in some cases it can be more advantageous to reduce the computational complexity and

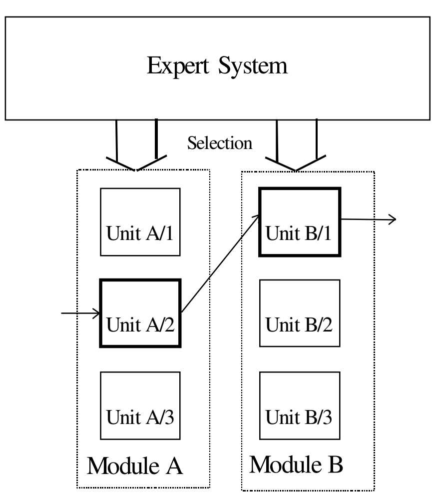

Fig. 1. Anytime system with modular architecture

accuracy of some parts of the system because the information processing or control actions need results or pre-results within a reduced response time or because it is preferable to rearrange the resources to an other, at the moment more important task.

Anytime processing has great advantages in the field of fault detection and isolation. Besides the possibility of coping with the temporarily available resources and data it offers a way for supporting ‘early’ decision-making concerning the necessary control actions if faults/problems occur in the supervised system. With this not only the expenses of the maintenance can be reduced but in some cases also serious alarm situations can be prevented. In the followings a fuzzy modeling based FDI scheme is proposed which is suitable for the anytime mode of operation, as well.

III. FUZZY MODELING BASED FAULT DETECTION AND ISOLATION

Fuzzy modeling, PDC system analysis, and fuzzy observer design methods [14] have emerged among the design approaches of fault diagnosis and isolation. As an extension of these development in this section an anytime FDI scheme is defined in which certain fuzzy models are constructed to identify the faulty and the fault-free behavior of the system under supervision. Residuals are generated by the fuzzy models of the plant that approximate the non-linear system by means of local linear models. Each local model is a linear approximation of the process in a subspace and the selection of the local model is fuzzy. The SVD or, in higher order cases, TPT applied on the scheme results in a TSK fuzzy model and ensures the Ruspini partition and the ordering of the rules, thus the direct PDC design and the anytime mode of operation can be accomplished.

Assume a given dynamic model:

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

where the nonlinear system matrix

$$\mathbf{S}(\mathbf{p}(t)) = \begin{pmatrix} \mathbf{A}(\mathbf{p}(t)) & \mathbf{B}(\mathbf{p}(t)) \\ \mathbf{C}(\mathbf{p}(t)) & \mathbf{D}(\mathbf{p}(t)) \end{pmatrix} \tag{2}$$

is element of a real valued tensor vector space, vectors $\mathbf{x}(t)$, $\mathbf{u}(t)$, and $\mathbf{y}(t)$ are the state, input, and output vectors, respectively and where $s\mathbf{x}(t)$ is the derivative of $\mathbf{x}(t)$ for a continuos-time system or $s\mathbf{x}(t)=\mathbf{x}(t+1)$ for a discrete-time system; further $\mathbf{p}(t)$ is time varying and is bounded by a real valued N-dimensional space Ω. $\mathbf{P}(t)$ may, e.g. include some elements of $\mathbf{x}(t)$ or $\mathbf{u}(t)$.

The TSK model describes model (1) by linguistic rules, such as:

IF $w_{1,i1}(p_1)$ **AND** $w_{2,i2}(p_2)$ **AND** … **AND** $w_{N,iN}(p_N)$ **THEN** $\mathbf{S}_{i1,i2,\dots,iN}$, (3)

whose explicit form via product-sum-gravity inference operator is:

$$\begin{pmatrix} s\mathbf{x}(t) \\ \mathbf{y}(t) \end{pmatrix} = \left(\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_N=1}^{I_N} \prod_{n=1}^{N} w_{n,i_n}(p_n(t)) \mathbf{S}_{i_1,i_2,\dots i_N} \right) \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix} \qquad (4)$$

$w_{n,i}(p_n)$ denotes the ith membership function of the nth variable. $\mathbf{S}_{i1,i2,\dots,iN}$ are the system matrices of the consequent systems. (4) can also be expressed in a more compressed tensor-product form:

$$\begin{pmatrix} s\mathbf{x}(t) \\ \mathbf{y}(t) \end{pmatrix} = S \bigotimes_{n=1}^{N} \mathbf{w}_n(p_n(t)) \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix} \qquad (5)$$

(For tensor operations, see [15]). Here the row vector $\mathbf{w}_n(p_n)$ contains the antecedent membership functions $w_{n,in}(p_n)$, the N+2-dimensional coefficient tensor S is constructed from the consequent system matrices $\mathbf{S}_{i1,i2,\dots,iN}$. The first N dimensions of S assigned to the dimensions of $\mathbf{p}(t)$. An important characteristic of the antecedent fuzzy sets is that they are in Ruspini-partition:

$$\begin{aligned} &\forall n, i, p_n(t) : w_{n,i}(p_n(t)) \in [0,1]; \\ &\forall n, i, p_n(t) : \sum_{i=1}^{I_n} w_{n,i}(p_n(t)) = 1. \end{aligned} \qquad (6)$$

When the antecedent fuzzy sets fulfill (6) the consequent systems define a fixed polytope, where the system varies in: $\mathbf{S}(\mathbf{p}) \in \{\mathbf{S}_1,\mathbf{S}_2,\dots,\mathbf{S}_R\}$. Matrices $\mathbf{S}_r$ are also termed vertex systems. Actually, the polytope defines the convex hull of the system of vertices:

$$\mathbf{S}(\mathbf{p}(t)) = co\{\mathbf{S}_1,\mathbf{S}_2,\dots,\mathbf{S}_R\}_{\mathbf{w}(\mathbf{p}(t))}.$$

A state-space dynamic model given over a bounded domain Ω can always be transformed to TSK model form (5) by TPT. The only problem with this is that TSK models are universal approximators only in theory, which can not really be exploited in practice because of their exponentially increasing complexity growing with the number of parameters. This means that if the number of the local units is bounded then the built model will only be an approximation of the original system. Thus, we have to find a balance between the computational complexity and the accuracy. On the other hand, after ensuring a needed or given accuracy, which may mean the application of a huge number of local models, the computational complexity can be reduced by applying some kind of exact or non-exact complexity reduction method like SVD (HOSVD).

SVD based complexity reduction is a good candidate for applying in anytime systems. First of all, it ensures that the resulted system will be minimal or quasi minimal from computational complexity point of view. Another advantageous aspect of SVD is that it directly gives the significance of the model parts (fuzzy rules), i.e. in which extent they contribute to the output. If we apply SVD (HOSVD) to TSK models, then the SVD or TP Transformed model can be computed via evaluating the fired rules one by one in decreasing contribution order, thus the result will be the best (or quasi-best) approximation in the given circumstances. TPT assigns a weighting value to each antecedent set. In the case of "anytime" computation first those antecedent and rules are evaluated which are assigned to larger weighting values. When the computation is interrupted we can bound the error by the sum of those weighting values which are assigned to the unevaluated rules. The more processing time we have the more precise result we obtain.

Applying the above model a direct link can be established to PDC design, as well. PDC analysis and design techniques suppose the Ruspini partitions of the antecedent fuzzy sets. The benefit of Ruspini option of TPT is that the PDC design can immediately be executed on the resulting fuzzy rule base. TPT can find exact representation. This property may play crucial role in PDC design when exact representation (or very small error) of the model is needed.

IV. ANYTIME FAULT DIAGNOSIS AND CONTROL OF A SUGAR FACTOR PLANT

In the followings the above ideas are investigated in connection with the industrial sugar factory process developed in the framework of the EC-funded Research Training Network *DAMADICS*. (For the detailed system description, benchmark studies, and real measured training data sets of the plant see the homepage of *DAMADICS*, [16], and [17].) The sugar factor plant producing fifty thousand tons of sugar annually, consists of a large number of evaporation plants, boiler houses, heaters and valves. For the production of sugar, beets are collected from the planters. Raw syrup is obtained from thin sliced beets using extractors. After cleaning, decalcifying and processing the syrup, the syrup containing 14% of sugar is condensed to 70% solution using the evaporation station. The waste steam from the steam turbine is used as a heat source for the complete process. A mixture of granulated sugar and syrup is obtained after crystallization process in syrup boilers. Main by-products of this process are the beet-pulp, used for feeding cattle and the remaining syrup used for manufacturing alcohol and organic acids. An automatic control system is used to operate and monitor the system. The whole process is optimized to consume minimal heat. Steam is generated by the boiler house, which operates longer than the beetroot campaign and comprises four technological complexes. Each complex consists of a coaling grate furnace having two grates, barrel boiler and initial steam super-heater. The thickness of the coal coating is controlled manually by a boiler operator. The

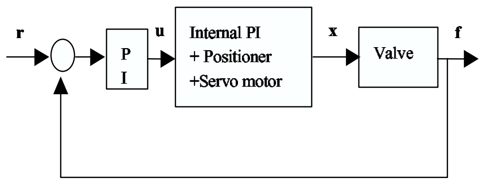

Fig. 2. Main parts of the valve assembly

steam pressure in the barrel boiler determines the feed of each grate and the air inflate through coal in each grate.

The local PI controller loop that controls the syrup levels in evaporator is throttling the syrup flow by acting on control valve, see later. The set point syrup level in the evaporator equals 500mm and should be controlled within the limits of a few percent. The alarms are set when the syrup level is lower than 250mm or higher than 750mm. Both situations are dangerous for the process and process operators. In the case of too low syrup level in the evaporator, the danger of exploding the boiler due to its overheating occurs. When the syrup level is to high, the sugar particles are throw out into the vapours leaving the evaporator, thus polluting further parts of the installation. This leads to process shut down and the necessity of running time wasting and energy consuming cleaning process. The flow rates are of the order of 360m^3/h. From knowledge of the syrup volume in the evaporator one can easily deduce that a dangerous situation could be detected by alarm analysis within 30s, giving further 30s for operator intervention. Shortening the analysis time may be the crucial point in this case. This is principally the task of advanced on- line FDI. The application of new diagnostic algorithms in smart final control elements may bring significant increase to overall system safety, reliability and economy whilst also providing a powerful tool for local performance monitoring.

In the followings we define the FDI scheme of the electro-pneumatic valve actuator discussed above as a crucial part of the process. The anytime operation is applied to this FDI scheme. The details of this FDI task can be found in [18].

The whole valve assembly consists of 3 main parts, see Fig. 2. The first is a PI block, which controls the positioner and actuator output to regulate the flow through the valve. Pneumatic pressure is applied to the servomotor diaphragm to control the stem position that changes the flow. The positioner adjusts this pressure input to the servomotor to obtain the correct stem position of the actuator, see Fig. 3. The valve is the final element in the assembly that alters the flow according to the valve stem position (Fig.2.).

The following faults are detected and isolated:

- Control Valve faults: F1: Valve clogging F2: Valve plug or valve seat sedimentation F3: Valve plug or valve seat erosion F4: Internal leakage (valve tightness) F5: Medium evaporation or critical flow

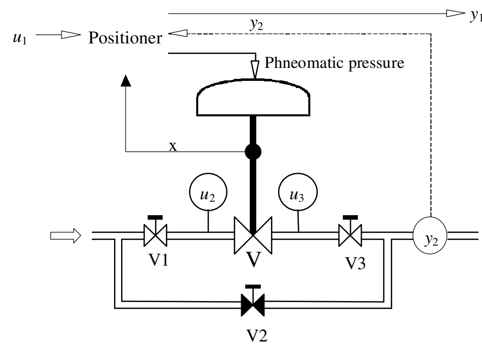

Fig. 3. The servomotor and valve assembly

- Positioner fault: F6: Rod displacement sensor fault
- General faults / external faults: F7: Unexpected pressure drop across valve F8: Fully or partly opened bypass valves F9: Flow rate sensor fault

The fault free and faulty models are fuzzy models with 4-input 2-output structure. The inputs of the model are: the control value (u_1), inlet pressure (u_2), outlet pressure (u_3) and temperature (u_4), while the outputs are: the stem displacement of electro-pneumatic servo motor (y_1) and 1-liquid flow through the valve (y_2).

To reach enough high accuracy ($O(10^{-3})$) a redundant TSK fuzzy model with 30 rules for the control value was identified in the different fault free and faulty regions (Fig.4.). After the exact complexity reduction it led to a minimum system where the number of independent system matrices equals to 11. As the next step the model was transformed to the TP form, showing the significance of the rules, to support the anytime evaluation of the model and the pre-determination of the faults according to the temporary circumstances. The membership functions of the resulting 11 antecedent fuzzy sets are shown in Fig. 5., while Table 1 summarizes the nonzero singular values in decreasing order, thus giving an upper error limit for each rule in case their are discarded. In Fig. 6. the approximation trade-off diagram between the number of evaluated rules and the relative mean square error of output y_1 caused by the truncation of the model can be followed. The analysis of this diagram shows that the anytime operation may reduce significantly the time needed for FDI and for the decision making in the PDC.

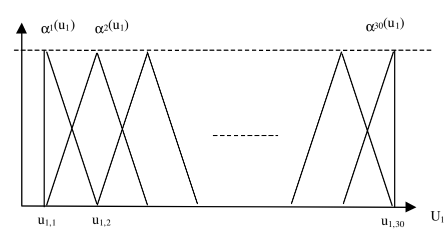

Fig. 4. Antecedent membership functions for bi-linear approximation

Singular values
5.81632874293360
3.98312848649596
3.64441899409118
3.28234697990228
2.78249269506906
2.57396931410666
1.77498807928005
1.68695868758962
1.60678955651332
1.26608566352683
1.08965455037928

Table 1. Singular values

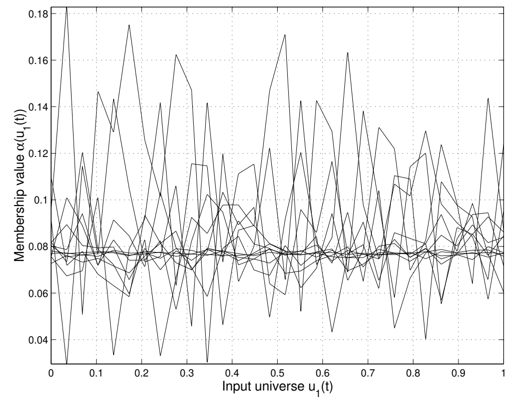

Fig. 5. Membership functions of the resulting 11 antecedent fuzzy sets

CONCLUSIONS

In real applications fault detection and diagnosis, system control, etc. should be carried out "on-line". The amount of available time and resources may depend on the actual circumstances, possibly leading to critical situations. Anytime information processing can help in preventing serious alarm problems. In this paper an anytime fuzzy modeling approach is introduced for FDI schemes and is applied to a real sugar factor plant.

ACKNOWLEDGEMENT

This work was sponsored by the Hungarian Fund for Scientific Research (OTKA T 035190).

REFERENCES

[1] S.A. Billings, "Identification of Nonlinear Systems – A Survey," *IEE Proc.* Vol. 127, Pt.D, No. 6, Nov. 1980, pp. 272-284.

[2] G.J. Klir, T.A. Folger, *Fuzzy Sets, Uncertainty, and Information*, Prentice-Hall International, Inc., 1988.

[3] R. Babuska, J. Sousa, H.B. Verbruggen, "Model-Based Design of Fuzzy Control Systems," *Proc. of the Third European Congress on Intelligent Techniques and Soft Computing EUFIT'95*, Aachen, Germany, pp. 837-841.

[4] Várkonyi-Kóczy, A.R., T. Kovácsházy, "Anytime Algorithms in Embedded Signal Processing Systems," *Proc. of the IX. European Signal Processing Conference, EUSIPCO-98*, Rhodes, Greece, Sep. 8-11, 1998, Vol. 1, pp. 169-172.

[5] Baron,C., J.-C Geffroy, G. Motet ed., *Embedded System Applications*, Kluwer Academic Publishers, 1997.

[6] Takagi, T., M. Sugeno, "Fuzzy identification of system and its application to modelling and control," *IEEE Trans. Systems, Man, and Cybernetics,* Vol. 15, 1975.

[7] Sugeno, M., G. Kang, "Structure identification of fuzzy model," *Fuzzy Sets and Systems,* Vol. 26, No. 1, pp. 15-33, 1988.

[8] Y. Yam, "Fuzzy Approximation via Grid Sampling and Singular Value Decomposition," *IEEE Trans. Systems, Man, and Cybernetics*, Vol. 27, No. 6, 1997, pp. 933-951.

[9] Patton, R.J., R. Clark, P.M. Frank, *Issues of Fault Diagnosis for Dynamic Systems*, Springer Verlag, 2000.

[10] S. Zilberstein, "Using Anytime Algorithms in Intelligent systems," *AI Magazine*, 17(3), 1996, pp. 73-83.

[11] Várkonyi-Kóczy, A.R., T. Kovácsházy, O. Takács, Cs. Benedecsik, "Anytime Algorithms in Intelligent Measurement and Control." *Proc. of the 2000 World Automation Congress*, WAC 2000, Maui, USA, June 11-16, 2000.

[12] Takács, O., A.R. Várkonyi-Kóczy, "Iterative-type Evaluation of PSGS Fuzzy Systems for Anytime Use," *Proc. of the 2002 IEEE Instrumentation and Measurement Technology Conference, IMTC/2002*, Anchorage, USA, May 21-23, 2002, pp. 233-238.

[13] Várkonyi-Kóczy, A.R., A. Ruano, P. Baranyi, O. Takács, "Anytime Information Processing Based on Fuzzy and Neural Network Models," *Proc. of the 2001 IEEE Instrumentation and Measurement Technology Conference, IMTC/2001*, Budapest, Hungary, May 21-23, 2001, pp. 1247-1252.

[14] Tanaka, K., Wang, H.O, *Fuzzy Control Systems Design and Analysis*, John Wiley & Sons, Inc. New York, 2001.

[15] Lathauwer, L.D., B.D. Moor, J. Vanderwalle, "A multi linear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, Vol. 21, No. 4, 2000, pp. 1253-1278.

[16] Waseiwicz, P., "Description of Sugar Technology Process," *Internal Report at the Institute of Automatic Control and Robotics*, Warsaw University of Technology, 1998. (The report is available from the authors of this paper.)

[17] Bartys, M., J. Caldo, R.J. Patton, J. Quevedo, "Fault diagnosis of industrial actuators in a sugar factory: a european study, special feature on intelligent instruments." *IEE Computing and Control Journal*, Vol. 13, No. 5, pp. 247-253, 2002.

[18] Uppal, F.J., R.J. Patton, V. Palade, "Neuro fuzzy based fault diagnosis applied to an electro-pneumatic valve," *15th Triennial world Congress of the IFAC,*2002.

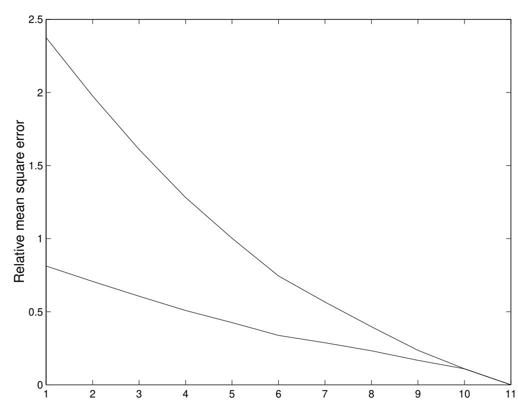

Fig. 7. Approximation trade-off diagram