CLASSIFICATION OF DYNAMICAL PATTERNS IN AUTONOMOUSLY CONTROLLED LOGISTIC SIMULATIONS USING ECHO STATE NETWORKS

B. Scholz-Reiter[(a)], T. Lensing[(b)], M. Görges[(c)] , L. Dickmann[(d)]

[(a-c)] BIBA GmbH, University of Bremen, Germany
[(d)] Research Group Digital Media, University of Bremen, Germany

[(a)]bsr@biba.uni-bremen.de, [(b)]leg@biba.uni-bremen.de, [(c)]goe@biba.uni-bremen.de, [(d)]dickmann@tzi.de

ABSTRACT
The concept of autonomous control aims at improving the robustness and performance of logistic systems in a dynamical environment. In this context logistic objects are able to make and execute routing decisions autonomously. On the one hand this enables logistics systems to react promptly on dynamic changes and disturbances. On the other hand autonomous control causes an inherent dynamic systems behavior, which depends mainly on decision logic and initial system states. In order to analyze the interplay between these internal dynamics and the overall systems performance, new approaches for describing and classifying the observed behavior are needed. This paper presents an approach, based on Echo State Networks, for identifying and comparing different dynamics, gained by a simulation model of an autonomous controlled production network. It will be shown, that echo-state networks are able to distinguish different patterns of this exemplary autonomously controlled production network.

Keywords: production networks, autonomous control, echo-state networks

1. INTRODUCTION

Production networks are characterized by company or cross company owned facilities, which aim at an integrated planning of geographical dispersed logistic processes and the usage of common resources (Wiendahl and Lutz 2002). In production networks with geographical dispersed production plants, additional tasks for production planning and control (PPC) arise, e.g. the assignment of jobs to plants or the planning of transports. Thus, the coordination of transport and production processes gets more and more important (Sauer 2006). Conventional incremental planning and control methods have shortcomings to cope with these additional tasks (Ivanov 2009). In this context decentralized approaches, e.g. autonomous control, appear to be promising. Autonomous control aims at improving the logistic performance of a system by coping with complexity in a distributed and flexible manner. According to this concept, single objects are able to make and execute routing decisions by themselves (Windt 2006). One can assume that an integrated autonomous control of both, production and transport processes may increase the logistic performance of production networks (Scholz-Reiter et al. 2009b). However, the implementation of autonomous control does not per se lead to the desired dynamical effects. Different dynamics, e.g. oscillating behavior or unpredictable fluctuations were observed for an autonomous controlled production system with varying start parameters (Scholz-Reiter et al. 2007). In order to get a deeper understanding of the underlying dynamics caused by autonomous control, novel methods of analysis, which go beyond a sole comparison of aggregated data like mean values and standard deviations are necessary. This kind of classical evaluation neglects dynamical aspects (e.g., periodicity), which are also of interest in production planning and control. In this context approaches that are able to classify different dynamics can be used to analyze the link between control logic, dynamic system's behavior, and system's performance. In the case at hand, we deal with output from a simulation model in the form of time series data. The output data exhibits characteristic patterns, which we intend to classify. Echo state networks (ESN) appear to be promising for this task. They denote a relatively novel approach from the domain of reservoir computing (RC) where the hidden layer of a recurrent neural network (RNN) is seen as a dynamical reservoir (DR). In contrast to other RNN approaches, the weights of units inside the hidden layer are not trained here. Instead training is performed by linearly combining signals from the hidden layer using linear regression, which is computationally cheap and guaranteed to converge. It has been shown that ESN are well-suited for time series classification due to their inherent short-term memory capabilities.

This paper presents an approach for analyzing time series data gained by discrete event simulations of an autonomously controlled production network. For this purpose an exemplary production network scenario is modeled. The material flow on the shop-floor-level and on the network-level is controlled autonomously by a pheromone based approach. This paper will show that dynamics caused by the pheromone approach can be classified using echo state networks. We present a cascaded network setup consisting of multiple ESNs for classifying different patterns in time series data of dynamics exhibited by the simulation model. Finally, we discuss the results of our classification experiments and outline the implications of our findings for future research.

2. AUTONOMOUS CONTROL

The collaborative research centre 637 "Autonomous cooperating Logistic Processes: A Paradigm Shift and its Limitations", which is founded by German research foundation, gives the following wide definition of autonomous control: "Autonomous control describes processes of decentralized decision-making in heterarchical structures. It presumes interacting elements in non-deterministic systems, which possess the capability and possibility to render decisions independently. The objective of autonomous control is the achievement of increased robustness and positive emergence of the total system due to distributed and flexible coping with dynamics and complexity" (Windt and Hülsmann 2007). According to this the general definition of autonomous control, the concept can be understood as decentralized coordination of intelligent logistic objects and routing through logistic systems by the intelligent objects themselves (Windt et al. 2008). Hence, the main idea of autonomous control is a shift of decision making capabilities from the overall system to single elements. These autonomously acting elements, i.e. intelligent logistic objects, are able to gather information about local system states and to make and execute decisions on the basis of this information. The term intelligent logistic object is broad defined. It covers physical objects (e.g. machines, jobs, etc.), as well as immaterial objects like production orders (Windt 2006). In the context of production logistics the decision making capability is transferred to the jobs, which are allowed to decide about routes through the entire production system. It has already been shown, that autonomous control may improve the handling of dynamic complexity and increase the logistic target achievement of production systems (Scholz-Reiter et al. 2009a). Moreover comparative studies remarked that the dynamic performance of different autonomous control methods depends mainly on the scenarios parameters, e.g. arrival rate (Hülsmann et al. 2008).

As far as production networks are concerned several autonomous control approaches have already been formulated. Similar to production systems autonomous control leads to promising results (e.g., Scholz-Reiter et al 2009a, Scholz-Reiter et al. 2010). Nevertheless, these studies confirmed the interdependences between dynamical systems behavior and internal and external parameters. Due to the high degree of structural complexity, these networks may be even more sensitive to these effects: compared to single plants, production networks include many more factors (e.g., transport mode, distances or transport schedules), which affect the dynamical behavior of the system. As a consequence, the analysis of the dynamics of autonomously controlled production networks has to cover a broad range of possible impact parameters.

Moreover, autonomous control methods often offer some degrees of freedom, in terms of facultative parameters, which also have a major impact on the systems behavior.

3. ECHO STAE NETWORKS

Echo state networks (ESN) as proposed by Jaeger (2001) denote a special kind of architecture and supervised learning principle for recurrent neural networks. In contrast to feed forward networks ESNs are able to naturally cope with continuous inputs and exhibit a fading memory of the input history in the network's reservoir. In the recent past ESNs have *inter alia* been successfully employed for speech recognition (Skowronski et al. 2007), robot control (Hertzberg et al. 2002), and time series prediction (Webb 2008).

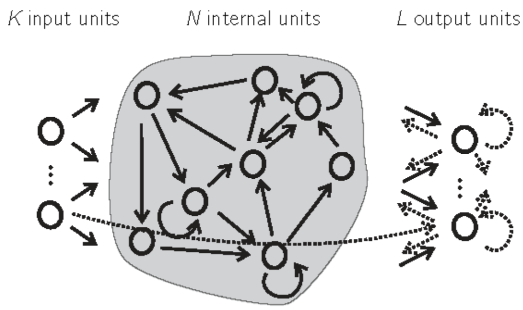


Figure 1: ESN architecture: an ESN has three layers. Solid arrows indicate mandatory types of connections, dashed arrows are optional connections. Figure reproduced and adapted from Jaeger (2001)

The typical architecture of a standard ESN is illustrated in figure 1. In an ESN the hidden layer containing the internal units is seen as a dynamical reservoir (DR), which is "tapped" using an attached linear readout module. In contrast to conventional RNN approaches only the linear readout is trained. This boils down to a linear regression task from a mathematical point of view, which is computationally cheap and guaranteed to converge. The hidden layer can be seen as a possibly high dimensional non-linear combination of the input signal, hence the name "dynamical reservoir" (Jaeger 2001). In order for the approach to work the DR must exhibit a special type of "damped" dynamics. A DR that exhibits this dynamics is attributed to have the echo state property, thus enabling a network based on that DR to be an echo state network. The echo state property basically ensures that for every internal signal there exists an echo function which maps input and output histories to the current state. That is, the network's DR does not exhibit chaotic dynamics that are independent of input or output. Echo state networks exhibit a form of short term memory, which can be seen as a consequence of the fact that echoes (possibly variations) of the input signal determine the current network state due to the echo state property. A thorough explanation of the architecture, setup and mathematical properties of ESNs is beyond the scope of this paper. However, for a very well-written introduction to the overall topic we would like to refer the reader to Jaeger's ESN tutorial (Jaeger 2005).

Echo state networks can be set up with different neuron types, different connectivity patterns and with or without output feedback connections. Generally, ESN setup parameters have to be chosen task-specifically. A common methodology here is to estimate network setup parameters manually following the intuition and experience of the experimenter as described by Jaeger (2005). What is more, numerous extensions and modifications for improving different aspects of the standard ESN architecture have been proposed in the literature. In the scope of this work, we utilize one of these, namely the delay&sum readout as proposed by Holzmann (2008). This extension to the ESN readout module allows for improved handling of long-term dependencies, which appears crucial in our experiments.

4. SCENARIO

A scenario with six different plants on four network stages is considered, similar to Scholz-Reiter (2009b). On stage one and on stage four there is only one plant. On stage two and three there are two collocated plants, respectively. Additionally, each plant consists of a shop floor with three parallel production lines and three machines per production line. Figure 2 shows the general structure of the scenario:

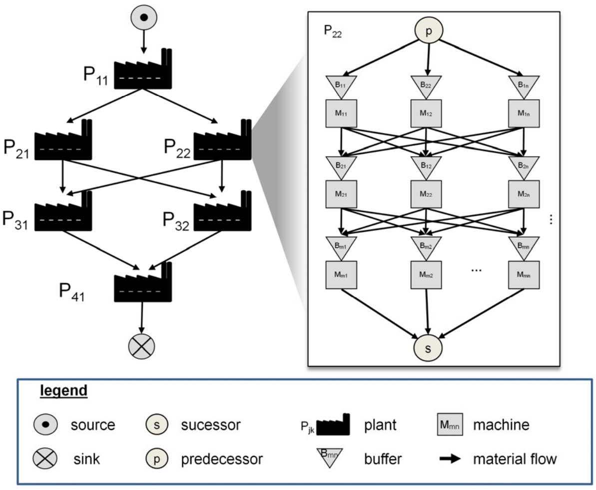


Figure 2: Production network scenario

There are three different job types. These job types differ in their processing times on the shop floor level. The processing times are set as summarized in Table 1:

Table 1: processing times

Plant	P_{11}; P_{41}			P_{21}; P_{22}; P_{31}; P_{32}		
Type / Line	1	2	3	1	2	3
Type A	2:00	3:00	2:30	4:00	5:00	4:30
Type B	2:30	2:00	3:00	4:30	4:00	5:00
Type C	3:00	2:30	2:00	5:00	4:30	4:00

The jobs arrive in plant 1 at stage 1 with a certain arrival rate. In order to model a dynamic seasonal demand, this arrival rate $\lambda(t)$ is set to a sine function:

$$\lambda(t) = \lambda_m + \beta \cdot \sin(t + \varphi) \tag{1}$$

This function has a phase shift φ of a period for each job type, so that the maximal arrival rates of all job types do no cumulate. The variable λ_m defines the mean of the arrival rate and is set to 0.4 1/h in all simulation runs. The second variable β determines the intensity of the arrival rate fluctuation.

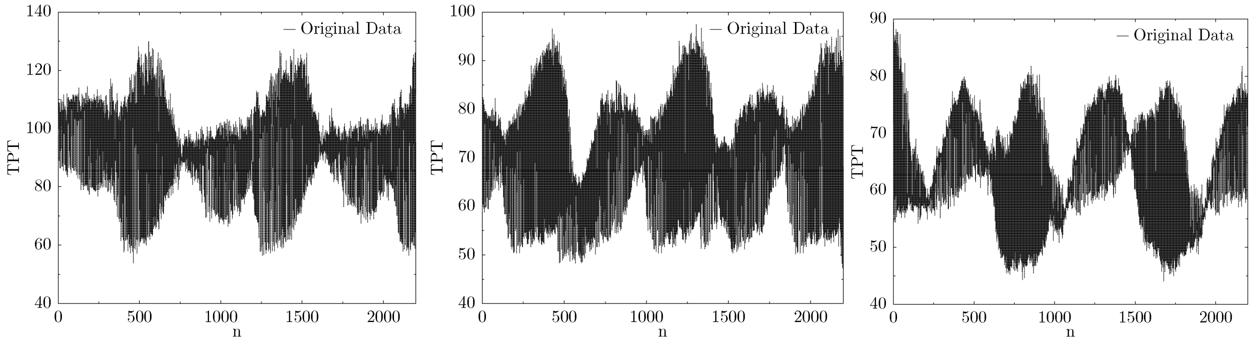
Transports between plants are triggered by a “go when full policy” (as described by Crainic 2002). This means, that a transport from one plant to the next plant starts as soon as a predefined amount of jobs has been finished in one plant. This transportation policy is commonly used in door to door transports (Crainic 2000). For this scenario the truck capacity q is set to 5 jobs. The distances between the plants D are 140 km each. The velocity of all trucks is set to 70 km/h. Hence, a transport from one plant to the next takes 2 hours.

Jobs in this scenario are able to decide about routes on the network and on the shop floor level, autonomously.

5. PHEROMONE-BASED METHODS

In order to route themselves through the scenario all jobs have two major decision problems: on the one hand the allocation to a plant has to be conducted. On the other hand they have to decide about routes within the respective plant. For both decision types a pheromone based approach is modeled, in order to reduce the throughput time of jobs through the entire network.

The pheromone approach is based on the idea to imitate the process in which ants mark possible routes to food sources in nature. Ants leave pheromone marks between their formicary and potential food sources. Other ants can detect those pheromones and will follow the tail with the highest concentration of pheromone (Parunak 1997). On the shop-floor-level this is done similarly (Armbruster et al. 2006): During the production process the jobs leave information about their processing and waiting times at a corresponding machine.



(a) (b) (c)

Figure 3: Excerpt plots (throughput time (TPT) against simulation steps n for characteristic patterns of original data retrieved from the simulation for different values of L (evaporation parameter). (a) pattern A with $L = 50$, (b) pattern B with $L = 150$, and (c) pattern C with $L = 250$. Note: the plots reflect the first 2200 samples of data points from each dataset after stripping data from initial attack times

Following jobs entering a stage on the shop floor compare this artificial pheromone concentration by computing an average value of the waiting time data of the last five jobs and then choose a production line. Thus, the pheromone concentration depends on waiting and processing times of previous jobs. To model the evaporation process of natural pheromones, a moving average of waiting time data is used.

What is more, on the network level the time spent to pass the transport system has to be considered. This time span is denoted as $T_{o,i}^{s,k}$, where i is the index of i th job which passed the plant k on stage s. Each plant has a vector $\overline{T}_o^{s,k}$ where o represents the product type. Whenever a job i leaves a plant, the waiting time for transports and processing times are appended to the corresponding $T_{o,i}^{s,k}$, see equation 2. The value $T_{o,i}^{s,k}$ of the last job i is appended at the first position.

$$\overline{T}_o^{s,k} = \begin{pmatrix} T_{o,i+1}^{s,k} \\ T_{o,i}^{s,k} \\ \vdots \\ T_{o,1}^{s,k} \end{pmatrix} \tag{2}$$

After processing in one plant the job has to choose a succeeding plant. Therefore a moving average over the last L jobs as an artificial pheromone (3) is calculated. Similar to the shop-floor-level the moving average is used to emulate the evaporation of natural pheromones.

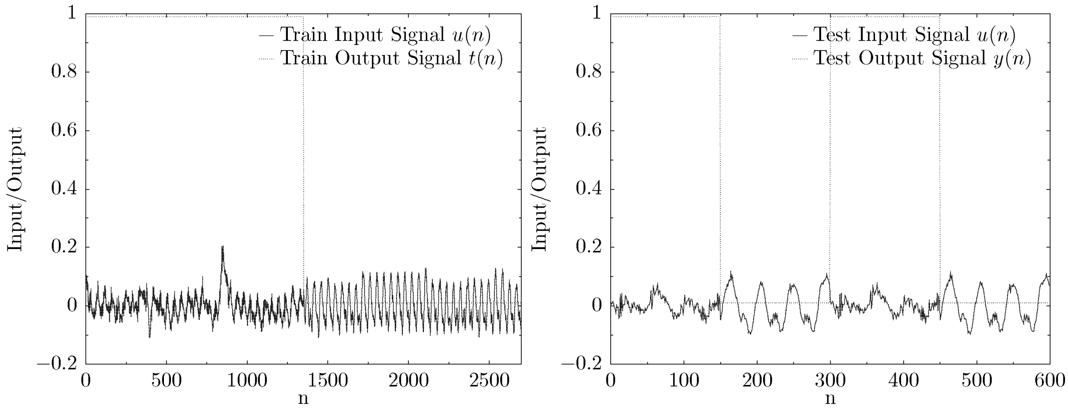
$$PHE_{s,k} = \frac{\sum_{i=1}^{L} T_{o,i}^{s,k}}{L} \tag{3}$$

After calculating the concentration of artificial pheromones, the job chooses the succeeding plant with the lowest average transportation and processing times. The scenario presented above is modeled by a discrete event simulation model. The pheromone method is implemented to this model for both: allocation decisions of jobs to plants on the network level and to production lines on the shop-floor-level. It is assumed that the evaporation parameter L has a major impact on the logistic performance and on the general dynamic behavior of the production network. In order to analyze the impact of parameter L, different simulation runs with varying values of L were conducted. Thereby, the logistic performance of the production network is recorded in terms of throughput times (TPT). This means the time span of each job to pass through the entire system from the source to the sink. The throughput times of every job are recorded as a time series.

Figure 3 shows three characteristic plots of TPT against simulation steps n for different values of L ($L = 50$, $L = 150$, $L = 250$). A visual comparison of these results shows that the curves follow different dynamic patterns. This indicates that the dynamic behavior differs in each case. With regard to the mean logistic performance these three time series cannot be distinguished that clearly. Only the simulation run with $L = 50$ differs distinctly from the others. The mean value of TPT for the simulation run with $L = 50$ is $TPT_{mean} = 92.87h$. While the mean TPT values of the other simulation runs are: $TPT_{mean} = 70.31h$ for $L = 150$ and $TPT_{mean} = 64.32h$ for $L = 250$. In order to investigate and classify the differences in the dynamic patterns of the time series an ESN is used.

6. SETUP AND RESULTS

In what follows, we describe our experimental setup as to classify two different characteristical patterns from the simulation runs. That is, in the scope of this work, we simplify the given problem to a binary classification task. Afterwards, we outline implications for possible further experimental setups, which aim at classifying more than two pattern classes in a multi-class network.



(a) (b)

Figure 4: ESN input and output signal plots for classification experiment with patterns A and C. n denotes ESN-based signal time steps (a) training signals (solid line: input, dotted line: output for teacher forcing), (b) testing signals (solid line: input, dotted line: output)

6.1. Data processing

To allow for utilization with the ESN approach, output datasets from the simulation runs must be normalized and further preprocessed.

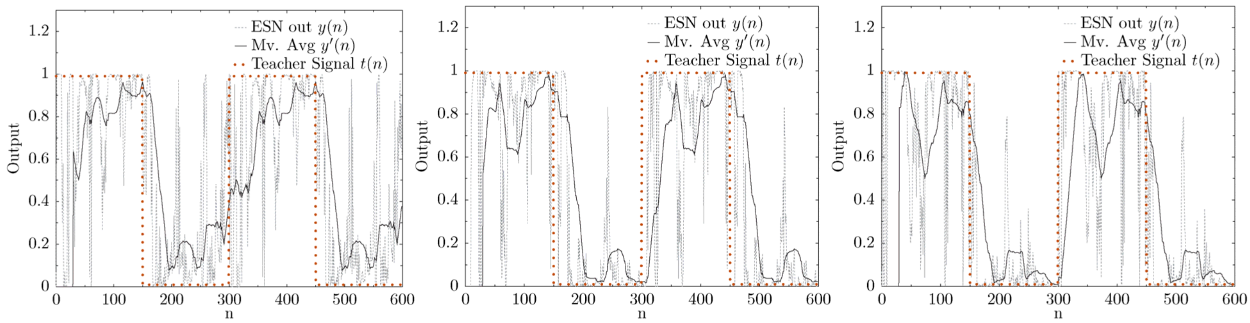
We outline our approach as to perform adequate data preprocessing in the case at hand as follows:

1. As a first step, data from initial attack times resulting from the simulation runs are stripped from each dataset. This is done to ensure that the ESN is not trained with chaotic patterns occurring only at the beginning of data collected from each simulation run.
2. Afterwards, we determine the global maximum ($g_{\max}$) and minimum ($g_{\min}$) values of all data that are to be utilized as an input signal to the ESN. This subsumes both training and testing data from all datasets used for the experiment.
3. We then normalize each dataset to range $[0,1]$ by the calculation of the term $d_{norm}(n) = (d(n) - g_{\min}) / (g_{\max} - g_{\min})$, where $d_{norm}(n)$ denotes the normalized data value and $d(n)$ represents the original data value at index n.
4. Subsequently, we shift the normalized data down by its mean, so that the represented signal oscillates around zero. This is done by calculating $d_{ms}(n) = d(n) - d_{mean}$, where d_{mean} is the arithmetic mean of the normalized dataset and $d_{ms}(n)$ denotes the final mean-shifted signal. Note that we chose to mean-shift each dataset on its own, i.e. each dataset is first shifted and later datasets are concatenated for training and testing using an ESN. This has been done to allow for commensurability with respect to the way ESN experiments are carried out later on.
5. In order for the ESN approach to work, the data must exhibit useful patterns at appropriate time-scales, which can be captured with limited short term memory inherently available in the network. Since we deal with very fine-grained original output data, each dataset is thus super-sampled in order to reduce the number of data points. This is done by segmenting the dataset into segments of $s = 20$ data values each, then calculating the average of each segment and finally storing the concatenation of all consecutive segment averages as a result in a newly created dataset. The value of s was determined empirically and may vary with different simulation scenarios.

6.2. Signal Composition

Finally, datasets are partitioned for training and testing as illustrated in figure 4. As is common practice in the machine learning domain, we use 90 percent of the data for training and the remaining 10 percent for testing. In the case at hand, we partition datasets so that the first 90 percent of the signal are devoted for training and the last 10 percent are utilized for testing. The datasets are then used for composing input signals $u(n)$ and output signals $y(n)$ for training and testing the ESN as explained in the following.

For training the ESN, two training datasets D_A and D_B with lengths l_A and l_B are chosen that exhibit a characteristical pattern, which is supposed to be recognized by the ESN. The training input signal $u(n)$ is then composed by concatenating training set D_A and D_B. The training output signal $t(n)$ is created of a block of ones (1.0) with length l_A following a block of zeros (0.0) with length l_B.



(a) (b) (c)

Figure 5: ESN output and test signal (dotted line: ESN output signal, solid line: moving average over ESN output signal, reddish thick dotted line: test signal for comparison of ESN output with ground truth

The training output signal can be considered a step signal, which is designed to train the ESN to vote for class C_A (1.0) or class C_B (0.0) depending on the underlying input signal at time step n. For testing the ESN the same scheme is applied, but this time the testing partitions of the respective datasets are employed and output $y(n)$ is used only for evaluation purposes. Figure 4 shows plots (a) of the resulting overall training signal and (b) of the testing signal.
Note that the testing signal was repeated once to improve illustration of the ESN output signals (see section 6.4).

6.3. Network setup

Table 1 gives an overview of the ESN configuration employed in the scope of this work. It contains all relevant parameters and values for the ESN used in the case at hand.

Table 2: Overview of the ESN setup for the classification experiments

Parameter	Value
Number of DR neurons N	400
DR connectivity	10%
Spectral radius α	0.4
Input connectivity	10%
Input scaling	6.0
Internal activation function	sigmoid: tanh
Output activation function	sigmoid: $\frac{1.0}{(1.0+e^{t})}$

An echo state network is set up with one input signal $u(n)$ and one output signal $y(n)$. Internal units use the tanh activation function while output units use the logistic sigmoid activation function. The number of internal units is set to $N = 400$. The network size was determined empirically, i.e. increasing the network size had no significant effect on performance while decreasing it led to considerably worse performance. The DR's spectral radius is set to $\alpha = 0.4$ in order to account for a required tradeoff between memory capacity and other factors influencing classification performance. The DR's connectivity parameter is set to 10 percent. Inputs are re-scaled by factor 6.0, thereby yielding input weights sampled from a uniform distribution in range $[-6.0, 6.0]$. The input unit is connected to 10 percent of the DR's internal units. Output feedback is completely omitted as to the nature of our classification experiments. We utilize a modified readout, namely the delay&sum readout as proposed by Holzmann (2008), which is not part of the ESN standard as proposed by Jaeger.

The delay&sum readout allows for taking into account long-term dependencies in the ESN's input signal without increasing network size beyond the limits of current computer hard- and software. We use the delay&sum readout with generalized cross correlation (GCC) and a maximum delay of 10 simulation steps. Training is done with a special pseudo-inverse algorithm with respect to estimating delays in the readout module as described by Holzmann (2008). ESN-based simulation (testing) is done with additional squared states as described by Jaeger (2003).

6.4. Results

Figure 5 shows ESN outputs for simulation runs on (a) pattern A and B, (b) pattern A and C, and (c) pattern B and C. The dotted graphs for each run represent the output signals generated by the ESN (trained with 2700 training samples) and tested on 600 test samples (300 samples repeated twice). During training the first 100 samples were omitted to washout chaotic dynamics induced by the random initialization of the ESN. Table 3 lists the results of the simulation runs performed with the ESN on the different training and testing sets in terms of the normalized root mean square error (NRMSE), which is commonly employed for ESN performance evaluation.

Table 3: ESN normalized root mean square errors for training $NRMSE_{Train}$ and testing $NRMSE_{Test}$ on each simulation run

Run	$NRMSE_{Train}$	$NRMSE_{Test}$
(a) A vs. B	0.325	0.804
(b) A vs. C	0.224	0.768
(c) B vs. C	0.214	0.596

6.5. Discussion

The NRMSE results suggest that both fitting on training data and correctly predicting testing data was not achieved with satisfactory performance. The best case can be seen in (c) where patterns B and C are classified with test performance $NRMSE_{Test} = 0.596$ while the worst result is achieved for (a) pattern A vs. B with $NRMSE_{Test} = 0.804$.

However, even though NRMSE result values indicate fairly bad performance of the network in this particular case the ESN output signals plotted in figure 5 indicate that it appears possible to perform a successful classification when we apply some post-processing. A fairly simple approach that turned out to be quite effective is applying a moving average with window size $w = 30$ (determined empirically) on the ESN output signal. Figure 5 illustrates this moving average with a solid black graph plotted above the original ESN output during testing. It can be seen that this adequately compensates for spikes to the wrong output value and enables partitioning of the output range to sub-ranges $(0.5, 1.0]$ for the first class and $[0.0, 0.5]$ for the second class respectively. A binary readout attached to the moving average output $y^{'}(n)$ of the ESN could thus determine crisp output class predictions $C(n)$ as indicated in equation 4:

$$C(n) = \begin{cases} true \ if \ y^{'}(n) > 0.5 \\ false \ if \ y^{'}(n) \leq 0.5 \end{cases} \tag{4}$$

Looking at figure 5 we can state that the described approach of post-processing the ESN output signal does at least hold for cases (b) and (c). Case (a) appears to be harder to handle for the ESN, which might be caused by the similarity of patterns A and B.

Concluding, it can be stated that we were able to successfully classify dynamical patterns retrieved from the logistics simulation using an ESN. An advantage of this method over e.g. the Fourier transform applied in Scholz-Reiter (2007) is that, as soon as a suitable ESN configuration is found, it can be applied very easily and without much manual effort.

7. CONCLUSION

In order to improve the robustness and performance of logistic systems in a dynamical environment we applied the concept of autonomous control in an exemplary simulation model scenario. In this context we introduced a pheromone-based approach for routing within such scenario. We identified the need for a more sophisticated analysis of the interplay between internal dynamics and the overall system's performance in such autonomously controlled environments. In order to enable an analysis going beyond a sole comparison of aggregated data like mean values and standard deviations, we presented an approach, based on Echo State Networks, for identifying and comparing different dynamics, gained by a simulation model of an autonomous controlled production network. We have shown that echo-state networks are able to distinguish different patterns of an exemplary autonomous controlled production network with varying initial parameters.

8. FUTURE RESEARCH

In the scope of this work, we have demonstrated that classifying different time series patterns retrieved from an exemplary autonomously controlled logistics simulation model with varying parameters is possible using an ESN. In the context of logistics research it is of interest to utilize and further develop methods like the one presented in this paper in order to gain a deeper understanding of algorithms and methods used in simulation models. For instance, our approach could be extended to identifying aspects of autonomous control methods that manifest in output patterns and may have influence on the logistics system's performance.

ACKNOWLEDGMENTS

The research presented in this paper was supported by the German Research Foundation (DFG) as part of the Collaborative Research Center 637 “Autonomous Cooperating Logistic Processes.”

REFERENCES

Armbruster, D., de Beer, C., Freitag, M. Jagalski, T. and Ringhofer, C., 2006. Autonomous control of production networks using a pheromone approach. In: *Physica A*: Statistical Mechanics and its Applications, 363(1), pp. 104–114

Camarinha-Matos, L.; Afsarmanesh, H. (2003): Elements of a base VE infrastructure. In: Computers in Industry 51, 139–163

Crainic, T.G., 2000. Service network design in freight transportation. In: *European Journal of Operational Research* 122(2),pp. 272-288

Hertzberg, J., Jaeger, H. and Schönherr, F., 2002. Learning to Ground Fact Symbols in Behavior-Based Robots. In *ECAI*, pp. 708– 712

Holzmann, G., 2008. Echo State Networks with Filter Neurons and a Delay and Sum Readout with Applications in Audio Signal Processing, Master's thesis, 2008.

Hülsmann, M., Scholz-Reiter, B., de Beer, C. and Austerschulte, L., 2008. Effects of Autonomous Cooperation on the Robustness of International Supply Networks - Contributions and Limitations for the Management of External Dynamics in Complex Systems,In: *LogDynamics International Conference*, pp. 241–250, Berlin, Springer.

Ivanov, D., 2009. An adaptive framework for aligning (re)planning decisions on supply chain strategy, design, tactics, and operations, In: *International Journal of Production Research*, (907681677), pp. 1–19

Jaeger, H., 2001. The "echo state" approach to analysing and training recurrent neural networks, Technical report, GMD - German National Research Institute for Computer Science

Jaeger, H., 2003. Adaptive nonlinear system identification with echo state networks', in Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, pp. 593–600

Jaeger, H., 2005. 'A tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach, Technical report, Fraunhofer Institute for Autonomous Intelligent Systems (AIS)

Parunak, H.V., 1997.Go the ant: engineering principles from natural multi agent systems. *Annals of Operations Research* 75, pp. 69–101

Sauer, J., 2006. Modeling and Solving Multi-Site Scheduling Problems. In: Meystel (ed.):*Planning in Intelligent Systems: Aspects, Motivations and Methods*, Wiley, pp. 281–299

Scholz-Reiter, B., de Beer, C., Freitag, M. and Jagalski, T., 2007. Analysing the Dynamics caused by Autonomously Controlled Logistic Objects. In: *Proceedings of 2nd International Conference on Changeable, Agile, Reconfigurable and Virtual Production*, El- Maraghy, H.A. and M. F. Zaeh (eds) , pp. 273–280,

Scholz-Reiter, B., Görges, M. and Philipp, T., 2009a. Autonomously controlled production systemsInfluence of autonomous control level on logistic performance. In: *CIRP Annals - Manufacturing Technology,* 58(1), pp. 395–398

Scholz-Reiter, B., Görges, M., Jagalski, T. and Mehrsai, A., 2009b. Modelling and Analysis of Autonomously Controlled Production Networks. In: *Proceedings of the 13th IFAC Symposium on Information Control Problems in Manufacturing*, pp. 850–855

Scholz-Reiter, B., Mehrsai, A., Görges, M., 2010. Handling the Dynamics in Logistics - Adoption of Dynamic Behavior and Reduction of Dynamic Effects. In: *Asian International Journal of Science and Technology in Production and Manufacturing Engineering (AIJSTPME)*, 2(2009)3, pp. 99-110

Skowronski, M.D. and Harris, J.G., 2007. Automatic speech recognition using a predictive echo state network classifier, In: *Neural Networks* 20, pp. 414–423

Webb, R., 2008. Multi-Layer Corrective Cascade Architecture for on-Line Predictive Echo State Networks, *Applied Artificial Intelligence*, 22(7), pp. 811–823

Wiendahl, H.P. and Lutz, S., 2002. Production in Networks, *CIRP Annals – Manufacturing Technology*, 51(2), pp. 573–586,

Windt, K. and Hülsmann, M. 2007. Changing Paradigms in Logistics – Understanding the Shift from Conventional Control to Autonomous Cooperation and Control. In: Understanding Autonomous Cooperation & Control - The Impact of Autonomy on Management, Information, Communication, and Material Flow. Hülsmann, M. and Windt, K. (eds), pp. 4–16

Windt, K., 2006. Selbststeuerung intelligenter Objekte in der Logistik. In: Selbstorganisation - Ein Denksystem für Natur und Gesellschaft. Vec, M., Hütt, M. and Freund, A. (eds), pp. 271–314

Windt, K.; Böse, F.; Philipp, T., 2008. Autonomy in Production Logistics - Identification, Characterisation and Application. In: International Journal of Robotics and CIM, 24(4), pp. 572-578

AUTHORS BIOGRAPHY

Prof. Dr.-Ing. Bernd Scholz-Reiter is head of the department Planning and Control of Production Systems at the University of Bremen and managing director of the BIBA - Bremer Institut für Produktion und Logistik GmbH at the University of Bremen. He is Editor of the professional journals Industrie Management and PPS Management. Prof. Scholz-Reiter is a corresponding member of C.I.R.P., the International Institution for Production Engineering Research and also a member of the Scientific Advisory Board of Bundesvereinigung Logistik e.V.

M.Sc. Tobias Lensing works as a research scientist at the BIBA - Bremer Institut für Produktion und Logistik GmbH at the University of Bremen in the division „Intelligent Production and Logistic Systems.

Dipl.-Wi.-Ing. Michael Görges works as a research scientist at the BIBA - Bremer Institut für Produktion und Logistik GmbH at the University of Bremen in the division „Intelligent Production and Logistic Systems.

M.Sc. Lutz Dickmann works as a doctoral canidate at Research Group Digital Media at the University of Bremen.