Viharos, Zs. J.: AI based, automatic production system decomposition; 5th International Conference on Digital Enterprise Technology, DET2008, October 22-24, 2008, Nantes, France, paper nr. 94.

AI BASED, AUTOMATIC PRODUCTION SYSTEM DECOMPOSITION

Dr. Zsolt János Viharos
Computer and Automation Research Institute, Hungarian Academy of Sciences
zsolt.viharos@sztaki.hu

ABSTRACT

The paper introduces a new approach for automatic plant decomposition, based on artificial intelligence (AI) techniques. A novel automatic solution, based on a generalized feature selection technique and on artificial neural network (ANN) training, was developed by the author. The main goal is to explore connections among parameters of a given database, and based on the modelled dependency sets, validated groups of parameters can form individual parts of the analysed system. Applying this technique to a production database containing data typically inherited from the shop-floor level through process monitoring systems (or based on virtual simulation models) results in groups of connected production parameters. Consequently, it allows decomposing a running or simulated manufacturing system into smaller, individual and autonomous components. Therefore, the approach can provide the basis for production system decomposition and reconfiguration, too.

KEYWORDS

Artificial Intelligence, Decomposition Method, Monitoring, Simulation, Reconfiguration

1. INTRODUCTION

Reliable process models are of key importance in computer integrated manufacturing (Merchant, 1998) as model-based solutions can make difficult problems of production control tractable. Increasing complexity is another characteristics which shows up in production processes, systems and in enterprise structures as well. Models facilitate elaborating new algorithms, supporting decisions, decreasing investment risks and coping with changes and disturbances. However, modelling manufacturing processes may bear difficulties: the diversity of operations, their multidimensional, nonlinear and stochastic nature, partially understood relations, unreliable or incomplete data sets etc. Often, the only feasible approach is the decomposition of the model into several smaller interconnected sub-models—though not equal to problem decomposition, but is one of the first steps towards it.

Learning denotes changes in the system adaptive in the sense that learning techniques enable the system to do the same or similar task more effectively next time. Artificial neural networks (ANNs) are general, multivariable, nonlinear estimators. This artificial intelligence (AI) technique can offer viable solutions, especially for problems where abilities for real-time functioning, uncertainty handling, sensor integration, and learning are essential features.

After highlighting some methods to decompose production systems, the current state-of-the-art in the structure determination of artificial neural network systems is presented, followed by the description of the sub-model exploration algorithm, and its positioning. Two applications for high- and low-levels of production control are explained for illustrating the applicability of the proposed method for decomposing production systems, e.g., plants. Highlighting some issues and further improvement fields is necessary to facilitate new ideas for researchers. Conclusions, acknowledgement and references close the paper.

2. PRODUCTION SYSTEM DECOMPOSITION AND RECONFIGURATION METHODS

Various approaches for decomposing production systems can be found in the literature, this paragraph highlights some examples only. Production system model is built up from two fixed elements, puffers and machines, while these elements can have various states and state transition processes (Colledani and Tolio, 2005). The structuring prescribed allows calculating system performance parameters, e.g. throughput, availability, etc., by using Markov computation techniques. An extension of IEC (International Electrotechnical Commission) 61499, standard on function blocks was introduced to allow a function block-oriented reconfiguration of factories (Olsen et al., 2005). Typical reconfiguration cases, prescribed from this stage on (e.g. failure of the controller, introduction of new processes), are ordered to function blocks. Self-reconfiguration of modular robots is set-up where the robot itself can adapt its prescribed configuration containing specific elements to execute uncommon tasks (Lau et al., 2008). The approach is based on agent techniques for solving the problem of distributed capacity allocation in manufacturing systems (Bruccoleri et al., 2005). The paper presents a valuable table about typical planning issues, decision makers and planning mechanisms in five different levels and horizons of production planning. The paper, together with the above enumerated ones, illustrates excellent examples for various kinds of production system decomposition elements and reconfiguration methods. A similarity among the methods is that the applied system elements are static and prescribed according to some traditional or modelling rules. This is typical also in building up production simulation systems (Monostori and Viharos, 2001). A promising technique is served by the approach of lean manufacturing where the elements of a manufacturing cell (machine, puffer, and material movement equipments) are typically prescribed but the number of sequential machines and stations involved are determined and calculated based on the customer's beat, so in this point of view, this approach gives a partly dynamic solution.

The paper introduces a different approach where only the parameters describing a production system are fixed but the elements inherited from groups of parameters of the decomposed manufacturing architecture are determined dynamically during decomposition.

3. STRUCTURES OF ARTIFICIAL NEURAL NETWORK MODELS

The description of various structure determination techniques is important for illustrating the position of the introduced ANN based model decomposition algorithm, moreover, it is probably more important to help facilitate new system (re)configuration techniques in production management.

Various approaches can be found for improving the structure of ANN models in the literature e.g., in the case of MLPs, too. Mainly two types of techniques are used for preparing the net of ANN models. In the first type the structure is pre-determined before the learning, consequently, it is not mainly based on the data set available, but the structure is defined typically according to some tasks of the given application field. The second group of neural network structure building techniques is applied during the learning phase, since, they are based on the explored dependencies in the analysed database. The following paragraphs describe typical solutions for structure determination of ANNs. Mainly publications the first ideas coming first in various fields are enumerated together with some basic features of the solutions.

3.1. PRE-DETERMINED STRUCTURES

The first group of pre-determined neural network structures relates to hierarchical modelling, but the second one combines, in various ways, similar equivalent sub-networks.

3.1.1. Hierarchical, pre-determined structures

A fixed, three-level hierarchical neural network system is presented (Ding and Yue, 2004) recognizing workpiece features. The hierarchy is organized according to a special engineering approach regarding the recognition task. Path algebra is used for building up hierarchical model(s) to describe a complex system (Gentila and Montmain, 2004). The approach requires human-machine interaction to drive the model development in the direction of the given analysis. The paper highlights also other interesting ideas, e.g., understanding system’s behaviour is an important requirement for supervision and is, generally, opposed to accuracy. It is also identified that hierarchical decomposition by refining models into components is crucial for managing complexity. A two-level hierarchy network for learning trajectory distribution patterns is initiated (Hu et al., 1997) where an internal net can be considered as a "big neuron" and the relationships are defined by neighbourhoods. Special sub-tasks, such as filtering, orientation extraction or filling-in in the field of texture processing determined the components of the hierarchical, artificial neural network based evaluation tool (Van Hulle and Tollenaere, 1993). Several possible hierarchies in neural network structures are shown (Kung et al., 1999) and were combined with fuzzy techniques. Contour orientation detection, decision making to alter the gradient magnitude and adjusting the direction of the edge element were the tree sub-tasks for edge enhancements that initiated the hierarchical structure incorporating three neural networks (Lu and Szeto, 1993). Frames were defined in a picture called texture window, based on field-specific targets to find similar regions in it (Goltsev, 1996). Neural network models were ordered to the defined regions and all sub-networks are incorporated into a single network by a neural activity control system. A very interesting comparison between mathematical techniques and artificial neural networks was described (Watanabe, 2001). It was presented that almost all homogeneous and hierarchical learning machines have singularities in their parameter space resulting in having no mathematical foundation to analyse their learning. Also the essential difference between the regular statistical models and artificial neural networks was given.

3.1.2. Non-hierarchical, pre-determined structures

To decrease the error of a system similar neural network models are used (Hashem, 1997; Tetko and Villa, 1997) for reaching the optimal combination of their outputs by applying correlation coefficients to measure the similarity between the cases from a domain. The weighted sum of pre-determined neural network model outputs another estimation technique combining several models at the same time (Wolperta and Kawato, 1998). Another interesting feature of their motor control solution is that they prepared forward and also backward models, however, these models have to incorporate the same dependencies among parameters. It was referred to that this symptom was recognized in the cerebellum.

3.2. SELF-DETERMINING STRUCTURES

This group of techniques for building neural network structures is applied during the learning phase, incorporating two sub-solutions: first, constructive building with growing the ANN structure, e.g., adding neurons and/or links to a momentary model during training, second, prunning is typical opposite solution with removing links and/or neurons, usually, to improve the accuracy and the learning speed.

Decision trees and neural network models were combined (Basak, 2004) in a way that ANNS are located in the branches of the decision tree model allowing to build up a hybrid model with an integrated learning technique. Also a tree combined with neural networks resulted in the self generating network of networks presented (Caelli et al., 1999). The paper shows various modular neural network based solutions, applications and it highlights the important aspect that neurons in human brain are very sparsely connected. Exploring independency among parameters in the neural network structure moves the model in this direction (Yasui, 1997). A constraint-based decomposition (CBD) training architecture was introduced (Draghici, 2001) with the corn idea behind to reduce the dimensionality of the search space through decomposing the problem into sub-problems using sub-goals and constraints defined in the problem space. The number of neural nodes is prescribed by four in one layer. There exist a node for fixed value, another node for linear mapping and two nodes for non-linear dependencies. The number of layers varies in a dynamic architecture (Ghiassi and Saidane, 2005). They also highlighted that for a class of problems (in their case for electrical load forecasting) a desired level of accuracy is often prescribed. Also a practical aspect of easy hardware realization was the motivation for ordering neural nodes and partly networks into hierarchy (Mason and Robertson, 1995). Similarly to one of the motivations of the model building algorithm presented in the paper, non-invertible and multi-value dependencies have to be handled by the introduced partitioning system (Gock and Katupitiya, 2005). Sequential Forward Search (SFS) was an applied technique for selecting output parameters (Guan et al., 2005) describing also that connected sub-models work better than one comprehensive model.

4. ANN BASED DECOMPOSITION

The ANN based decomposition method is introduced in the next paragraph; it is illustrated by the user point of view. 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. This paragraph describes the algorithm from the user's point of view.

The application of the algorithm has two main prerequisites:

- The user has to provide 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. One has to define when to stop a learning process. Implicitly, the user defines the level of estimation accuracy, or the error allowed. This setting differs from parameter to parameter but it has to be determined before 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, the settings of the accuracy is one of the main domains for further improvement of the method.

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

- Net of accepted sub-models. They can perform the estimation of their output parameters with the prescribed, individual accuracy. They can have common parameters; consequently, the result is a net of neural networks.
- List of rejected sub-models. These models were analysed during the search but they were rejected. 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 sub-models. This allows the prompt application of the whole, or a part of the net of sub-models for solving various assignments.

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

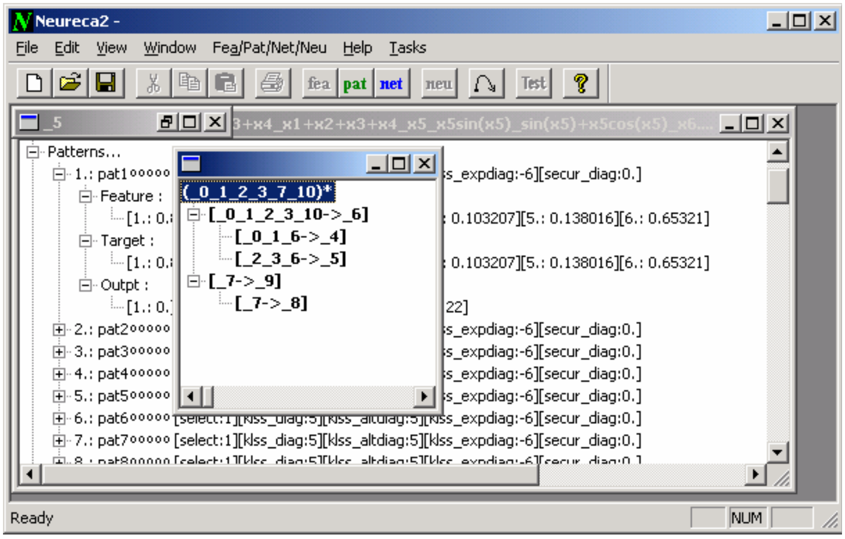


Figure-1: The resulted sub-model structure of a complex system

5. POSITIONING THE INTRODUCED ALGORITHM

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

According to the core modelling technique, the solution is based on neural networks. Multi-Layer 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 backpropagation called SuperSab (Tollenare, 1990) but it was modified several times.

The model building method can be considered as a special learning algorithm, too. It does not require pre-determining 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. Moreover, having certain input-output configurations at any stage or at the end of this algorithm run, the related models can be trained in a supervised way; consequently, the learning algorithm can be considered a special one with two faces of supervised and unsupervised learning at the same time.

The modelling of many-valued mapping is also solved by the introduced algorithm. A similar problem is identified and solved excellently by a totally different approach (Brouwer and Pedrycz, 2003). By coincidence, in the next step they examined also the field of incomplete data as this was also the case with the presented algorithm of authors (Viharos, et al., 2002). Currently the introduced sub-model identification solution is fully prepared to handle databases having many incomplete data sets.

Various approaches can be found in the literature (see above) 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 in an outcome similarly to a pruning/learning process combination, so it can be considered as a very special form of pruning solution, too.

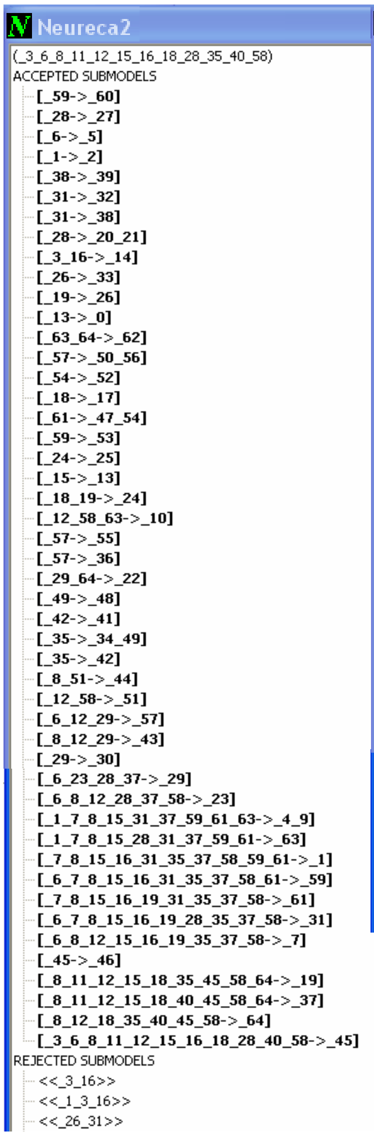
The applications of ANNs are typically preceded by a feature selection algorithm, especially in the field of manufacturing, to surmount their capability restrictions, with respect to the number of parameters and thus, model sizes. It can be found that feature selection and training processes are separated. The new, introduced algorithm brakes with this practice: it is a dynamic, integrated combination of these two steps. Consequently, the algorithm can be considered as a special feature selection solution, or also as a hybrid combination of feature selection and learning based model building.

6. DECOMPOSITION OF PRODUCTION SYSTEMS

The next two paragraphs illustrate two applications of the sub-model exploration algorithm. Application results are presented based on a data source collected to a production line, followed by an approach for high-level manufacturing application.

6.1. DECOMPOSITION AT LOW-LEVEL PRODUCTION CONTROL

A part of a Hungarian R&D project was aimed at building production models where dependencies among parameters are unknown. Modelling through learning is based on collected manufacturing data sets. No measurements were needed in the application in question, because of the huge number of related monitoring parameters. The 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 stating "there should be some connections among these parameters" was quite promising at the beginning. The introduced sub-model finding technique was applied.



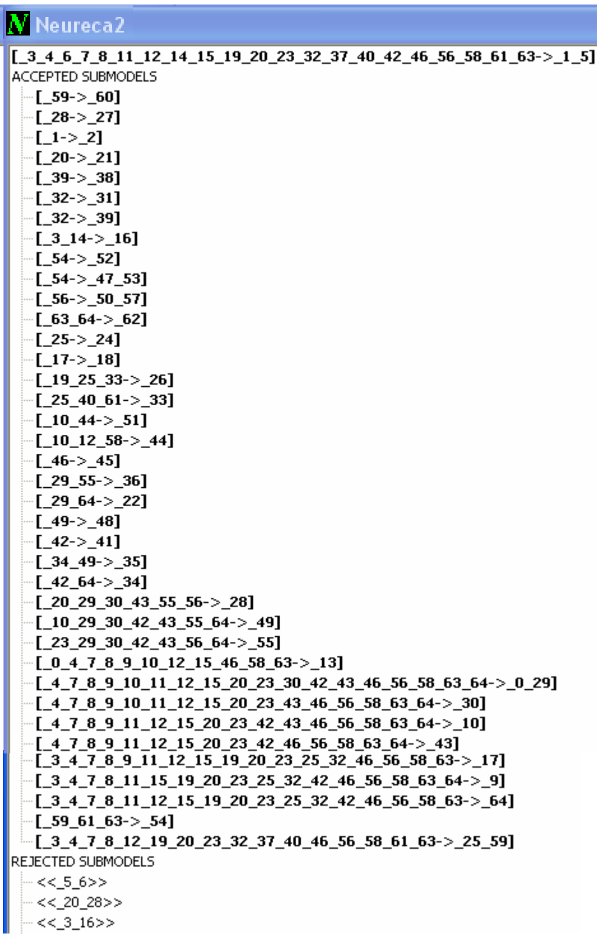


Figure-2 (LEFT): The explored complexity of dependencies among different parameters (represented as numbers) of machines within a production line prescribed with 8.8% of expected estimation accuracy. Figure-3 (RIGHT): The explored complexity of dependencies among different parameters (represented as numbers) of machines within a production line prescribed with 5.6% of expected estimation accuracy.

➔ Figure-2 and Figure-3 show results 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. Consequently a part of a production system was studied, indicating that a comprehensive analysis is an enormously complex and difficult task in the production line level. The expected levels of accuracy are different concerning Figures 2. and 3., namely, +/- 8.8% and +/- 5.6% respectively. Higher levels of allowed errors provide more and also smaller sub-models as represented in the pictures. It has to be mentioned that only a short list of examples of the rejected sub-models is highlighted in the pictures. The ratio in the number of accepted and rejected sub-models is approximately 1:7 in both of the cases.

6.2. DECOMPOSITION AT HIGH-LEVEL PRODUCTION CONTROL

This paragraph details the concept how to apply the above introduced algorithm for high-level 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. Agent identification for control production systems is, usually, solved through field-specific approaches. A well-known example is the PROSA architecture (Valkenaers, et al., 2001) where the identified agents are components of manufacturing systems, such as product, resource, order and staff. Not overemphasising either self-determination of main entities or downgrading the otherwise very important professional know-how incorporated in the pre-determined structure, still 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 sub-model exploration assignments. The following paragraph shows the analogues between the sub-model and agent identification, as the basis of the concept, as follows:

- The exploration of separate, small sub-models 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. Opposite to the usually great number of parameters, an agent is used for incorporating local information; consequently, it considers only a part of a parameter set.
- Decision making and reasoning are other important aspects of the analogue. Based on the definition of the agent itself, they make decisions, usually, to attain their own goals. To achieve the targets, it is especially important to have internal foresight capability. Models are needed in their own knowledge representation, which allows inferences for time ahead. The local information can be in accordance with the parameters of one or more sub-models 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.

- The 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 introduced solution. Consequently, the analogue can be detected when these sub-models with learning ability are internal parts of the agent knowledge base.
- Finally, the analogue derived from the network nature should be emphasised. If the sub-models or sub-model groups are ordered to individual agents, the received net of sub-models can be corresponded to agents communicating with each other through sharing values of common system parameters.

The application of the method in high-level production control: realizing learning agent identification with combined simulation and sub-model identification techniques. Analogues detailed above serve as basis for the identification of agents in production systems. As mentioned before, one of the prerequisites of the sub-model 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 generated by a simulation model (Gerdes, et al., 2005).

Decision points incorporated in production systems are analysed first. Let us assume that there exists an agent structure describing a given system (e.g. its restructuring is the main assignment) and also a simulation model has been built up. Other cases can be treated similarly to this one.

The next part defines the contents of the data vectors as a coding of system states. Agents make decisions, consequently, a part of the data vector parameters consist of their variables. Another part of these vectors is formed by the internal measures of the agents, while a further part consists of parameters from the observation of their environment. Examples for the first 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 second 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 sub-model exploring algorithm. A set of sub-models having at least one common parameter is ordered to one agent, giving the knowledge base to it and ensuring learning ability, too. A separated set of sub-models 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 sub-models 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 sub-model findings result in a separated model set. In these cases the minimisation of common parameters among model sets can specify the individual agents. The values of these parameters have to be shared among agents, initiating 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 sub-model finding technique and the simulation model of the analysed system. These individual steps can be solved 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.

7. CONCLUSIONS

The paper introduced a new approach for automatic plant decomposition based on artificial intelligence techniques. A novel automatic solution based on a generalized feature selection technique and on artificial neural network training was developed by the author. The main goal is to explore connections among parameters of a given database. Based on the modelled dependency sets validated groups of parameters can form individual parts of the analysed system. Applying this technique to a production database containing data inherited typically from the shop-floor level through process monitoring systems (or based on virtual simulation models) results in groups of connected production parameters. Consequently, this solution allows decomposing a running or simulated manufacturing system into smaller, individual and autonomous components, threrefore, the approach can provide the basis for production system decomposition and reconfiguration. Two applications of the sub-model exploration algorithm were illustrated. Firstly application results were presented based on a data source collected from a production line, secondly an approach for high-level manufacturing application was detailed. The latter makes the identification of the individual parts of a production system possible. Consequently, it can be used for system decomposition at plant level and for identifying agents of a distributed production control solution.

8. ACKNOWLEDGMENT

The work presented here was supported partly by the European Framework projects of RELIAWIND – Cost effective wind farms and wind turbines design, operations and maintenance through focus on reliability (Proposal Nr. 212966), and Hungarian National Research Fund under grant No. OTKA T049481.

REFERENCES

Basak, J., “Online adaptive decision trees”, *Neural Computation*, Vol. 16, No. 9, 2004, pp. 1959 - 1981

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

Bruccoleria, M., Nigroa, G. Lo., Perronea, G., Rennab, P. and Diega, S. Noto La, “Production planning in reconfigurable enterprises and reconfigurable production systems”, *Annals of the CIRP*, Vol. 54, No. 1, 2005, pp. 433-436.

Caelli, T., Guan, L. and Wen, W., “Modularity in neural computing”, Proceedings of the IEEE, Vol. 87, No. 9, 1999, pp. 1497-1518.

Colledani, M. and Tolio, T., “A Decomposition Method to Support the Configuration / Reconfiguration of Production Systems”, *Annals of the CIRP*, Vol. 54, No. 1, 2005, pp. 441-444.

Ding, L. and Yue, Y., “Novel ANN-based feature recognition incorporating design by features”, *Computers in Industry*, Vol. 55, No. 2, 2004, pp. 197-222.

Draghici, S., “The constraint based decomposition (CBD) training architecture”, *Neural Networks*, Vol. 14, No. 4, 2001, pp. 527-550.

Gentila, S. and Montmainb, J., “Hierarchical representation of complex systems for supporting human decision making”, *Advanced Engineering Informatics*, Vol. 18, 2004, pp. 143-159.

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

Ghiassi, M. and Saidane, H., “A dynamic architecture for artificial neural networks”, *Neurocomputing*, Vol. 63, 2005, pp. 397-413.

Gock, K. and Katupitiya, J., “Backwards neural networks optimization”, Proc. of the 16th IFAC World Congress, Prague, Czech Republic, 2005. CD version available

Goltsev, A., “An assembly neural network for texture segmentation”, *Neural Networks*, Vol. 9, No. 4, 1996, pp. 643-653.

Guan, S.-U., Yinan, Q., Tan, S. K. and Li, S., “Output partitioning of neural networks”, *Neurocomputing*, Vol. 68, 2005, pp. 38-53.

Hashem, S., “Optimal linear combinations of neural networks”, *Neural Networks*, Vol. 10, No. 4, 1997, pp. 599-614.

Hu, Y.H., Palreddy, S. and Tompkins, W., “A patient adaptable ECG beat classifier using a mixture of experts approach”, *IEEE Trans. Biomed. Eng.* Vol. 44., 1997, pp. 891-900.

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

Lau, H. Y. K., Ko, A. W. Y. and Lau, T. L., “The design of representation and analysis method for modular self-reconfigurable robots”, *Robotics and Computer-Integrated Manufacturing*, Vol. 24., 2008, 258-269.

Lu, S. and Szeto, A., “Hierarchical artificial neural networks for edge enhancement”, *Pattern Recognition*, Vol. 26, No. 8, 1993, pp. 1149-1163.

Mason, R.D. and Robertson, W., “Mapping Hierarchical Neural Networks to VLSI Hardware”, *Neural Networks*, Vol. 8, No. 6, 1995, pp. 905-913.

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

Monostori, L., Viharos, Zs. J., “Hybrid, AI- and simulation-supported optimisation of process chains and production plants”, *Annals of the CIRP*, Vol. 50, No. 1, 2001, pp. 353-356.

Tetko, I. V. and Villa, A. E. P., “Efficient partition of learning data sets for neural network training”, *Neural Networks*, Vol. 10, No. 9, 1997, pp. 1361-1374.

Tollenaere, T., “SuperSAB: Fast adaptive backpropagation with good scaling properties”, *Neural Networks*, Vol. 3, 1990, pp. 561-573.

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

Van Hulle, M. and Tollenaere, T., “A modular artcial neural network for texture processing”, *Neural Networks*, Vol. 6, No. 1, 1993, pp. 7-32.

Viharos, Zs. J., Monostori, L and Vincze, T., “Training and application of artificial neural networks with incomplete data”, *Lecture Notes of Artificial Intelligence*, Vol. 2358., 2002, pp. 649-659.

Watanabe, S., “Algebraic geometrical methods for hierarchical learning machines”, *Neural Networks*, Vol. 14, 2001, pp. 1049-1060.

Wolperta, D. M. and Kawato, M., “Multiple paired forward and inverse models for motor control”, *Neural Networks*, 1998 Special Issue, Vol. 11, 1998, pp. 1317-1329.

Yasui, S., "Convergence Suppression and Divergence Facilitation: Minimum and Joint Use of Hidden Units by Multiple Outputs", *Neural Networks*, Vol. 10, No. 2, 1997, pp. 353-367.