AUTOMATIC PARTITIONING OF PROBLEMS THROUGH SUBMODEL DECOMPOSITION— A PROMISING TECHNIQUE OF DIGITAL ENTERPRISE TECHNOLOGY

Zsolt János Viharos, László Monostori, Zsolt Kemény
Computer and Automation Institute, Hungarian Academy of Sciences
{viharos,monostor,kemeny}@sztaki.hu

This paper presents two main groups of results in the field of process modeling; first, highlighting complexity-related properties shared by several, if not all, levels of production; second, a family of methods set up to handle the aforementioned problems. The presented algorithms lead up to a submodel decomposition method combining generalized feature selection and artificial neural networks which further research may extend to an automatic determination of agent boundaries in complex systems. Aside from theoretical presentation, practical results in various levels of actual industrial production demonstrate the feasibility of the methods, suggesting that they can cope with complexity-related problems of different production levels in a uniform way.

1. INTRODUCTION

Reliable process models are of key importance in computer integrated manufacturing (see Merchant 1998) as model-based solutions can make difficult problems of production control tractable. Models facilitate elaborating new algorithms, supporting decisions, decreasing investment risks and coping with changes and disturbances.

However, modeling 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* to several smaller interconnected submodels—though not equal to *problem decomposition* but a first step towards it.

In modeling complex systems, it is common practice to highlight relevant variables in measurement data, e. g. as through *feature selection*. From the selected parameters, process models are often obtained and maintained through learning (see Viharos and Monostori 2001). This can be achieved e. g. through artificial neural networks (ANNs)—general, multivariable, nonlinear estimators which can be trained to represent a given model.

In the first part of this paper (section 2), common characteristics of various production levels are addressed, along with the idea of applying a method elaborated for one production level to other layers of the hierarchy. Section 3 presents a submodel decomposition method combining feature selection and ANN training, meant to handle complexity-related problems of various production levels. Finally, section 4 gives practical application examples in various levels of production.

2. SIMILARITIES BETWEEN PRODUCTION CONTROL LEVELS WITH RESPECT TO MODELING AND ASSIGNMENT TASKS

2.1 Hierarchic levels of production systems

No single prevailing scheme identifying production levels exists, yet all share some fundamental principles. A general overview of noteworthy stratification schemes can be found e. g. in Horváth and Markos (1995), Tóth (1998), Luttervelt *et al.* (1998) and T. Tóth (1989). The figure below shows the comparison of the MAP/TOP model for communication in production, and the layers as addressed throughout this paper.

The examination of production levels, independently from the specific scheme used, shows one general phenomenon: various forms of complexity, as detailed in the next subsection, appear throughout all layers of the hierarchy.
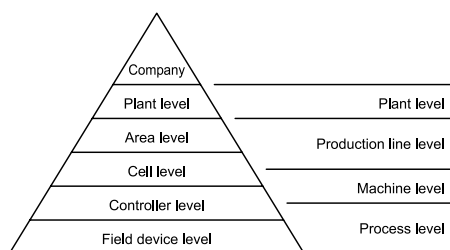

Figure 1. Classification of production levels according to the scheme used in MAP/TOP (left), and the convention used by Viharos et al.(right)

2.2 Similarities between levels of production systems

As many other domains, all levels of production systems exhibit high complexity and uncertainty, requiring efficient methods for learning the system's properties and dependencies and depositing them in a reconfigurable form to suit various interpretations. Though all levels of production bear specific inherent difficulties, the following characteristics are shared by all systems and levels in general:

A high number of parameters. For systems consisting of many components, such as production lines, biochemical processes etc., a large number of parameters is required for description, and the control and modeling experts' efforts to work with simplified models may fail. For manufacturing processes, a variety of approaches exists for simplification (see Tóth 1998), but even using those leave a large number of variables.

A high number of dependencies. Production systems are composed of several components, each of which may have its own complex set of relations. Combining these components in a large system, the number of relevant dependencies quickly increases, even if some of the less relevant inherent relations of the components are omitted.

Nonlinear dependencies. Though some "conventional" methods can also handle nonlinear models, in most cases their success depends on a-priori information about the general nature of the nonlinearity— which is rarely given to a proper degree in unknown complex systems, especially in the field of production.

Uncertainty of measurement data. Gathering knowledge about physical systems through measurement introduces noise and data uncertainty. This calls for robust methods to handle uncertain parameters within the corresponding tolerance limits.

Incomplete information. Related to uncertainty is the partial availability of information, meaning that some elements of measurement data vectors are missing (either due to sensor malfunction or because at the given point of the process, the data element

has no sensible meaning) or invalid. Discarding these incomplete vectors can often be unpracticable as they are a substantial part of all data. Therefore, a method is needed to handle incomplete data sets (see Viharos *et al.* 2002, Zhang and Rong 2005).

Unknown input/output character of parameters. The input/output nature of relevant parameters may change—either depending of the problem to be solved or the given point of view of modeling (and may thus be different for another problem), or it may be entirely unknown if a-priori knowledge about the system is sparse—this is typical for ANN application cases (see Viharos and Monostori 1999c).

Numerous methods can master one or a few of the above challenges, yet few can address all of them simultaneously—this may give the latter approaches an outstanding role in future production control. It is of key importance to develop methods that address generic system properties in the fundamental structure of the methodology, while the algorithms applied can be exchanged for others in a "plugin-like" manner. This—just as the fact that all production levels share the above features—encourages researchers to apply one method, elaborated for one given level of production, to solve problems of other layers of the hierarchy as well, promising a versatile problem-solving technology. This paper presents such a method, submodel decomposition, which is able to cope with the aforementioned challenges and was, in a number of industrial application and test cases, successfully used in various levels of production.

3. INTRODUCTION TO SUBMODEL DECOMPOSITION

3.1 Description of the submodel decomposition method

Now, let us present the submodel decomposition method in detail. The procedure we propose combines generalized feature selection and improved ANN training in a dynamic way. Generalized feature selection proposes a set of assumed submodels which are individually validated by ANN training, either to be accepted or to be rejected.

3.1.1 Generalized feature selection. Let us assume that submodels have to be extracted from a list of n parameters whose input/output nature is not known in advance. Feature selection proposes n possible submodels, each having another output variable, and a list of further parameters in the order of their potential impact on the output.

The original feature selection algorithm by Devijver and Kittler (1982) assumes a pure classification task with the goal of reducing the number of inputs needed for one single output. As a generalization, continuous output parameters can be mapped onto the discrete classification scheme with an appropriate heuristics. The first step selects the output and its values encountered in the training data set are grouped into the highest possible number of clusters (i. e. intervals of equal length), so that at least one element is contained in each interval (this, in itself, being the first heuristic decision).

Ranking remaining variables using suitable heuristics (see Devijver and Kittler 1982) with respect to a given "measure of distinction" delivers a list of variables and corresponding measure values. Taking more and more of them for input, the separability measure of the corresponding output deteriorates, and a further heuristic decision can determine how many of the best-ranked variables should be taken. These variables, together with the selected output, form one possible submodel, *ignoring whether they were selected for input or output during feature selection*. This is repeated with all parameters of the system selected as outputs, resulting in as many proposed submodels as the total number of parameters. Continuing the same with recursive partitioning of all submodels already created, the possible submodels form a multilevel decision tree.
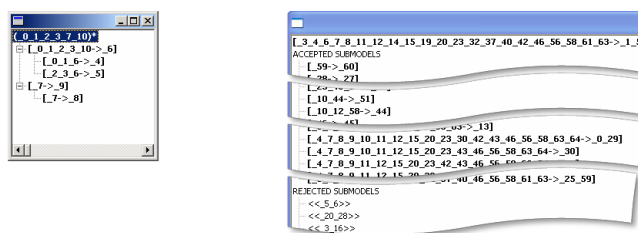
Figure 2.   Two cases of submodel decomposition. In the example to the left, the net of accepted submodels consists of five main relations (in brackets), partitioning a system containing eleven description parameters. The fourth row in the window, e. g. shows that the algorithm identified a submodel with parameters 2, 3 and 6 as inputs for the estimation of output 5. The four identified submodels have common parameters, e. g. parameter 6 is estimated by the submodel shown in the second row, but it is to be found among the input variables of the next two submodels, too. Thus, a structure of interconnected submodels can be recognized additionally to the identification of its individual parts. To the right, the result of submodel decomposition in an industrial example with a large number of system parameters is shown.

3.1.2   ANN training. The three heuristic decision steps taken to obtain the candidate submodels makes them only *assumptions*, to be either *verified* or *rejected* by ANN training. To overcome difficulties of "classical" ANN application, a generic, reusable ANN-based model should compiled—such a strategy is introduced in the approach of Viharos and Monostori (1999c). Supplying the algorithm with a sufficiently large set of training patterns and corresponding tolerances, the best input/output configuration for learning the dependencies of the training vectors will be found automatically. The successful completion of the latter is equal to a proof of the assumed submodel as it is practically realized by the ANN. Does this not succeed, the model is rejected and will be avoided in subsequent steps.

3.1.3   Dynamic submodel decomposition. While the separate application of preparing training data by feature selection and subsequent ANN learning already reduces the computational costs of modeling the entire system, the flexibility of the method is most exploited if feature selection and ANN training are executed as complements in alternating steps. Here, feature selection first selects only one submodel candidate for validation by ANN training involving only the variables of the proposed submodel. Is the ANN training successful, the submodel is accepted and removed from the free pool of unidentified submodels, and the rest of the learning parameters is processed again by a feature selection procedure, delivering a modified submodel candidate set which is again tested by a corresponding ANN training.

3.1.4   The user's perspective. For practical application, by non-expert users as well, a software package was developed which, upon supplying measurement data in a suitable form, automatically determines and learns the net of submodels of a complex system. As a prerequisite, the user has to provide the following:

(i) A sufficiently large data set, e. g. a database table, columns meaning the system variables and each row standing for these variables recorded at a given time.

(ii) Since in subsequent parts of the algorithm, an ANN is employed to test whether a given variable can be estimated using other parameters, a maximal tolerable error has to be assigned to each variable when estimating it with an ANN model.

Is a data set supplied in the above form, the submodel decomposition algorithm can be started, which delivers the following results:

(i) A set of valid submodels, with as many of the parameters labeled as output as the ANN algorithm could find.

(ii) A set of rejected submodels judged invalid by the ANN algorithm. Storing these is useful for an early pruning of submodel candidates bound to fail. Also, this list can be used to reject hypotheses concerning the analysed system.

(iii) Since valid submodels were spotted as ANN's were learning their dependencies, this knowledge is readily accessible as a network separate neural nets, each of them representing one submodel.

Figure 2 shows a screenshot of an actual industrial application in a rather low level of manufacturing where a part of a production line is modeled using more than sixty parameters.

3.2 The place of submodel decomposition among related and preceding methods

Having presented our proposed submodel decomposition technique, let us examine which classes it belongs to and which are some of its forerunners and related methods.

- As for modeling, submodel decomposition relies on *ANNs*, more specifically, *multilayer perceptrons* (MLPs). A modified form of the accelerated backpropagation method *SuperSAB* (see Tollenare 1990) is used for training the MLPs.

- The model building method can be looked upon as a special case of learning algorithms as well. The fact that no predetermined layout of input and output parameters is required, classifies it as an *unsupervised learning* algorithm. This is a noteworthy achievement as here, feed-forward ANN structures are created with unsupervised learning. *Moreover, the resulting networks can be tuned with both supervised and unsupervised learning later on.*

- Modeling of *many-valued mapping* is solved by the introduced algorithm as well. Brouwer and Pedrycz (2003) addressed a similar problem with a totally different approach before aiming at handling incomplete data sets—similarly, by coincidence, to the authors of the submodel finding method presented here (see Viharos *et al.* 2002).

- Several approaches are known in literature which, also for the case of MLPs, can alter the structure of the networks to improve them, usually adding or deleting neurons whenever needed for further training (see Yasui 1997). The *network of (interconnected) networks* arrangement built in submodel decomposition is actually the result of a process of this kind, and can be considered a special result of a *structure determination process* of ANNs, as well as the product of a specific class of combined *pruning-learning* methods.

- The application of ANNs is often preceded by some kind of preprocessing of raw learning data. This consists, especially in manufacturing (see Viharos and Monostori 1999c, Erdélyi and Hornyák 2003), in picking out parameters that "are worth learning," i.e. in *feature selection*. This is also practiced in our submodel decomposition method, however, as opposed to the traditional separation of feature selection and *subsequent* ANN training (see Egmont-Petersen *et al.* 1998), the submodel identification approach presented here *combines feature selection and ANN training in alternating steps*, resulting in a hybrid technique.

4. APPLICATION IN HIGH AND LOW LEVELS OF PRODUCTION CONTROL

Submodel decomposition, as well as the fixed-topology ANNs applied therein, found application in various levels of industrial production. The following examples show application cases of the techniques that led up to submodel decomposition—ranging from feed-forward ANN application over ANN-model-based problem solving and optimization to submodel decomposition. These cases show the versatile applicability of the techniques in various levels of manufacturing which may open a broad scope of research to transfer these—and related—techniques to other levels of production.

4.1 Low level application examples

Classical modeling. Application of "plain" modeling with fixed-topology ANNs in lower levels of production is shown by Viharos *et al.* (2003) for simulating chip formation in turning and milling processes and for modeling surface quality properties of a cutting process depending on various technical parameters whose choice may change from task to task.

Problem solving. The ANNs obtained through input/output search, without the decomposition to submodels, can be used to solve a variety of estimation problems. Numerous practical application examples for low levels of production (metal cutting again, as in the examples mentioned before) are given by Viharos and Monostori (1999c). Another application example for lower levels of manufacturing, Viharos *et al.* (2002b) shows another case for multiple solutions of a non-invertible relation. In Viharos *et al.* (2002b), the influence of the simultaneous selection of several requirements is examined from the point of view of estimation accuracy.

Optimization. With the help of the ANNs, a solution can also be found using iterative optimization. An application example of multipurpose optimization with constraints is shown by Viharos and Monostori (2001) where *simulated annealing* is used to obtain a set of valid solutions to manufacturing problems.

Submodel decomposition. Figure 2 shows a screenshot of an actual industrial application in a rather low level of manufacturing where a production line is modeled using more than sixty parameters.

4.2 High level application examples

Classical modeling. Application of ANNs for modeling higher levels of production was proposed by Monostori *et al.* (2001), where an improved ANN modeling concept is extended to process chains and entire production plants.

Problem solving. An example for the use of the generic ANN model in higher levels of production is given by Viharos and Monostori (2001) where various problems related to efficiency improvement had to be solved in manufacturing processes of multilayer printed circuit boards.

Optimization. Constrained optimization techniques initially applied to only one production step are extended by Viharos and Monostori (2001) to a higher level of production: The block-oriented *ProcessManager* framework presented by Viharos and Monostori (2001) can deal with an entire process chain where the result of an earlier step may influence all subsequent steps. Even higher levels of production are handled by Monostori *et al.* (2001) where a hybrid optimization technique (supported by AI, machine learning (ML) and simulation) is used to optimize the arrangement of manu-

facturing processes within a production plant. A substantial gain in optimization time (acceleration by a factor of 6000) is reached by substituting discrete event simulation with ANNs trained by results of earlier simulation runs.

Submodel decomposition. The identification of submodels in a complex system was also applied in an intermediate level of production (see Viharos *et al.* 2003b).

5. FURTHER RESEARCH

Currently ongoing research activities are aimed at extending submodel decomposition towards an agent-based framework where knowledge specific to an agent is mapped onto a given submodel. The feasibility of this generalization is assumed because of remarkable analogies between ANN-based submodels and agents: the fact of decomposability, the existence of localized knowledge with strongly limited connections beyond a given neighborhood, a network architecture, learning or adaptive behavior and estimation or prediction abilities. Additionally to the submodel principle, the automatic decomposition approach itself is expected to be applicable to autonomous agents as well, moreover, agents could be dynamically set up, grouped or split up according to various efficiency criteria, such as learning ability or skills of predicting relevant events. It is envisaged that such a multi-agent system can be erected as a higher level envelope for lower level production control to determine an efficient initial layout of entire production plants or provide decision support for their reorganization.

6. CONCLUSION

The first part of this paper highlighted fundamental phenomena equally shared by higher and lower levels of manufacturing (complexity due to a dense network of interdependencies, and a large number of relevant system parameters).

To handle these, a submodel decomposition method, integrating generalized feature selection and ANN training, was presented in the second part of the paper. The versatility of the method was demonstrated by examples of practical use for various levels of manufacturing systems. The results obtained so far suggest that the approach, if proven successful for one production level, can be modified for use in other levels, such new experience possibly bringing mutual improvement in all domains of application.

Finally, as a concluding remark to the method, future research plans of combining submodel decomposition with a flexible multi-agent system were outlined.

ACKNOWLEDGEMENT

REFERENCES

Brouwer, R.K., Pedrycz, W., Training a feed-forward network with incomplete data due to missing input variables. Applied Soft Computing vol. 3, 2003 23–36

Devijver, P.A., Kittler, J., eds.: Pattern recognition, a statistical approach. Prentice-Hall International Inc, England, London, 1982

Egmont-Petersen, M., Talmon, J.L., Hasman, A., Ambergen, A.W., Assessing the importance of features for multi-layer perceptrons. Neural Networks vol. 11, 1998 623–635

Erdélyi, F., Hornyák, O.: Advanced simulation of nc turning operations. In: Production Systems and Information Engineering. Miskolc University Press, 2003, 41–53

Ghiassi, M., Saidane, H., A dynamic architecture for artificial neural networks. Neurocomputing, vol. 63, 397–413

Horváth, M., Markos, S., Gépgyártástechnológia. Univ. lecture notes, Technical University of Budapest, 1995

Luttervelt, C.A., Childs, T.H.C., Jawahir, I.S., Klocke, F., Venuvinod, P.K., Present situation and future trends in modelling of machining operations. In: Progress report of the CIRP working group "Modelling of Machining Operations", CIRP STC cutting keynote paper, 1998 1–47

Merchant, M.E., An interpretive look at 20th century research on modelling of machining (inagural address). Proc. of the CIRP International Workshop on Modelling of Machining Operations, 1998, 27–31

Monostori, L., Viharos, Zs.J., Markos, S., Satisfying various requirements in different levels and stages of machining using one general ANN-based process model. In: Proc. of the 15th International Conference on Computer-Aided Production Engineering, CAPE'99, 1999 477–484

Monostori, L., Márkus, A., Brussel, H.V., Westkämper, E., Machine learning approaches to manufacturing. CIRP Annals vol. 45, 1996, 675–712

Monostori, L., Viharos, Zs.J.: Hybrid, AI- and simulation-supported optimisation of process chains and production plants. CIRP Annals vol. 50, 2001, 353–356

Tollenare, T.: SuperSAB: fast adaptive backpropagation with good scaling properties. Neural Networks vol. 3, 1990, 561–573

Tóth, T., Tervezési elvek, modellek és módszerek a számítógéppel integrált gyártásban. Univ. lecture notes, Publishing House of the University of Miskolc, 1998

T. Tóth, I.D., On a new theoretical approach of integration. In: Information Control Problems in Manufacturing Technology, INCOM89, 1989, 273–279

Viharos, Zs.J., Monostori, L., Markos, S., A framework for modelling, monitoring and optimisation of manufacturing processes and process chains by using machine learning and search algorithms. In: Proceedings of the 9th IMEKO TC-10 International Conference on Technical Diagnostics, 1999, 249–254

Viharos, Zs.J., Monostori, L.: Automatic input-output configuration of ANN-based process models and its application in machining. In: Lecture Notes of Artificial Intelligence - Multiple Approaches to Intelligent Systems, Conference, Cairo, Egypt. (1999) 659–668

Viharos, Zs.J., Monostori, L., Optimisation of process chains and production plants using hybrid, AI- and simulation based general process models. The Fourteenth International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems, Book: Lecture Notes of Artificial Intelligence, Springer Computer Science Book, Springer-Verlag Heidelberg, 2001, 827–835

Viharos, Zs.J., Monostori, L., Vincze, T., Training and application of artificial neural networks with incomplete data. In: Proc. of the 15th International Conference on Industrial & Engineering Application of Artificial Intelligence & Expert Systems, 2002 649–659

Viharos, Zs.J., Novák, K., Tóth, G., Markos, S., Modelling of different aspects of the cutting process by using ANNs. In: Proceedings of the XIII. Workshop on Supervising and Diagnostics of Machining Systems – Open and global manufacturing design (CIRP), 2002, 241–249

Viharos, Zs.J., Markos, S., Szekeres, C., ANN-based chip-form classification in turning. In: Proceedings of the XVII. IMEKO World Congress – Metrology in the 3rd Millennium, 2003, 1469–1473

Viharos, Zs.J., Monostori, L., Novák, K., Tóth, G., Csongrádi, Z., Kenderesy, T., Sólymosi, T., Lőrincz, Á.., Koródi, T., Monitoring of complex production systems, in view of digital factories. In: Proceedings of the XVII. IMEKO World Congress – Metrology in the 3rd Millennium, 2003, 1463–1468

Viharos, Zs.J.: Automatic generation of a net of models for high and low levels of production. In: Proc. of the 16th IFAC World Congress., 2005 Reg. No. 05127

Yasui, S., Convergence suppression and divergence facilitation: Minimum and joint use of hidden units by multiple outputs. Neural Networks vol. 10, 1997 353–367

Zhang, Q., Rong, G., Industrial application of data reconciliation for hybrid systems. In: Proc. of the 16th IFAC World Congress, 2005, Reg. Nr. 01903