Elman Networks for the Prediction of Inventory Levels and Capacity Utilization

F. Harjes, B. Scholz-Reiter, A. Kaviani Mehr

***Abstract*—** Today´s production processes face an increase in dynamics and complexity. Therefore, production control techniques face a demand for continuous advancement. Methods from the field of artificial intelligence, such as neural networks, have proven their applicability in this area. They are applied for optimization, prediction, classification, control and many other production related areas. This paper introduces an approach using Elman Networks for the workstation-specific prediction of inventory levels and capacity utilization within a shop floor environment. It includes the selection of the appropriate network architecture, the determination of suitable input variables as well as the training and validation process. The evaluation of the proposed approach takes place by means of a generic shop floor model.

***Keywords*—**Artificial neural networks, Elman networks, predictive control, shop floor production

I. Introduction

Multi variant and customized products with short lifecycles are typical for today`s market [1]. The corresponding production processes and material flows are often complex and dynamic. Consequently, established production planning and control (PPC) approaches need a continuous advancement [2] [3].

Particularly in the field of shop floor production, prototypes and small series as well as the specific technical organization complicate the handling of control related tasks [4]. At this point, methods from the field of artificial intelligence, such as neural networks, have proven their applicability as methods for classification, pattern recognition or production control [5], [6], [7].

This paper introduces an approach of a neural network based prediction of inventory levels and capacity utilization for workstations within a shop floor environment. The approach can be seen as a contribution to the development and implementation of innovative decentralized and/or predictive control strategies [8].

Manuscript received June 15, 2011: Revised version July 12, 2011. This work was supported by the German Research Foundation (DFG) as part of the project "Automation of continuous learning and examination of the long-run behavior of artificial neural networks for production control", index SCHO 540/16-1.

Dipl.-Inf. F. Harjes is with the BIBA – Bremer Institut für Produktion und Logistik GmbH at the University of Bremen, Hochschulring 20, 28359 Bremen, Germany (phone: +49 (0) 421/218 – 5627, fax: +49 (0) 421/218 – 5640, e-mail: haj@biba.uni-bremen.de).

Prof. B. Scholz-Reiter is with the BIBA – Bremer Institut für Produktion und Logistik GmbH at the University of Bremen, Hochschulring 20, 28359 Bremen, Germany (e-mail: bsr@biba.uni-bremen.de)

M. Sc. A. Kaviani Mehr studied production engineering at the University of Bremen, Bibliothekstraße 1, 28359 Bremen (e-mail: amir.kavianimehr@uni-bremen.de).

At this, the structure of the paper is as follows. The next section introduces the special production form shop floor, followed by a short examination of predictive control in Section 3. Section 4 presents neural networks in general, followed by a brief description of the newly developed neural predictors regarding their structure and training results in section 5. Section 6 presents the shop floor model for the evaluation of the new predictors and the obtained experimental results. Finally, the article closes with a conclusion that summarizes the obtained results and gives an outlook on future research in section 7.

II. Shop Floor Production

The prediction concept presented in this paper refers to a shop floor scenario. Shop floor production is characterized by a customer oriented production of single pieces, prototypes and small series with correspondingly small lot sizes [9] [10].

Organizationally and spatial, shop floor manufacturing is divided into several specialized workshops such as a sawmill or a turnery [11] (Fig. 1). Workpieces can pass the different workshops in any order, depending on their individual machining sequence.

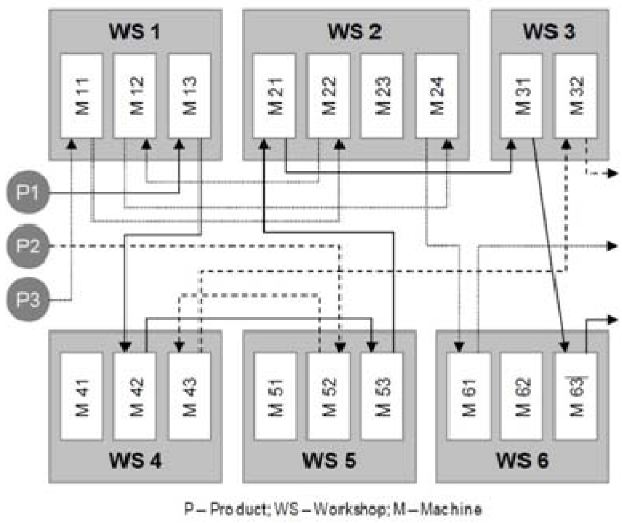


Fig. 1 Shop floor organization **[12]**

This leads to a high flexibility, with a fast adaption to changing situations and disturbances, such as machine downtimes, e.g. [9]. Unfortunately, this also results in a dynamic material flow and complex dependencies between machining, transportation and handling steps [4]. As this conditions are difficult to handle for established production planning and control approaches, PPC systems need a continuous advancement to furthermore enable an efficient handling [13]. One approach in this field is the implementation of predictive control strategies.

III. Predictive Control

Predictive control systems basically rely on the prediction of the control variables` future development [14]. Predictive control is also known as "model predictive control" (MPC) or "model based predictive control" (MBPC) [15], [16]. For this, a model of the controlled system acts as a kind of function to compute the system outputs from the system inputs [17]. The considered time period shifts along the time axis and has a range of N sampled time steps (Fig. 2, upper half).

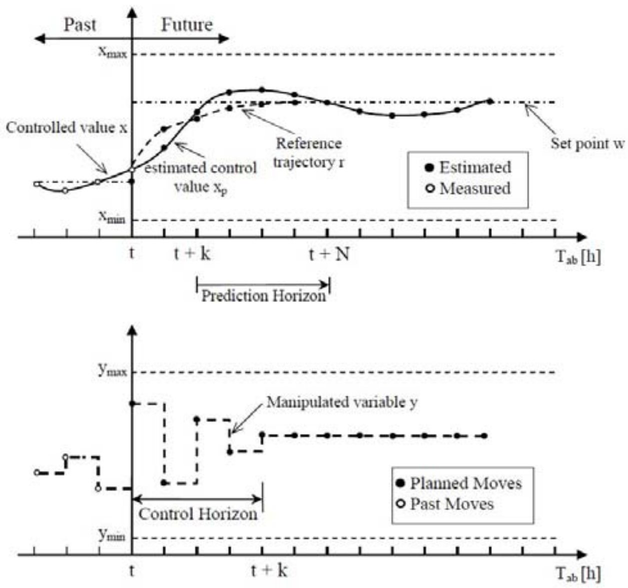


Figure 2 Principle of predictive control **[18]**

Correspondingly, the prediction horizon ends at t + N time steps, starting from the current time t. The number of time steps k, the control structure covers, denotes the control horizon t + k (Fig.2, bottom half). This period is usually shorter than the prediction horizon [15].

From the process and the hardware perspective, the classic control loop is extended with a prediction component (Fig. 3).

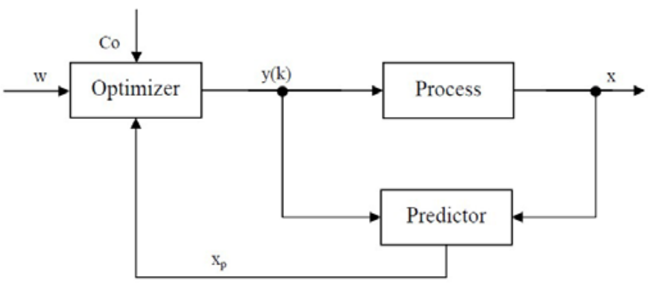


Fig. 3 Predictive control loop (simplified)

Within this predictive control loop, the controller (here called optimizer) processes the future course of the set point w, the constraints C_o and predicted value of the control variable x_p [19]. The result of the calculation is a series of optimal manipulated variables y. Their first element y (k) enters the controlled system as actual control variable. At this, the prediction bases on the actual values and the settings y_k of the previous control cycle [14] [20].

The technical implementation of predictive control approaches is feasible through a number of technologies such as fuzzy logic, artificial neural networks or agent based approaches [21] [22].

IV. Artificial Neural Networks

Artificial neural networks emulate the structure and functionality of neural systems in nature [23]. They typically consist of nodes, which are arranged in at least two or more layers and are interconnected via weighted links [24] (Fig. 4). At this point, the number of layers and the direction of the connections depend on the type of network [25].

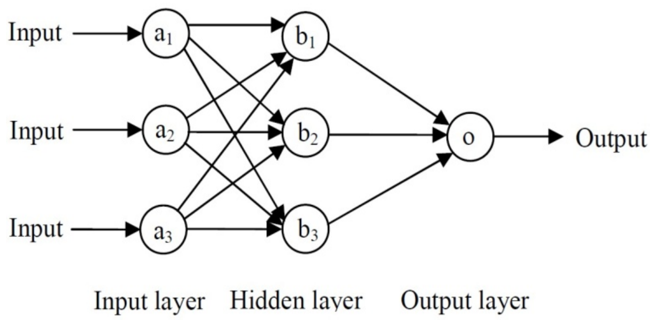


Fig. 4 Example of a neural network

The nodes of a neural network act as a kind of neural processor [23]. In general, the sum of the input values serves as calculation basis for the so called activity function [26]. Common activity functions are the sigmoid or the tangens hyperbolicus [27]. The activity value is either directly transmitted to the subsequent nodes or a special output function calculates the output value based on the activity. It is also possible to choose the identity function for the output calculation. In this case, the output also corresponds to the activation [23].

Neural networks offer a fast data processing, a comparatively small modelling effort and the ability to learn from experience [28]. Further, they are able to approximate complex mathematical coherences that are either unknown or not completely describable [29]. In order to do so, neural networks act in a black box manner [30].

Depending on the type of neural network, three general learning procedures can be distinguished. Supervised Learning denotes a procedure, where pairs of input and output data are presented to the neural network. During the learning process, the network adapts its connection weights, so that the input leads to the desired output [25]. Reinforcement Learning only comprises the presentation of input data. Instead of the corresponding output, the network receives a feedback, whether the output was correct [23]. Finally, Unsupervised or Self-Organized Learning takes place without any default values for the output or the corresponding feedback. At this point, the neural network tries to recognize patterns within the input data autonomously [31].

Common for all approaches is the validation of the learning results with a second dataset. This ensures the generalization of the learning process and avoids a mere memorization of the training data, the so called Overfitting [23].

V. The Neural Predictors

A. Elman Networks

As mentioned above, the structure of a neuronal network strongly depends on the application area. For prediction purposes, recurrent or partly recurrent architectures are common [32]. But in individual cases, other network types were successfully adapted to prediction related tasks.

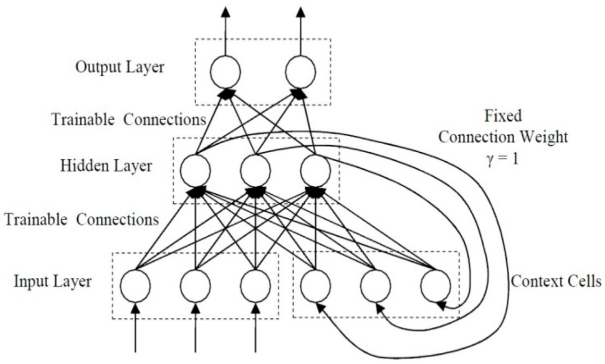


Fig. 5 Elman Network **[26]**

In 2008 for example, Hamann introduced an intelligent inventory-based production control system using neural networks [14]. Within his approach, feed-forward networks come into operation both for control and for prediction. According to Hamann, the training effort of feed-forward networks is lower than the one of other network architectures in this field. In contrast, the prediction quality is only average, with a double-digit error for a prediction horizon of 7 days. Experiments with a longer horizon of 21 days show an unacceptable error rate.

With regard to Hamann`s results, the approach presented in this paper focuses on Elman networks, a partially recurrent network architecture [33]. Elman networks are feedback networks, containing a special layer of so called context cells [34] (see Fig. 5).

These context cells save the neural activation of previous states and therefore ensure that the prediction takes past events into account. Thus, the connection weight between the hidden layer and the context cells determines how much past states influence the prediction. A connection weight near or equal to 1 stands for a strong influence of past states, a smaller value mitigates this effect. The general concept of Elman networks is extendable to topologies with multiple hidden layers. These networks contain context cells for each present hidden layer and are called hierarchical Elman networks [26].

B. Structure of the Neural Predictors

The proposed concept comprises the workstation-specific prediction of inventory level and capacity utilization. For this purpose, the neural networks consider the actual state of the regarded workstation as well as the conditions of the predecessors. Correspondingly, the predictor networks` topology depends on the position, the considered workstation has within the material flow.

In the following, a workstation with two predecessors serves as an example. The neural predictor for the inventory level is a 5:10:10:1 Elman Network (Fig.6). It processes 5 input values, these are:

1) The actual inventory level of workstation *n*, manufacturing stage *m* at time *t* (Inventory $(t)_{n,m}$),

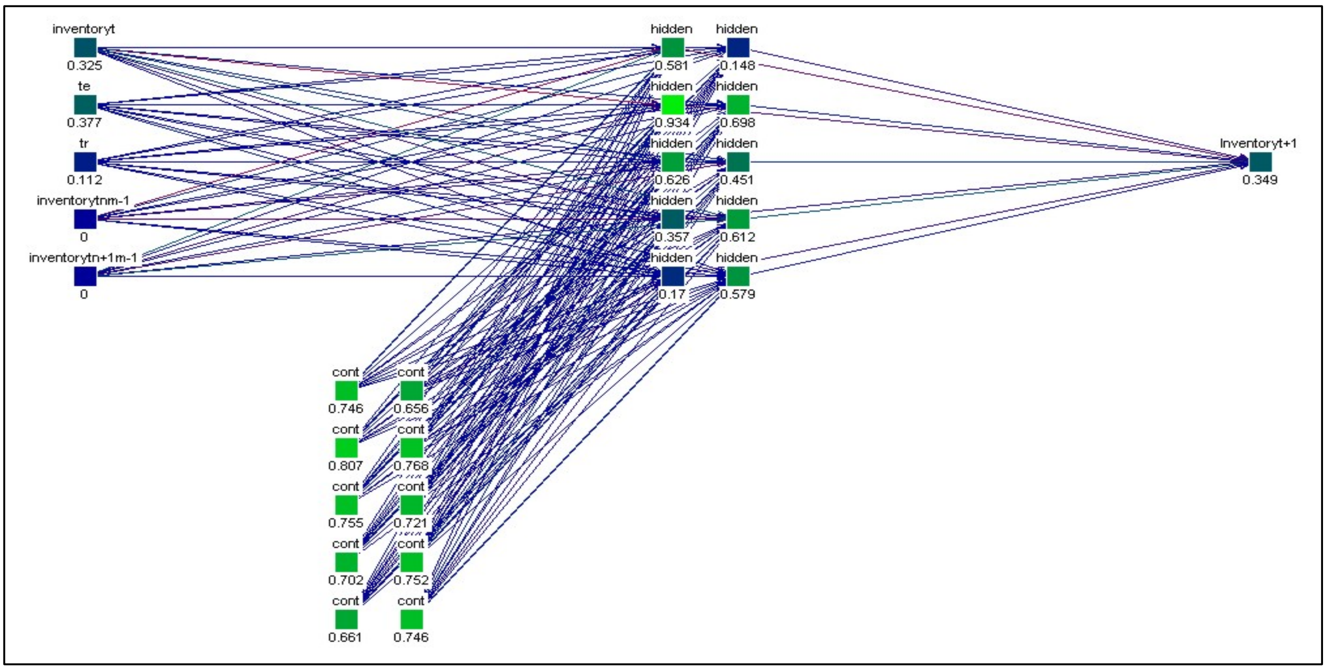
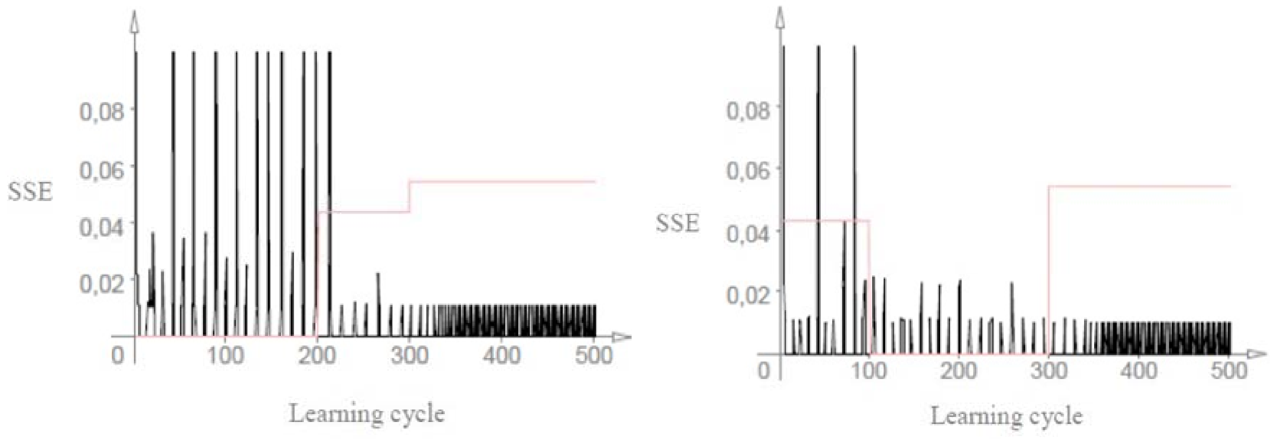


Fig. 6 Topology of the inventory predictor (screenshot)



(a) (b)

Fig. 7 Exemplary training results; (a) Quickprop (b) Backpropagation with Momentum term

2) the machining time ($te_{n,m}$) and

3) the setup time ($tr_{n,m}$)of all orders waiting in front of the workstation,

4) the actual inventory level of predecessor *n,* production stage *m-1* at time *t* (Inventory$(t)_{n,m-1}$),

5) the actual inventory level of predecessor *n+1*, production stage *m-1* at time *t* (Inventory $(t)_{n.m-1}$.

The output value of the network represents the predicted inventory level at time t+1. At this point, the prediction horizon amounts four hours, depending on the shift plan of the underlying shop floor model.

The capacity predictor has a similar 4:10:10:1 topology. While the number of hidden neurons and context cells is identical, the network needs only four input neurons. These neurons process the following values:

1) The capacity of workstation *n*, production stage *m* at time *t* (Capacity $(t)_{n,m}$),

2) the occupancy of workstation *n*, production stage *m* at time *t* (Occupancy $(t)_{n,m}$),

3) the current inventory level of workstation *n,* production stage *m* at time *t* (Inventory $(t)_{n,m}$) and

4) the waiting time of workstation *n*, production stage *m* at time *t* (Waiting $(t)_{n,m}$).

At this point, capacity defines the maximum number of workpieces that can be produced within the prediction horizon of 4 hours (half a work shift). The determination of the corresponding period length is described in section 4. Finally, the waiting time denotes the amount of time, the workstation pauses due to disturbances, breaks, etc.

C. *Training and Validation*

The initial training and validation process of both prediction networks is carried out using the Java Neural Network Simulator (JNNS), a Java based simulation platform [35]. This simulation program is the successor of the Stuttgart Neural Network Simulator (SNNS) that comes into operation in the experimental validation (see section 6) [36].

The neural predictors` training process uses the supervised learning method following the Resilient Propagation algorithm. Previous Experiments with other training algorithms, such as Quick Propagation and Backpropagation with Momentum term show inadequate results. Figure 7 depicts two exemplary results from these experiments, covering 500 training cycles each. The lower line represents the results (summed square error) of the training dataset, while the upper line denotes the same for the validation data.

Regarding the learning and training curves, both learning algorithms show an inadequate learning behavior. For the Quickpropagation approach (Fig. 7(a)), the training curve oscillates during the whole learning process. At this, the prediction error is between 100 % for the first 200 cycles and 10 to 20% for the 300 following cycles. Further, the corresponding validation curve is nearly zero during the first 200 cycles and skips in two steps to a prediction error of almost 60% for the remaining 300 cycles.

The Backpropagation algorithm with Momentum term also leads to oscillation training and validation curves with inadequately high prediction errors (Fig. 7(b)). In Contrast to the Quickpropagation approach, Backpropagation reaches error levels between 20 and 40% with three high peaks reaching an error of 100%. The validation data leads to an error of 40% for the first 100 cycles and 50% for the last 200 cycles. Between these two peaks, the neural network reaches an error of 0 %.

These results can be reduced to the inner structure of the datasets used for learning. Obviously, both learning methods are not able to determine a suitable weight matrix for the network. As mentioned above, the Resilient Propagation algorithm obtains adequate results and therefore comes into operation for the following experiments.

The necessary learning and validation datasets result from test runs of the shop floor model that is also used for evaluation purposes in the next section. The test runs take approximately 30 days with an average of 1770 orders. At this point, the recording of input/output pairs takes place every four hours. Fig. 8 depicts the learning curve of the network for capacity prediction. The training process converges after approximately 700 cycles, when both curves reach their minimum.

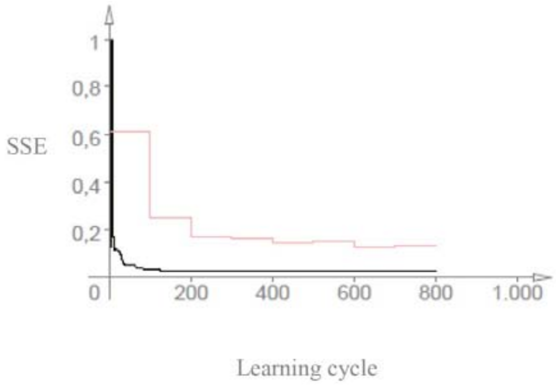


Fig. 8 Learning process of the capacity predictor

A further training would lead to an increasing error for the validation data and a slight improvement for the initial training set. This is a typical indication for an overfitting of the neural network [36].

The minimal error during the training process is less than 0,1 (1≈100%). Transferred to the original prediction task, this implies an average prediction error of approximately 5%. The learning process of the inventory predictor converges after approximately 400 cycles (Fig. 9). At this point, the minimal error is again less than 0,1, but slightly higher than the capacity predictor`s result.

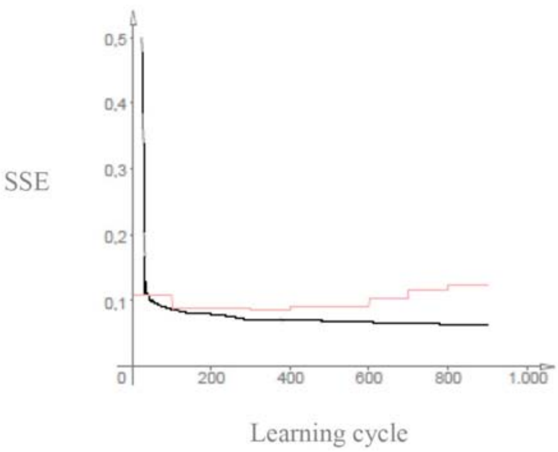


Fig. 9 Learning process of the inventory predictor

VI. Experiments

A. Settings

The evaluation of the neural predictors takes place by means of a generic shop floor model. As software platform, the material flow simulation "Plant Simulation" comes into operation [37]. The Plant Simulation model comprises eight workstations on four production stages (Fig 10). Every workstation has an input buffer in front of it. The workpieces pass the buffer following the FIFO principle (First-In-First-Out). The shop floor operates in three shifts of eight hours each. To enable a quick reaction to changing production situations, the prediction horizon is set to the half of a shift (four hours).

During the simulated period of 30 days, six different workpiece types run through the shop floor. The order release takes place piecewise the setup and processing times differ for every type of workpiece, depending on the technical properties of the workstations. Hence the processing and setup times are in the range of one up to 40 minutes.

The processing order is sequential, so that every workpiece passes all four production stages. The distribution of workpieces between the production stages follows an inventory based control approach. A finished workpiece is always transferred to the successor at the following production stage with the comparatively lowest inventory level.

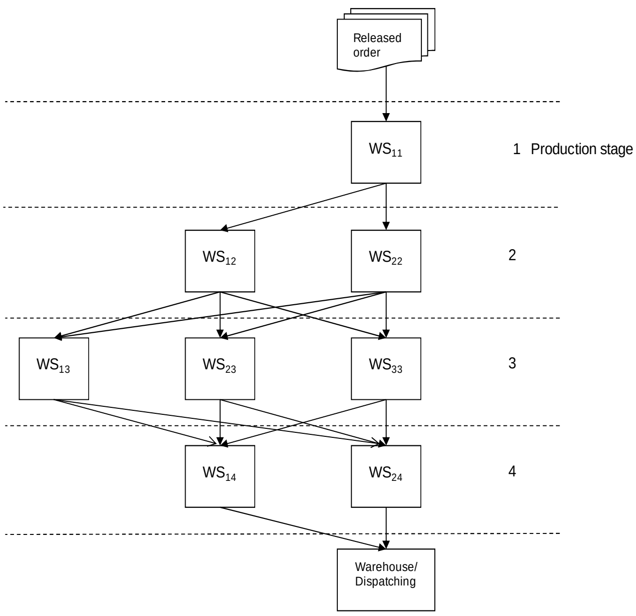


Fig. 10 Layout of the shop floor model

While the shop floor model runs in Plant Simulation, the simulation of the neural predictors takes place by means of the Stuttgart Neural Network Simulator (SNNS), a C++ based simulation platform for neural networks [38]. The connection to the shop floor model in Plant Simulation is implemented via network (Ethernet), using the TCP/IP protocol. For this, the data flow is as follows.

The input data for the neural networks is recorded within Plant Simulation and send via a TCP/IP socket to the running

SNNS instance. The answer contains the prediction results of the networks.

B. *Results*

In the following, the prediction results of workstation ws_{13} serve as an example for the whole shop floor. This workstation is located at production stage 3 and has two predecessors as well as two successors.

Figure 11 depicts the comparison between the actual and the predicted capacity utilization for this workstation over a period of 20 hours. This timeframe contains five predictions with a horizon of four hours each. At this point, the curve for the actual values represents continuously recorded data. The prediction curve depicts an approximation between the performed five predictions. This results in a relatively uneven curve shape.

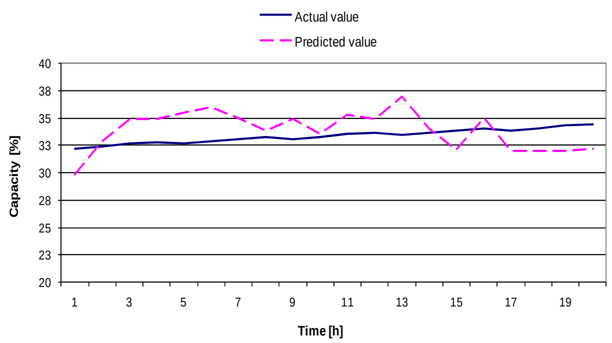


Fig. 11 Actual and predicted capacity utilization for WS_{13}

The evaluation further shows an average workload scarcely above 34%. The time of inactivity is attributable to disturbances, breaks, setup times and maintenance. The predicted capacity utilization is close to the actual data, with a deviation of 3.2% maximum (Fig.12).

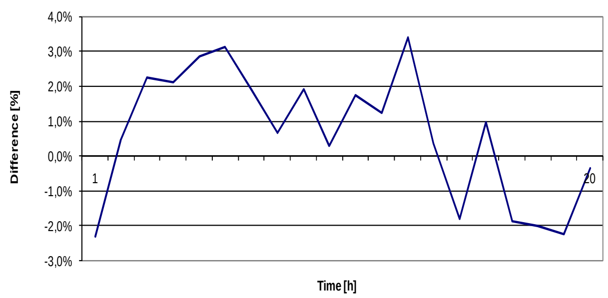


Fig. 12 Deviation of the prediction error for the inventory levels

The course of the inventory prediction is similar, with an error between nearly zero and a maximum of approximately 6% (Fig. 13). As it is for the capacity prediction, the actual values represent continuous and event-oriented data. In contrast, the predicted values depict an approximation of the inventory development.

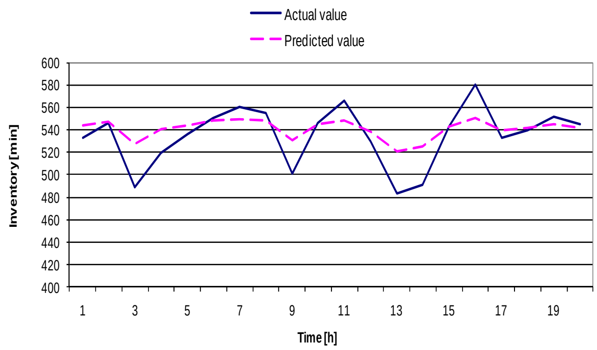


Fig. 13 Actual and predicted inventory level for WS_{13}

The predicted values differ from the real inventories averagely 2.5% (Fig. 14). Nevertheless, the prediction deviates up to 40 minutes from the recorded inventory level. Due to the setup and processing times, deviation can correspond to 1-4 workpieces.

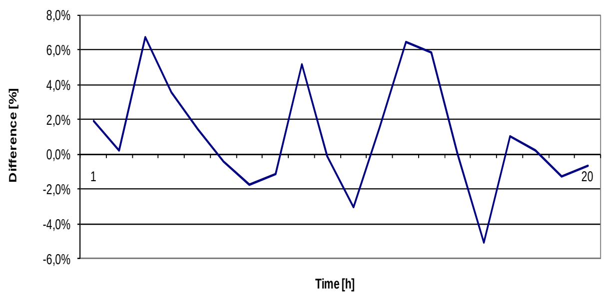


Fig. 14 Deviation of the prediction error for the capacity utilization

VII. Conclusion

This paper introduces an approach for the workstation-specific prediction of capacity utilization and inventory levels in a shop floor environment using partially recurrent Elman networks. The experimental results render a low monadic prediction error with a maximum of 6% for a prediction horizon of four hours. This is sufficient in the case of capacity utilization. For the inventory levels, an even more precise prediction is desirable. At this point, the deviation between the real and predicted values can correspond to multiple workpieces.

Therefore, future research should focus on the reduction of prediction errors in coordination with an increase of the prediction horizon. A possible starting point is the evaluation of other network architectures or topologies. Another point of interest should be the practical integration of the introduced prediction approach into modern production control strategies, e.g. Model Predictive Control (MPC). Further, the preparation of training and validation data should be systemized, as the choice of an adequate training method is difficult and often based on a trial and error proceeding.

In the field of neural network research, there is a fundamental interest in making continuous adaptations to changing shop floor situations, such as shifting setup- and processing times and the varying number of workpiece types. At this point, the long-time application of neural networks in practical environments is an important field. The remaining question is now: Is it possible to implement a continuously learning production control system using neural networks?

REFERENCES

[1] J. Barata and L. Camarinha-Matos, "Methodology for Shop Floor Reengineering Based on Multiagents," in *IFIP International Federation for Information Processing - Emerging Solutions for Future Manufacturing Systems*, L. Camarinha-Matos, Ed. Boston: Springer, 2005, vol. 159, pp. 117-128.

[2] W. Schäfer, R. Wagner, J. Gausemeier, and R. Eckes, "An Engineer's Workstation to Support Integrated Development of Flexible Production Control Systems," in *Integration of Software Specification Techniques for Applications in Engineering*, vol. 3147/2004, Berlin Heidelberg, 2004, pp. 48-68.

[3] I.I. Siller-Alcalá, J. Jaimes-Ponce, and Alcántara-Ramírez, "Robust Nonlinear Predictive Control," in *Proceedings of the 7th WSEAS international conference on System science and simulation in engineering*, Venice, 2008, pp. 164-167.

[4] B. Scholz-Reiter, C. Toonen, and D. Lappe, "Job-shop-systems: continuous modeling and impact of external dynamics," in *Proceedings of the 11th WSEAS international conference on robotics, control and manufacturing technology, and 11th WSEAS international conference on Multimedia systems and signal processing ROCOM'11/MUSP'11*, Venice, 2011, pp. 87-92.

[5] B. Scholz-Reiter, F. Harjes, J. Mansfeld, T. Kieselhorst, and J. Becker, "Towards a Situation Adaptive Shoop Floor Production," in *Proceedings of the Second International Conference on Business Sustainability 2011*, Guimarães, Porto, 2011, pp. 1-8.

[6] B Scholz-Reiter, T Hamann, H Höhns, and G. Middelberg, "Decentral Closed Loop Control of Production Systems by Means of Artificial Neural Networks," in *Proceedings of the 37th CIRP - International Seminar on Manufacturing Systems*, Budapest, Hungary, 2004, pp. 199-203.

[7] J. Rutkowski and D. Grzechca, "Use of artificial intelligence techniques to fault diagnosis in analog systems," in *Proceedings of the 2nd conference on European computing conference*, Malta, 2008, pp. 267-274.

[8] B. Scholz-Reiter and T. Hamann, "The behaviour of learning production control," *CIRP Annals - Manufacturing Technology*, vol. 7, no. 1, pp. 459-462, 2008.

[9] M., Säfsten, K. Bellgran, *Production Development – Design and Operation of Production Systems*. London: Springer Verlag, 2010.

[10] T. Gudehus and H. Kotzab, *Comprehensive Logistics*. Berlin: Springer Verlag, 2009.

[11] B. Scholz-Reiter, F. Harjes, and D. Rippel, "An Architecture for a Continuous Learning Production Control System based on Neural Networks," in *7th CIRP Int. Conference on Intelligent Computation in Manufacturing Engineering – CIRP ICME '10*, Capri, Italy, 2010.

[12] H.C. Pfohl, *Logistiksysteme: Betriebswirtschaftliche Grundlagen*. Berlin: Springer Verlag, 2010.

[13] B. Scholz-Reiter, M. Freitag, A. Schmieder, A. Pikovsky, and I. Katzorke, "Modelling and Analysis of a Re-entrant Manufacturing System," in *Nonlinear Dynamics of Production Systems*, G, Radons and R. Neugebauer, Eds.: Wiley-VHC, 2004, pp. 55-69.

[14] T. Hamann, *Lernfähige intelligente Produktionsregelung*, B. Scholz-Reiter, Ed. Berlin: Gito Verlag, 2008, vol. 7.

[15] R. de Keyser, "The MBPC approach," in *Proceedings CIM-Europe Workshop on Industrial Applications of Model Based Predictive Control*, Cambridge, 1992.

[16] R. de Keyser and C.M. Ionescu, "The disturbance model in model based predictive control," in *Proceedings of 2003 IEEE Conference on Control Applications*, vol. 1, Istanbul, 2003, pp. 446-451.

[17] J. Vehi, J. Rodellar, M. Sainz, and J. Armengol, "Analysis of the Robustness of Predictive Controllers via Modal Intervals," *Reliable Computing*, vol. 6, no. 1, pp. 281-301, January 2000.

[18] M. Rau, *Nichtlineare modellbasierte prädiktive Regelung auf Basis lernfähigerZustandsraummodelle*, TU München, Ed. München, 2003.

[19] P.S. Agachi, Z.K. Nagy, and M.V. Cristea, *Model Based Control; case study in process engineering*. Weinheim: Wiley-VCH Verlag, 2006.

[20] W. Wendt and H. Lutz, *Taschenbuch der Regelungstechnik*, 6th ed.: Deutsch Harri GmbH, 2005.

[21] J. Jantzen, *Foundations of Fuzzy Control*, 1st ed.: Jon Wiley and Sons, 2007.

[22] D.H. Scheidt, "Intelligent agent-based control," *JOHNS HOPKINS APL TECHNICAL DIGEST*, vol. 23, no. 4, pp. 383-395, 2002.

[23] S. Haykin, *Neural Networks and Learning Machines (3rd Edition)*. New Jersey, USA: Prentice Hall, 2008.

[24] W-H. Steeb, *The Nonlinear Workbook: Chaos, Fractals, Neural Networks, Genetic Algorithms, Gene Expression Programming, Support Vector Machine, Wavelets, Hidden Markov Models, Fuzzy Logic with C++, Java and SymbolicC++ Programs*, 4th ed. Singapore: World Scientific Publishing Co. Pte. Ltd, 2008.

[25] D.K. Chaturvedi, "Artificial neural networks and supervised learning," in *Soft Computing: Techniques and its Applications in Electrical Engineering*. Berlin Heidelberg: Springer, 2008, pp. 23-50.

[26] W-M. Lippe, *Soft-Computing mit Neuronalen Netzen, Fuzzy-Logic und Evolutionären Algorithmen*. Berlin: Springer, 2006.

[27] Y. Bar-Yam, *Dynamics Of Complex Systems (Studies in Nonlinearity)*.: Westview Press, 2003.

[28] G. Dreyfus, *Neural Networks Methodology and Application*. Berlin Heidelberg: Springer Verlag , 2005.

[29] P.M. Fonte, G. Xufre Silva, and J.C. Quadrado, "Wind Speed Prediction using Artificial Neural Networks," in *Proceedings of the 6th WSEAS Int. Conf. on NEURAL NETWORKS*, Lisbon, 2005, pp. 134-139.

[30] D. Rippel, F. Harjes, and B. Scholz-Reiter, "Modeling a Neural Network Based Control for Autonomous Production Systems," in *Artificial Intelligence and Logistics (AILog) Workshop at the 19th European Conference on Artificial Intelligence 2010*, Amsterdam, 2010, pp. 49-54.

[31] T. Kohonen, *Self-Organizing Maps*, 3rd ed. New York: Springer, 2001.

[32] D. Mandic and J. Chambers, *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability (Adaptive and Learning Systems for Signal Processing, Communications and Control Series)*. Hoboken, USA: Wiley-Blackwell, 2001.

[33] J.L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179-211, 1990.

[34] A.A. Akbari, K. Rahbar, and M.J. Mohammadi Taghiabad, "Induction Motor Identification," in *Proceedings of the 5th WSEAS Int. Conf. on Signal Processing, Robotics and Automation*, Madrid, 2006, pp. 153-157.

[35] I. Fischer, F. Hennecke, C, Bannes, and A. Zell. JavaNNS:Java Neural Network Simulator. [Online]. http://www.ra.cs.uni-tuebingen.de/software/JavaNNS/manual/JavaNNS-manual.pdf

[36] S. Lawrence and C.L. Giles, "Overfitting and neural networks: conjugate gradient and backpropagation," in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, vol. 1, Como, Italy, 2000, pp. 114-119.

[37] S. Bangsow, *Manufacturing Simulation with Plant Simulation and Simtalk: Usage and Programming with Examples and Solutions*, 1st ed. Berlin: Springer, 2010.

[38] A. Zell. Stuttgart Neural Network Simulator. [Online]. http://www.ra.cs.uni-tuebingen.de/downloads/SNNS/SNNSv4.2.Manual.pdf

Prof. Dr.-Ing. Bernd Scholz Reiter is managing director of the Bremer Institut für Produktion und Logistik GmbH at the University of Bremen (BIBA) and head of the research center “Intelligent Production and Logistics Systems (IPS)”.

Born in 1957, he studied Industrial Engineering and Management with a specialty in Mechanical Engineering at the Technical University of Berlin. After his doctorate in 1990 on the “Concept of a computer-aided tool for the analysis and modelling of integrated information systems in production companies ", he was an IBM World Trade Post Doctoral Fellow at the IBM T.J. Watson Research Center, Yorktown Heights, NY, USA, in Manufacturing Research until the end of 1991. Subsequently, he worked as a research assistant at the Technical University of Berlin and in 1994 was appointed to the new chair of Industrial Information Technology at the Brandenburg Technical University of Cottbus. From1998 to 2000, he was head of and founder of the Fraunhofer Application Center for Logistics Systems Planning and Information Systems in Cottbus, Germany. Since 2000 he heads the newly created chair of Planning and Control of Production Systems in the Department of Manufacturing Engineering at the University of Bremen. At the Bremer Institut für Produktion and Logistik (BIBA), Prof. Scholz-Reiter works in applied and industrial contract research.

Prof. Scholz-Reiter is a full member of the German Academy of Engineering Sciences, full member of the Berlin-Brandenburg Academy of Sciences, Associate Member of the International Academy for Production Engineering (CIRP), member of the Scientific Society of Manufacturing Engineering, a member of the group of university professors with an expertise on business organization, member of the European Academy of Industrial Management and a member of the Advisory Commission of the Schlesinger Laboratory for Automated Assembly at the Technion - Israel Institute of Technology, Haifa, Israel. He is Vice President of the German Research Foundation.

Prof. Scholz-Reiter is the speaker of the Collaborative Research Centre 637 "Autonomous Cooperating Logistic Processes - A Paradigm Shift and its Limitations," speaker of the International Graduate School for Dynamics in Logistics at the University of Bremen and speaker of the Bremen Research Cluster for Dynamics in Logistics.

Prof. Scholz-Reiter is editor of the professional journals Industry Management and PPC Management, and a member of editorial committees of several international journals.

Dipl.-Inf. Florian Harjes, born in 1981, is a scientific research assistant at the Bremer Institut für Produktion und Logistik GmbH (BIBA) at the University of Bremen.

He received a diploma in computer science from the University Bremen in 2008, where he pursued his thesis “Exact synthesis of multiplexor circuits” at the same year. During this time, he developed a tool for the automated synthesis of minimal multiplexor circuits for a corresponding Boolean function.

In BIBA, Dipl.-Inf. Florian Harjes is in charge of long time simulations of neural networks and the development of a hybrid architecture for the continuous learning of neural networks in production control.

M. Sc. A. Kaviani Mehr, born 1978 finished his studies in production engineering at the University of Bremen in 2011.