Viharos, Zs. J.; Automatic generation a net of models for high and low levels of production control, *16th IFAC World Congress*, July 4-8, 2005, Prague, Czech Republic, *accepted paper.*

AUTOMATIC GENERATION A NET OF MODELS FOR HIGH AND LOW LEVELS OF PRODUCTION CONTROL

Zsolt János Viharos

Computer and Automation Research Institute, Hungarian Academy of Sciences, Kende u. 13-17, Budapest, POB 63, H-1518, Hungary, e-mail: zsolt.viharos@sztaki.hu

Abstract: The paper presents two applications of the novel, artificial neural network (ANN) and feature selection based combined, dynamic technique to automatically dissolve a large, complex system into a net of connected submodels. The first application is a solution for the lower level of customised mass-production systems, for increasing their productivity. The second one is a concept for the identification and reorganisation of manufacturing agents, based on simulation experience. The main idea behind, is to give the learning capability to agents already at their definition phase, and to maximise their foresight power.

Keywords: Industrial production systems, Production control, Quality control, Productivity, Machine learning, Neural-network models, Agents

1. INTRODUCTION

The paper presents two applications of the novel, artificial neural network (ANN) and feature selection based combined, dynamic technique to automatically dissolve a large, complex system into a net of connected submodels.

The introduced submodel detection technique assumes that a relative comprehensive database about the analysed system is given serving as the basis of analysis. Different methods provide this condition in the two application areas considered. Large databases collected by process monitoring systems connected to manufacturing systems and production lines are available in the assignment related to control of low-level manufacturing processes (Viharos, *et al*., 2003a). The saved values of the state and communication variables of the simulation model of the analysed production system serve with the database at the high-level of production control.

The algorithm resulting in a net of submodels explores the dependencies inside a given database, as typical in data-mining assignments. The concrete algorithm processes a large database table, having high number of columns and average number of rows. This can be a common table inside the database, but typically inherits from various basic tables through various database preparation steps, typically through a number of table joins. The solution builds up small submodels, so the meaning of the database attributes and the incorporated information content determine the explored structure and the possible way of its industrial applications.

Also the applications of the results are different on the two manufacturing control levels. The identification and definition of production agents are the main targets at the higher level, in order to directly receive a control system with learning capabilities. Dependency exploration and technological improvements are the goals of the application at the lower level.

The rest of introduction deals with various consideration aspects according to the introduced algorithm and with approaches for ANN-based decision support structures. The second part details the applied submodel finding method followed by the description of its applications for high and low levels of production control. Conclusions and references close the paper.

1.1 Various consideration aspects related to the introduced algorithm

This paragraph aims at positioning the introduced submodel finding algorithm. Various aspects can be enumerated when specifying the place of a modelling and model building technique, only a part, considered important, is mentioned.

According to the core modelling technique, the solution is mainly based on *neural networks*. *MultiLayer Perceptron* (MLP) ANN models are used exclusively, mirroring the position of the technique among the wide range of neural network types. The training algorithm is based on an *accelerated back-propagation* called SuperSab (Tollenare, 1990) but *it was modified* several times.

According to the last remark in the previous section, the model building method can be considered as a special learning algorithm, too. It does not require predetermining whether a parameter is on the input or output side of the model for building up, consequently, it can be ordered also into the class of *unsupervised learning algorithms*.

Modelling of many-valued mapping is solved by the introduced algorithm, too. A similar problem is identified and solved excellently by a totally different approach of Brouwer and Pedrycz (2003). By coincidence, in the next step their research turned into the *field of handling incomplete data* as this was the case with the presented algorithm and authors, too (Viharos, *et al*., 2002).

Various approaches can be found in the literature for improving the structure of ANN models also in the case of MLPs, adding and deleting neurons and links are typical steps of this approach. The resulted net of connected neural sub-networks can be considered also as *a special solution of a structure determination process of ANNs*. The proposed algorithm can result similar outcome than a pruning-learning process combination, so *it can be considered as a very special form of pruning solution*.

The applications of ANNs are typically preceded by a feature selection algorithm, especially in the field of manufacturing (Monostori, *et al*., 2000) to surmount their capability restrictions, with respect to the number of parameters and thus, model sizes. It can be found that feature selection and training processes are typically separated (Viharos, *et al*., 2003a). The new, introduced algorithm brakes with this practice; it is a dynamic, integrated combination of these steps. In this aspect, it can be considered as a *special feature selection algorithm*, or also as a *hybrid combination of feature selection and learning based model building*, too.

1.2 Approaches for ANN based decision support structures

This paragraph makes an overview of various solutions to prepare ANN structures containing several, individual and connected submodels. This part is intended to inspire that usually the structure and connections of the applied modes are more predetermined based on different functional approaches than self-structuring.

The first group of these algorithms relates to techniques where the structures of the connected submodels are predetermined before learning (Caelli *et al*., 1999). The paper highlights that the biological neural networks are relatively rare connected inspiring the necessity of distributed modelling solutions and that the assignment solutions of various applications can be quicker having a computer model consisting of net of submodels structure built up usually on the base of field specific know-how. Image recognition is this field in Lu and Szeto, (1993) with different ANNs for contour detection, gradient adjustment and orientation adjustment. The structure and connections among these submodels are predetermined according to the image evaluation process steps. Different applications such as texture classification, face and currency recognition are solved through the same model structure, combining fuzzy and neural techniques showing the possibility of having a common, hierarchical decision support structure for various assignments built up on the base of decision making components (Kung, *et al*., 1999). A two-layered fix structure is presented in the paper of Hu, *et al.*, (2004) about learning the motion trajectories where the ANN structure is predetermined by its working behaviour. One of the most elegant and promising approaches is the hierarchical mixture of experts (Jordan and Jacobs, 1994) which consist of decision making components, too.

The second group of the modelling algorithms in this aspect relates to relatively rare appearing techniques, where the structures of the connected submodels are dynamically determined during the learning. Dynamics is represented in Guan, *et al.*, (1997) through the automatic clustering of the ANNs in the introduced network-of-networks (NoN) model applied in the field of image regularization. A simple restructuring algorithm is presented in Mason and Robertson, (1995) to make the hardware realization of ANNs through the modification of their given hierarchy. High-level dynamics is presented in Basak, (2004) during building up online adaptive decision trees where ANN models are applied in the branch points. All ANNs at all of the decision points receive the whole input data set as inputs resulting in a very promising and adaptive approach.

2. DESCRIPTION OF THE APPLIED SUBMODEL FINDING METHOD

Because of the great variety of manufacturing description parameters, it is very difficult to build up a comprehensive model, e.g. for a production process even if a part of the whole system is modelled. Identifying parts which can be modelled independently is one of the main issues of modelling. A very important goal of research is to automatically determine individual parts like this based on the given parameters and artificial neural network models. The following paragraph describes the algorithm from the user's point of view.

The application of the algorithm has two main prerequisites:

- The user has to serve with a set of data describing the analysed system. This can be satisfied typically with a database table where columns are the description variables and the rows contain their values belonging together. Various settings of these features allow different analysis of the same system.
- A further prerequisite of the application is the setting of allowed, excepted errors or required estimation accuracy for all of the system variables. This requirement is inherited from the ANN based learning technique, it has to be defined when to stop a learning process. Implicitly, the user defines by what level of estimation accuracy, or allowed error can be stated, that a parameter can be estimated based on other ones. This setting can be different for the individual system parameters but it has to be determined before the algorithmic run, consequently, in some respect it is an advantage, but in other respect it is a disadvantage of the solution. Repeated run with various accuracy requirements can mirror the variety of solutions in respect to this prerequisite. It is worth mentioning also that based on some ideas, of the authors this is one of the main domains for further improvement of the method.

Satisfying the above requirement allows to run the developed algorithm. Its result can be grouped into three main parts:

- Net of accepted submodels. They can perform the estimation of their output parameters with the prescribed, individual accuracy. They can have common parameters, so the result is a net of neural networks (similar to NoN, above).
- List of rejected submodels (Not highlighted in Figure 1.). These models were analysed during the search but they were rejected. The basics of this evaluation method is described in (Monostori, *et al.*, 2000) with the extension that a model is accepted if at least one of their parameters can be estimated with the prescribed accuracy, based on the remaining ones.
- Because models are identified through their building up process, the algorithm results also in the concrete neural network models for each of the accepted submodels. This allows the prompt application of the whole, or a part of the net of submodels for solving various assignments. A sample technique of this solution is detailed in (Monostori, *et al.*, 2000).

The algorithm can be applied also when the data set incorporates also incomplete information (Viharos, *et al.*, 2002).

Figure 1. shows an example of a resulted net of accepted submodels having five main parts (in brackets), dividing a system containing eleven (indexed from zero to ten) description parameters. The fourth row of the demonstrated software window shows that the algorithm identified a submodel where parameters no. 2, 3 and 6 as model inputs are able to estimate the variable no. 5. The four identified submodels have common parameters, e.g. parameter no. 6 is estimated by the submodel showed in the second row, but it is to be found among the input variables of the next two submodels, too, showing that this technique recognises a structure of connected submodels, over the identification of its individual parts.

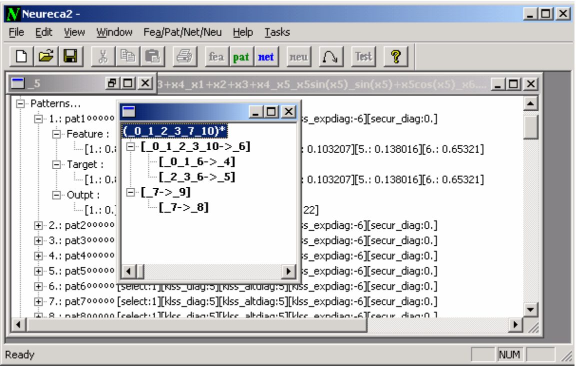

Fig.1.The resulted submodel structure of a complex system

3. APPLICATIONS OF THE ALGORITHM FOR HIGH AND LOW LEVELS OF PRODUCTION CONTROL

3.1 Application of the method for low level of production control

Manufacturing systems in our epoch work in a fast changing environment full of uncertainties. Increasing complexity is another characteristic which shows up in production processes and systems and in enterprise structures as well (Merchant, 1998).

A Hungarian R&D project, called Digital Factory was started to make all the important, production-related information available and manageable in a controlled, user-dependent way by the efficient application of information and communication technologies (Monostori, *et al.*, 2002).

The project covers the following - partly overlapping - main directions to be treated in a comprehensive way:

- Management and scheduling of large-scale projects (Kovács, *et al.*, 2003).
- Tele-presence and interactive multimedia (Haidegger and Popa, 2002).
- Monitoring of complex production structures (Viharos, *et al.*, 2003b)

The paper considers a small part of aspects, steps and results of the third cluster, called “Monitoring of complex production systems” of the above introduced project. All of the clusters incorporate three main work areas representing the continuous development starting with basic and applied research, followed by research and development (R&D) assignments, and ending in the market-oriented demonstrations of the cluster results.

A part of the R&D work tried to build up production models where dependencies among parameters are unknown. The modelling was based on learning form collected manufacturing data sets. No measurements were needed in this case, because there was a huge number of related monitoring parameter. These data are stored in big databases incorporating the high value of information on the experience through production supervision collected in course of several years. Engineers' opinion was quite interesting at the beginning: "there should be some connections among these parameters".

The new submodel finding technique, illustrated earlier, was applied beyond the ANN-based model building with predetermined input-output parameters, for analyzing the dependencies among data, driven by some not satisfactory analysis.

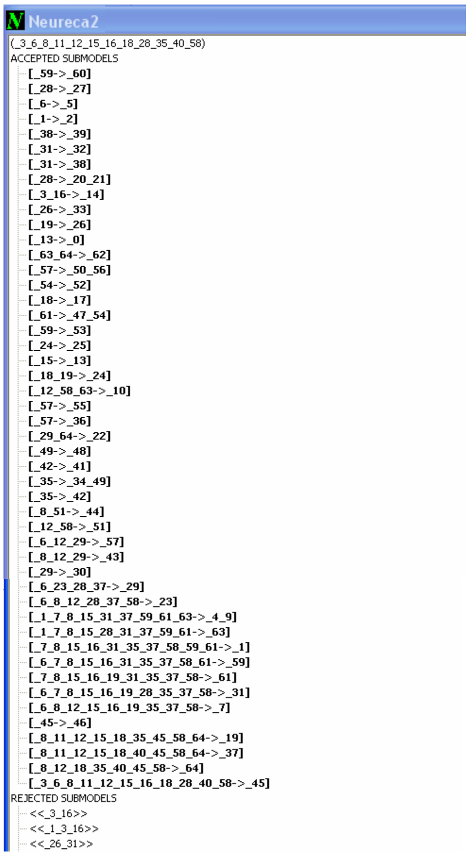

Fig. 2. The explored complexity of dependencies among different parameters (represented as numbers) of machines inside a production line with 8.8% of expected estimation accuracy.

Figure 2. and 3. show an interesting result representing the complexity of dependencies in our production equipment. Sixty-five parameters (represented as numbers) were used for the description of some machines and processes, consequently, only a part of the whole production line was taken into account, and also only a special production aspect was studied, indicating that a comprehensive analysis is an enormously complex and difficult task in the production line level. The expected level of accuracy is different concerning the Figure 2. and 3., it is +/- 8.8% at the first and +/- 5.6% at the later one, higher level of accepted errors allows more and also smaller submodels as represented in the pictures. It has to be mentioned that only a short examples of the rejected submodels is highlighted in the pictures. The ratio in the number of accepted and rejected submodels is approximately 1:7 in both of the cases.

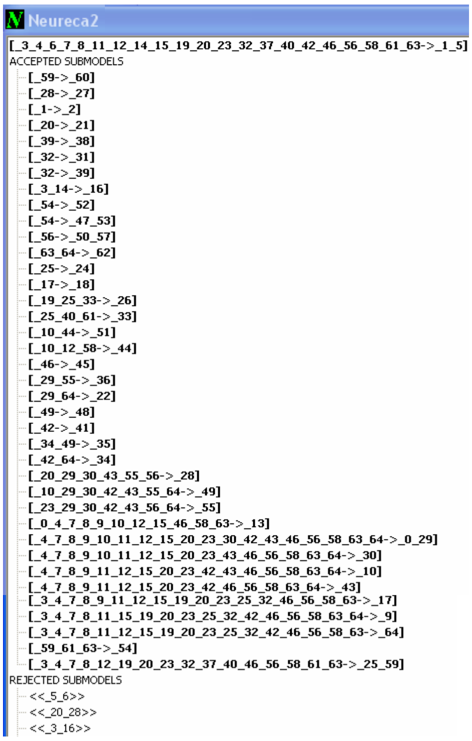

Fig. 3. The explored complexity of dependencies among different parameters (represented as numbers) of machines inside a production line with 5.6% of expected estimation accuracy.

In spite of this difficulty, one of the main future targets of the research and development activities is to extend these modelling and dependency exploration techniques into the comprehensive production line level.

The explored dependencies among manufacturing data can be considered as an important improvement in the data analysis process after two, unsuccessful examinations of the given production steps but with predetermined input-output parameters. The highlighted connections can directly serve as the model basis of this low-level manufacturing assignment and may indicate new ideas for technical improvements. The determination of appropriate expected accuracy levels based on the collected statistical process control (SPC) data is one of the main directions for further developments in application.

3.2 Application of the method for high-level of production control – main motivation

This paragraph details the concept how to apply the above introduced algorithm for high-level of manufacturing control. An agent-based control technique is addressed and agent identification is the target of the solution with a special aspect to directly receive learning agents. It can be seen that agent identification for control production systems is typically solved through field-specific approaches. A well-known example and an excellent solution is the PROSA architecture (Valkenaers, *et al*., 2001) where the identified agents are typical components of manufacturing systems, like product, resource, order and staff. Neither to overemphasise the self-determination of main entities nor to downgrade the otherwise very important professional know-how incorporated in the predetermined structure but the current approach breaks with the field-specific solution it tries to identify agents automatically.

The application of the method for the high-level production control: highlighting analogues between learning agent identification and submodel exploration assignments. The following paragraph highlights the analogues between the submodel and agent identification, as main basis of the concept.

The exploration of separate, small submodels is quite similar to the agent definition tasks, because an agent can be considered as a small part of a larger system. More obvious is the analogue from the system parameters point of view, used to model it. Opposite to the typical great number of them, an agent is used to incorporate local information; consequently, it considers only a part of this parameter set.

Decision making and reasoning are other important aspects of the analogue. Based on the definition of the agent itself, it makes decisions, usually, to attain their own goals. Typically, time is needed to achieve or to come nearer to their targets; consequently, it is especially important to have internal foresight capability. It needs models in its own knowledge representation which allow inferences for time ahead. The local information can be in accordance with the parameters of one or more submodels explored with the introduced algorithm that is only a part of the whole parameter set. The analogue of foresight capability can be satisfied through the application of the method on a database having parameters concerning the description of the time relevant behaviour of the analysed system. *Especially important is the main basis of the analogue, namely, this can be ensured, because the basic target of the agent is to reach the highest level of foresight capability in the presented approach.*

A basic feature of artificial neural networks is their learning ability which is also one of the most required properties of agents. As explained above, a set of ANNs is one of the main results of the solution, consequently, the analogue can be detected when these submodels with learning ability are internal parts of the agent knowledge base.

Finally, the analogue inherited from the network nature should be emphasised. If the submodels or submodel groups are ordered to individual agents the received net of submodels can be corresponded to agents communicating with each other through sharing values of common system parameters.

The application of the method for high-level of production control: realizing learning agent identification with combined simulation and submodel identification techniques. Analogues detailed in the previous paragraph serve as basis for the identification of agents in production systems. As explained before, one of the prerequisites of the submodel exploration technique required also for agent identification is a table containing data vectors describing the behaviour of the system concerned. This data set can be collected by production control systems connected to manufacturing equipment or can be typically generated by a simulation model (Gerdes, *et al*., 2005).

Decision points incorporated in production systems are analysed at first. To make it simple, let us assume that there exists an agent structure describing the given system (e.g. its restructuring is the main assignment) and also a simulation model have been built up. Other cases can be treated similarly to this one. The next part defines the contents of the date vectors as a coding of system states.

Agents make decisions, consequently, a part of the date vector parameters consist of their variables. Another part of these vectors is formed by the internal measures of the agent, while a further part consists of parameters from the observation of its environment. Examples for the first part are, e.g. capacity utilisation from the past and from the future, level of present, bidded, scheduled occupation, values of own target function, foreseen order types. The later one can be formed by external environment observations but, moreover, by some communication of agents. These three types of parameters will be specified for the solution, e.g. these variables have to be defined and collected for all of product, resource, order and staff agents in the case of PROSA architecture (Valckenaers, *et al*., 2001).

Ordering the parts of parameters to each-other is the next main question. Various, e.g. time-shifted solutions can be introduced; preparation of date vectors with parameters coming from handling the elements of the same order can be an order oriented, very simple solution.

Having the data set defined allows running the submodel exploring algorithm. *A set of submodels having at least one common parameter is ordered to one agent giving the knowledge base to it and serving with learning ability, too. A separated set of submodels allows identifying different agents.* The contents of data table, as the basis for dependency exploration, contains, in an explicit or implicit way, the time parameter, consequently, the ordered submodels ensure the required foresight capability, too. The requirement for continuous validation of the agent's knowledge base will be emphasised, moreover, system restructuring is required if repeated learning cannot result in an appropriate level of the model accuracy. This makes the application possibilities of reinforcement learning techniques stronger in this field.

Not all the submodel findings result in a separated model set. In this case the minimisation of common parameters among model sets can specify the individual agents. The values of these parameters have to be shared among agents, causing a continuous communication among them. Another communication of agents is inherited from the information exchange between the whole analysed system and its environment.

This paragraph described a concept and steps for automatic agent identification by using the submodel finding technique and the simulation model of the analysed system. These individual steps can be solved also in another way bringing up further research activities. One of the main challenges is to find the balance between the field-specific agent (pre)definition and the introduced, automatic agent identification approaches. This research is already in the conceptual phase; concrete realizations and test are just running.

4. CONCLUSIONS

The paper presents two applications of the novel, artificial neural network and feature selection based combined, dynamic technique to automatically dissolve a large, complex system into a net of connected submodels. The first application is a solution for the lower level of customised mass-production systems, for increasing their productivity. The second one is a concept for the identification and reorganisation of manufacturing agents, based on simulation experience. The main idea behind, is to give the learning capability to agents already at their definition phase, and to maximise their foresight power.

5. ACKNOWLEDGEMENTS

The research was supported partially by the National Research Found, Hungary, Grant No. T043547.

REFERENCES

Basak, J. (2004). Online adaptive decision trees, *Neural Computation*, **Vol. 16**, 1959-1981

Brouwer, R. K. and W. Pedrycz (2003). Training a feed-forward network with incomplete data due to missing input variables. *Applied Soft Computing*, **Vol. 3**, pp. 23-36

Caelli, T., L. Guan and W. Wen (1999). Modularity of Neural Computing. *Procedings of the IEEE*, **Vol. 87, NO. 9**, pp. 1497-1518

Gerdes, M, T. Barth, M. Grauer (2005). Performance analysis of distributed solution approaches in simulation-based optimization, *CMS Computational Management Science*, **Vol. 2.**, pp. 57–82

Guan, L., J. A. Anderson, J. P. Sutton (1997). A Network of networks processing model for image regularization. *IEEE Transactions on Neural Networks*, **Vol. 8, NO 1.**, pp. 169-174

Haidegger, G. and A. Popa (2002). Assessment of multimedia technologies for use in digital factory projects. *Third Conference on Mechanical Engineering*, pp. 495-499

Hu, W., D. Xie and T. Tan (2004). A hierarchical self-organising approach for learning the patterns of motion trajectories. *IEEE Transactions on Neural Networks.* **Vol. 15, NO 1.**, pp. 135-144

Jordan, M. I., R. A. Jacobs (1994). Hierarchical mixture of experts and the EM algorithm, *Neural Computation*, **Vol. 6,** pp. 181-214

Kovács, A., J. Váncza, B. Kádár, L. Monostori and A. Pfeiffer (2003). Real-life scheduling using constraint programming and simulation. *7th IFAC Workshop on Intelligent Manufacturing Systems*, pp. 233-238

Kung, S-Y., J. Taur, S-H. Lin (1999). Synergistic modeling and applications of hierarchical fuzzy neural networks. *Proceedings of the IEEE,* **Vol. 87, NO 9**, pp. 1550-1574

Lu, S. and A. Szeto (1993). Hierarchical artificial neural networks for edge enhancement. *Pattern Recognition*, **Vol. 26, NO. 8**, pp. 1149-1161

Mason, R., W. Robertson (1995). Mapping hierarchical neural networks to VLSI hardware. *Neural Networks*, **Vol. 8, NO 6**, pp. 905-913

Merchant, M. E. (1998). An interpretive look at 20th century research on modelling of machining, Inaugural Address, *Proc. of the CIRP International Workshop on Modelling of Machining Operations*. pp. 27-31

Monostori, L., G. Haidegger, J. Váncza and Zs. J. Viharos (2002). Digital Enterprises: A national R&D project in Hungary. *Proceedings of the 1st CIRP (UK) Seminar on Digital Enterprise Technology, DET02, Part V: Enterprise Integration Technologies*, pp. 269-272

Monostori, L., Viharos Zs. J. and S. Markos (2000). Satisfying various requirements in different levels and stages of machining using one general ANN-based process model. *Journal of Materials Processing Technology*, **Vol. 107, NO 1-3**, pp. 228-235

Tollenare, T. (1990). SuperSAB: fast adaptive backpropagation with good scaling properties, *Neural Networks*, **Vol. 3**, pp. 561-573

Valckenaers, P., H. Van Brussel, M. Kollingbaum, M., and O. Bochmann (2001). Multi-Agent Coordination and Control Using Stigmergy Applied to Manufacturing Control. *European Agent Systems, Summer School*, pp. 317-334

Viharos, Zs. J., L. Monostori and T. Vincze (2002). Training and application of artificial neural networks with incomplete data, *Lecture Notes of Artificial Intelligence*, pp. 649-659

Viharos, Zs. J., L. Monostori and Z. Csongrádi (2003a). Realizing the digital factory: Monitoring of complex production systems. *7th IFAC Workshop on Intelligent Manufacturing Systems*, pp. 31-36

Viharos, Zs. J., S. Markos and Cs. Szekeres (2003b). ANN-based chip-form classification in turning. *XVII IMEKO World Congress - Metrology in the 3rd Millennium*, pp. 1469-1473