SVD Based Anytime Modeling and Control of Nonlinear Systems[1]

Annamária R. Várkonyi-Kóczy*,+, Orsolya Takács*,+, Péter Baranyi,+**

* Dept. of Measurement and Information Systems
Budapest University of Technology and Economics
Magyar Tudósok krt 2., H-1521 Budapest, Hungary
Phone:+36 1 4632057, Fax: +36 1 4634112
koczy@mit.bme.hu, takacs@mit.bme.hu

** Dept. of Telecommunication and Telematics
Budapest University of Technology and Economics
Magyar Tudósok krt 2., H-1521 Budapest, Hungary
baranyi@alpha.ttt.bme.hu

\+ Integrated Intelligent Systems Japanese-Hungarian Laboratory

Abstract: Nowadays practical solutions of engineering problems involve model-integrated computing. Model based approaches offer a very challenging way to integrate a priori knowledge into the procedure. Fuzzy and neural network based models can also successfully be applied in cases of highly nonlinear problems, where only partial, uncertain and/or inaccurate data is available. Although, even their use is limited by the exponentially increasing computational complexity. In this paper the applicability of (Higher Order) Singular Value Decomposition based Soft Computational models is analyzed with an extension towards anytime usage.

Keywords: anytime systems, fuzzy modeling, fuzzy control, singular value decomposition, higher-order tensor decomposition, complexity reduction

1 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 measuring and control, etc. devices. Thus, model based approaches offer a very challenging way to integrate a priori knowledge into the procedure.

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

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 now practically no systematic method was available which could be offered to solve a larger family of nonlinear engineering problems. The efforts on the fields of fuzzy and neural network (NN) modeling and control 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.

Singular Value Decomposition (SVD) technique has successfully been used to reduce the complexity of a large family of systems based on both classical and soft techniques [4]. 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) can be applied.

SVD is serious candidate to overcome the complexity problems arising in modeling of complex systems where we have an analytical description of the system, which is too complicated to be handled or if we have a system represented by input-output sample pairs. In these cases we can build a model approximating the system using local (linear) models. Such techniques include Takagi-Sugeno (TS) fuzzy model approximation [5] or polytopic model approximation (PMA). These methods have theoretically a universal approximation property, however, it can not be really exploited because they have an 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.

A further reason for dealing with so called soft computational model based techniques is that 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 [6]. The concept of "anytime" processing tries to handle the case of too many abrupt changes and their consequences in larger scale embedded systems [7]. 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 novelty of the paper is that it proposes a uniform frame for a family of modeling methods based on soft computing techniques together with an anytime extension. This results in low (optimal or nearly optimal) computational complexity, easy realization, robustness, a possibility to maintain continuous operation, and to cope with the limits arising in the system or in its environment. Furthermore, the accuracy can also easily and flexibly increased and we do not need any a priori or expert knowledge about the system to be modeled. It can be used either when a mathematical description (possibly too complex to be handled) or when only (possibly partial, inaccurate, and uncertain) measurement data is available.

The paper is organized as follows: in the first section the generalized idea of anytime processing is introduced, while in the second the basics of singular value decomposition are summarized. Section three is devoted to the complexity reduction of models based on soft computing techniques. In the next section Takagi-Sugeno fuzzy model approximation is briefly discussed which is followed by two sections with the ideas of anytime modeling and anytime control.

2 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 is used and the evaluation of the 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 [8] 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 [9].
The algorithms/computing methods, which are suitable for anytime usage have to fulfill the following expectations:

- 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.
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 [10]. The applicability of this technique is more general with a burden of lower flexibility and it needs extra planning and considerations.

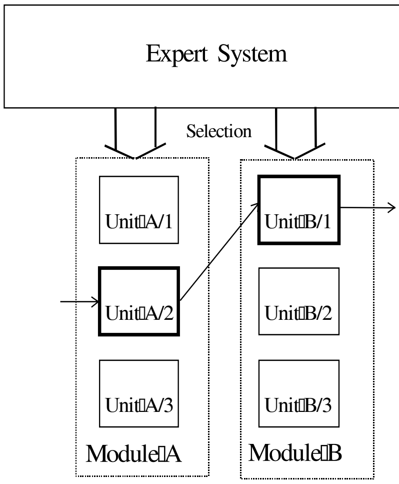

Fig. 1. Anytime system with modular architecture

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, output, 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 accuracy of some parts of the systems and rearrange the resources to an other, at the moment more important task.
Although, the units implementing a certain task may have different internal structure, from several point of view (see e.g. [10]) it is advantageous if they are built of similar structure. In this case the adaptation or change between the units means only the change of some parameter set.

3 Singular Value Decomposition

The SVD-based complexity reduction algorithm is based on the decomposition of the F matrix:

$$F_{(n_1 \times n_2)} = A_{1,(n_1 \times n_1)} B_{(n_1 \times n_2)} A^T_{2,(n_1 \times n_2)} \qquad (1)$$

where A_k are orthogonal matrixes ($A_k A_k^T = E$), and B contains the λ_i singular values of F in decreasing order. The maximum number of the nonzero singular values is $n_{SVD} = \min(n_1, n_2)$. The singular values indicate the significance of the corresponding columns of A_k. Let the matrices be partitioned in the following way:

$A_k = \left| A^r_{k,(n_k \times n_r)} \quad A^d_{k,(n_k \times n_d)} \right|$ and $B_k = \left| \begin{matrix} B^r_{(n_r \times n_r)} & 0 \\ 0 & B^d_{((n_1 - n_r)\times(n_2 - n_r))} \end{matrix} \right|$, where r denotes "reduced" and $n_r \le n_{SVD}$.

If B^d contains only zero singular values then B^d and A_k^d can be dropped: $F = A_1^r B^r A_2^{rT}$. If B^d contains nonzero singular values, as well, then the $F' = A_1^r B^r A_2^{rT}$ matrix is only an approximation of F and the maximal difference between the values of F and F' [11]:

$$E_{RSVD} = |F - F'| \le \left(\sum_{i=n_r+1}^{n_{SVD}} \lambda_i\right) 1_{(n_1 \times n_2)} \qquad (2)$$

For higher order cases Higher Order SVD (HOSVD) can be applied in a similar way (see e.g. [12]).

Here we have to remark that if SVD is applied to a two dimensional matrix then it can be proved that the resulting matrix of lower rank will be the best approximation of the original matrix in least-squares sense (i.e. the reduction is "optimal"). Although, if HOSVD is applied then the minimum property does not hold anymore. We can state only that the "significant" singular values will have the lower indices, however in most cases if there is a considerable difference among the singular values it results in an approximation which is very near to the possible best one.

4 Exact and Non-Exact Complexity Reduction of Neural Network and Fuzzy Models Based on SVD

SVD based complexity reduction can be applied to various types of soft computational systems. Here, as examples the complexity reduction of the product-sum-gravity fuzzy systems with singleton consequents (PSGS) and the reduction of generalized neural networks (GNN) will be presented.

4.1 Reduction of fuzzy rule-bases with SVD

Consider a fuzzy rule base with two inputs, where the antecedent fuzzy sets are in Ruspini-partition and the consequence fuzzy sets are singletons. So the rules are:

$R_{i,j}$: If x_1 is $A_{1,i}$ and x_2 is $A_{2,j}$ then $y = y_{i,j}$, where $i = 1 \ldots n_1$ and $j = 1 \ldots n_2$.

The fuzzyfication method is singleton and during the inference, product T-norm and sum S-norm are used. The result of the fuzzy inference in case of the (x_1^*, x_2^*) input values will be:

$$y^* = \sum_{i_1, i_2} y_{i_1, i_2} \mu_{A_{1,i_1}}(x_1^*) \mu_{A_{2,i_2}}(x_2^*) . \qquad (3)$$

Let F be a matrix, containing the $y_{i,j}$ elements, then apply the above mentioned procedures to obtain $F \approx F' = A_1 B A_2^T$, where A_1 and A_2 are SN (Sum-Normalized: the sum of each row equals to one) and NN (Non-Negative). Then the new rule-base will be:

$R'_{i,j}$: If x_1 is $A'_{1,i}$ and x_2 is $A'_{2,j}$ then $y = y'_{i,j}$,

where $i = 1 \ldots n_1^r$, $j = 1 \ldots n_2^r$, $y'_{i,j}$ are the elements of B, and the new membership functions can be obtained as:

$$\mu_{A'_{k,i}}(x_k) = \sum_j \mu_{A_{k,j}}(x_k) A_{k,j,i} , \qquad (4)$$

where $A_{k,j,i}$ is the (j,i)th element of A_k .

The reduced rule-base contains only n_r^2 rules instead of $n_1 * n_2$ rules and the error can be estimated from the discarded singular values.
The method can be extended to n-dimension cases by applying HOSVD.
PSGS fuzzy systems can be evaluated iterative-type in anytime systems. The proof and the necessary transformation is described in [17].

4.2 Reduction of neural networks with SVD

The classical multi-layer neural network can be generalized, if the non-linear transfer functions are moved from the nodes into the links. It results in neurons that apply only a sum operation to the input values, and links that are characterized by possibly non-linear weighting functions, instead of simple constant weights.

Let us focus on two neighbouring layers l and $l+1$ of a forward model. Let the neurons be denoted as $N_{l,i}$, $i=1..n_l$ in layer l, where n_l is the number of neurons. Further, let input values of $N_{l,i}$ be $x_{l,i,k}$, $k=1..n_{l-1}$ and its output $y_{l,i}$. The connection between layers l and $l+1$ can be defined by the $f_{l,j,i}(y_{l,i})$ weighting functions ($j=1..n_{l+1}$). Thus

$$x_{l+1,j,i} = f_{l,j,i}(y_{l,i}) \qquad (5)$$

Therefore, the output of neuron $N_{l+1,j}$ will be

$$y_{l+1,j} = \sum_{i=1}^{n_l} f_{l,j,i}(y_{l,i}) \, . \qquad (6)$$

The weighting functions can also be changed during the training: the unknown weighting functions are approximated with linearly combined known functions, where only the linear combination must be trained. For this approximation the above described PSGS fuzzy systems can be used, with one input and one output:

$$y_{l+1,j} = \sum_{i=1}^{n_l} f_{l,j,i}(y_{l,i}) = \sum_{i=1}^{n_l} \sum_{t=1}^{m_{l,i}} \mu_{l,i,t}(y_{l,i}) b_{l,j,i,t} \qquad (7)$$

To reduce the size of a generalised neural network the SVD-based complexity reduction can be used. Eq. (7) can always be transformed into the following form:

$$y_{l+1,j} = \sum_{z=1}^{n_{l+1}^r} a_{l,j,z} \sum_{i=1}^{n_l} \sum_{t=1}^{m_{l,i}^r} \mu_{l,i,t}^r(y_{l,i}) b_{l,z,i,t}^r \qquad (8)$$

where "r" denotes "reduced", further $n_{l+1}^r \le n_{l+1}$ and $\forall i : m_{l,i}^r \le m_{l,i}$.

The reduced form is represented as a neural network with an extra inner layer between layers l and $l+1$. Between the original layer l and the new layer the weighting functions are approximated from the reduced PSGS fuzzy systems, and layer $l+1$ simply computes the weighted sum ($a_{l,j,z}$) of the outputs of the new layer.

The reduction means the reduction of the $B = \left[b_{l,j,i,t}\right]$ three-dimensional matrix by using HOSVD. The maximal error of the resulting neural network can be computed from the discarded singular values [13], [14].

5 Takagi-Sugeno Fuzzy Model Approximation

Takagi-Sugeno (TS) fuzzy modeling is a technique to describe a nonlinear dynamic system using local linearized models [5]. The idea is that the system dynamics is captured by a set of fuzzy implications, which characterize local regions in the state space. The overall fuzzy model i.e. the description of the whole system is achieved by the convex combination - fuzzy blending – of the linear models. The combination is usually defined by an array of local models and the tensor product of basis functions, which express the local dominance of the local models. Using the TS fuzzy model approximation, the controller design and Lyapunov stability analysis reduces to solving the Linear Matrix Inequalities (LMIs) problem.

TS fuzzy modeling technique can be used both if we have an analytical description of the system, i.e. the system is given e.g. by differential equations or if only input-output samples are given, thus we make a black-box modeling. In the first case we are to sample the system over a rectangular hyper-grid, which leads to a similar problem as black-box modeling, accept that in this case the samples, i.e. the approximation points can directly serve as linear models, while in the latter case we have to evaluate a Lagrange interpolation to adapt the local models to force the overall model to copy the behavior of the system in the sampling points.

TS fuzzy models are theoretically universal approximators. Despite this advantage their use is practically limited, because their computational complexity grows exponentially with the number of parameters and the universal approximation property doesn't hold if the number of antecedent sets are limited [15]. Thus, if we want to get a good approximation we usually have to apply a high number of antecedent fuzzy sets with a burden of possibly unmanageable complexity. Consequently, methods helping to find the minimum number of necessary building units to a given accuracy are highly desirable.

SVD (or HOSVD) is an excellent tool for the above purpose. It can not only be used to filter out the redundancy of a system. The singular values can be applied for the decomposition of the system, as well and they indicate the degree of the significance of the decomposed parts, i.e. in which extent they contribute to the output of the overall system. Thus, if it is necessary or advantageous we can apply a further non-exact reduction to decrease the computational complexity by truncating the model, i.e. by discarding the weakly contributing parts to find the trade off between computational complexity and accuracy.

6 Anytime Modeling: Complexity Reduction and Improving the Approximation

With the help of the SVD-based reduction not only the redundancy of the rule-bases (or neural nets) can be removed, but further reduction can also be obtained, considering the allowable error. This latter can be down adaptively according to the temporal conditions, thus offering a way to use soft computational, i.e. fuzzy and generalized neural network (NN) based models in anytime systems.

The method also offers a way for improving the model if later on we get into possession of new information (approximation points) or more resources. An algorithm can be suggested which finds the common minimal implementation space of the compressed original and the new approximation points, thus the complexity will not exploit as we include new information into the model (for more details see [16]). These two techniques, nonexact complexity reduction and the improvement of the approximation accuracy ensure that we can always cope with the temporarily available (finite) time/resources thus find the balance between accuracy and complexity.

The steps of using anytime models are the followings: First a practically "accurate" fuzzy or NN system is to be constructed. For the determination of the rule-base expert knowledge can be used. Further improvement can be obtained by the use of training data and some learning algorithm.

In the second step applying the SVD-based complexity reduction algorithm a reduced but accurate model can be generated. To make possible the use of the model in anytime systems, either the iterative algorithm described in [17] can be followed or further variations of the rule-base of the model must be constructed, with different accuracy and complexity. An alternative rule-base can be characterized by its complexity and its error that can be estimated by the sum of the discarded singular values.

In this latter case the different rule-bases form the different units realizing a given module (Fig. 1.). The expert system, monitoring the actual state of the supervised system, can adaptively change the units - rule-bases - according to the available computing time and resources at the moment. Because the inference algorithm is the same, only the rule-bases - a kind of parameter set - must be changed. It is worth mentioning, that the SVD-based reduction finds the optimum, i.e., minimum number of parameters which is needed to describe the system.

7 Anytime Control Using SVD Based Models

Recently, the popularity of fuzzy control has grown rapidly. There are numerous successful applications of fuzzy control which affect on the analysis and design of fuzzy control systems. The previously discussed ideas and tools can fruitfully be combined if a certain type of fuzzy model based control, namely TS fuzzy modeling and for the controller design Parallel Distributed Compensation (PDC) [18] is used. If the model approximation is given in the form of TS fuzzy model, the controller design and Lyapunov stability analysis of the nonlinear system reduce to solving the LMI problem. This means that first of all we need a Tagaki-Sugeno model of the nonlinear system to be controlled. The construction of this model is of key importance. This can be carried out either by identification starting from input-output data pairs or we can derive the model from given analytical system equations.

The PDC offers a technique to design a fuzzy controller from the TS fuzzy model. This procedure means that to each local model a local controller is determined. This implies that the more complex the system model is the more complex controller will be obtained. According to the complexity problems outlined in the previous sections we can conclude that when the approximation error of the model tends to zero the complexity of the controller explodes to infinity. This pushes us focus on possible complexity reduction and anytime use.

SVD based complexity reduction can be applied on two levels in the TS fuzzy controller. First, we can reduce the complexity of the local models (local level reduction). Secondly, it is possible to reduce the complexity of the overall controller by neglecting those local controllers, which have negligible or less significant role in the control (model level reduction). Both can be applied in an anytime way, where we take into account the "distance" between the current position and the operating point, as well. The model granularity or the level of the iterative evaluation can cope with this distance: the more far we are the more rough control actions can be tolerated. Although, the approximated models may not guarantee the stability of the original nonlinear system, this can also be ensured by introducing robust control (see e.g [19]).

8 Conclusion

In modern monitoring, diagnostics, measurement, and control systems, the available time and resources are often not only limited, but could change during the operation of the system. In these cases, the so called anytime algorithms can be used advantageously. While different soft computing methods are wide-spread used in system modeling, their usability is limited, because the lack of any universal method for the determination of the needed complexity often results in huge and redundant neural networks/ fuzzy rule-bases. This paper proposes a possible way to carry out anytime information processing and control in fuzzy systems or neural networks, with the help of the (Higher Order) Singular Value Decomposition based complexity reduction algorithm.

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] Y. Yam, "Fuzzy Approximation via Grid Sampling and Singular Value Decomposition," *IEEE Trans. S.M.C.*, Vol. 27, No. 6, 1997, pp. 933-951.
[5] T. Takagi, and M. Sugeno, "Fuzzy Identification of Systems and its Applications to Modeling and Control," *IEEE Trans. S.M.C.*, Vol. 15, 1985, pp. 116-132.
[6] 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.
[7] Baron,C., J.-C Geffroy, G. Motet ed., *Embedded System Applications*, Kluwer Academic Publishers, 1997.
[8] S. Zilberstein, "Using Anytime Algorithms in Intelligent systems," *AI Magazine*, 17(3), 1996, pp. 73-83.
[9] 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, pp..
[10] Várkonyi-Kóczy, A.R., A. Ruano, P. Baranyi, O.Takács, "Anytime Information Processsing 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.
[11] Yam, Y., P. Baranyi, C.T. Yang, "Reduction of Fuzzy Rule Base Via Singular Value Decomposition," *IEEE Trans. on Fuzzy Systems*, Vol. 7., No. 2., Apr., 1999, pp.120-132.
[12] Baranyi, P., A.R. Várkonyi-Kóczy, Y.Yam, P.Michelberger, "HOSVD Based Computational Complexity Reduction of TS Fuzzy Models," *Proc. of the Joint 9th IFSA World Congress and 20th NAFIPS International Conference, IFSA / NAFIPS 2001*, Vancouver, Canada, July 25-28, 2001, pp. 2482-2485.
[13] Takács, O., I. Nagy, "Error-bound of the SVD Based Neural Networks," *Proc. of the IFAC Symp. on Artificial Intelligence in Real-Time Control, AIRTC'2000*, Budapest, Hungary, Oct 2-4, 2000, pp. 139-144.
[14] Takács, O., A.R. Várkonyi-Kóczy, "Error-Bound for the Non-Exact SVD-Based Reduction of the Generalized Type Hybrid Neural Networks with Non-Singleton Consequents," *Proc. of the 2001 IEEE Instrumentation and Measurement Technology Conference, IMTC/2001*, Budapest, Hungary, May 21-23, 2001, pp. 1607-1613.
[15] Tikk, D., "On nowhere denseness of certain fuzzy controllers containing prerestricted number of rules," *Tatra Mountain Math. Publ.*, 16, 1999.
[16] Baranyi, P., A.R. Várkonyi-Kóczy, "Adaptation of SVD Based Fuzzy Reduction via Minimal Expansion," *Proc. of the IEEE Int. Workshop on Intelligent Signal Processing, WISP'2001*, Budapest, Hungary, May 24-25, 2001, pp. 171-176.
[17] 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.
[18] Tanaka, K., Wang, H.O, *Fuzzy Control Systems Design and Analysis*, John Wiley & Sons, Inc. New York, 2001.
[19] Tanaka, K., T. Taniguchi, H.O. Wang, "Robust and Optimal Fuzzy Control: A Linear Matrix Inequality Approach," Proc. Of the 1999 IFAC World Congress, Beijing, July 1999, pp. 213-218.