

## ADAPTIVE PREDICTION OF STOCK EXCHANGE INDICES BY STATE SPACE WAVELET NETWORKS

MIETEK A. BRDYŚ<sup>\*,\*\*</sup>, ADAM BOROWA<sup>\*\*</sup>, PIOTR IDŹKOWIAK<sup>\*,\*\*\*</sup>, MARCIN T. BRDYŚ<sup>\*\*\*\*</sup>

<sup>\*</sup> School of Electronic, Electrical and Computer Engineering, College of Engineering and Physical Sciences  
University of Birmingham, Edgbaston, Birmingham B15 2TT, UK  
e-mail: mbrdys@bham.ac.uk

<sup>\*\*</sup> Department of Control Systems Engineering  
Gdańsk University of Technology, ul. G. Narutowicza 11/12, 80–233 Gdańsk, Poland  
e-mail: mbrdys@ely.pg.gda.pl

<sup>\*\*\*</sup> Philips Lightning Poland BU CLE / IPLC GLS 10NC & CFL-i Burners Assistant Industrial Engineer

<sup>\*\*\*\*</sup> Corporate Banking & MIB Division, Financial Markets Department, Planning, Controlling & Support  
Bank Pekao SA, ul. Grzybowska 53/57, 00–950 Warsaw, Poland  
e-mail: marcin.brdys@pekao.com.pl

The paper considers the forecasting of the Warsaw Stock Exchange price index WIG20 by applying a state space wavelet network model of the index price. The approach can be applied to the development of tools for predicting changes of other economic indicators, especially stock exchange indices. The paper presents a general state space wavelet network model and the underlying principles. The model is applied to produce one session ahead and five sessions ahead adaptive predictors of the WIG20 index prices. The predictors are validated based on real data records to produce promising results. The state space wavelet network model may also be used as a forecasting tool for a wide range of economic and non-economic indicators, such as goods and raw materials prices, electricity/fuel consumption or currency exchange rates.

**Keywords:** forecasting, stock exchange, artificial intelligence, state space wavelet network, simulated annealing.

### 1. Introduction

Nowadays the world is global market and one may say that all worldwide markets become parts of the general financial system. Moreover, the perpetual pursuit of money is an unquestionable fact. Stock markets offer quick profit opportunities. Therefore, they are key parts of financial systems. On the other hand, the stock market is the biggest gambling area where all investors speculate about share prices. The Polish stock market in Warsaw is not an exception. The key Warsaw Stock Exchange (WSE) index is WIG20. It is the index of twenty leading companies on the WSE. Contracts between investors regarding the future price of the WIG20 index constitute a very popular financial instrument on the WSE. Therefore, investors are greatly interested in having a reliable tool for WIG20 forecasting. Forecasting stock prices has been widely investigated nearly since the beginning of stock market his-

tory. There is a number of 'classical' forecasting methods such as technical analysis, fundamental analysis, Efficient Markets Hypothesis (EMH), or various time series methods like ARMA based methods (Mun, 2006; Tsang *et al.*, 2007). Also, artificial intelligence models based on neural networks and/or fuzzy systems are of wide interest (Zhang *et al.*, 1998; Kuo *et al.*, 2001; Tsang *et al.*, 2007).

In this paper, a recently developed artificial dynamic neural network with wavelet processing nodes and internal states called the State Space Wavelet Network (SSWN) is proposed for forecasting the stock market index value. The SSWN was initially proposed for modelling nonlinear and nonstationary processes with multiple time scales in internal dynamic and hardly measured states under uncertainty in the inputs and dynamic models. It was successfully applied to the input–output modelling in a state-space form of a wastewater treatment plant (Borowa *et al.*, 2007). The wastewater treatment

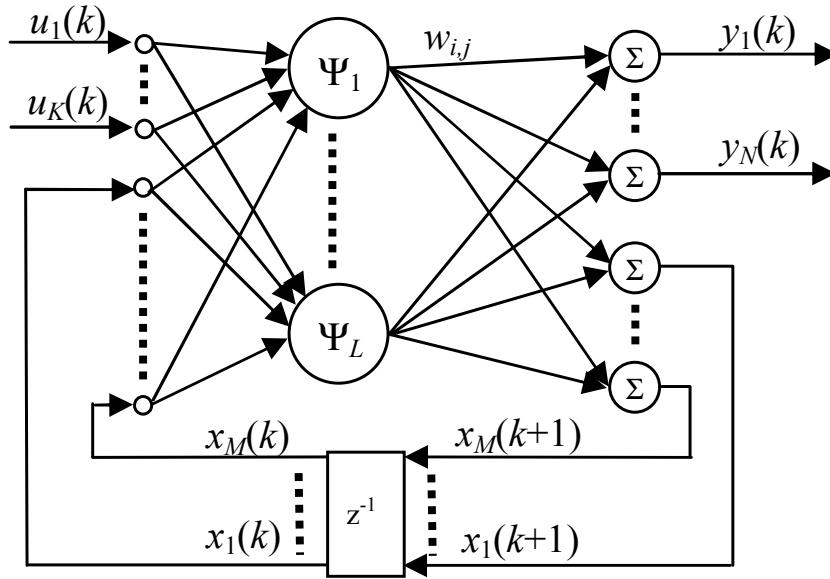


Fig. 1. General structure of the SSWN.

process is commonly recognised as complex (Brdyś *et al.*, 2008). In addition, it is also a large scale system due to its geographical distribution. The stock market has already been recognised as a large scale and complex system. Therefore, the SSWN considered is seen as a suitable candidate for a modelling base for developing a tool for reliable and accurate stock market forecasting. The structure of the SSWN is illustrated in Fig 1, where  $y_i, i = 1, \dots, N$ ;  $x_i, i = 1, \dots, M$  and  $u_i, i = 1, \dots, K$  denote network outputs, internal states and inputs, respectively. All the variables are discrete time, and the time variable is denoted by  $k$ .

Network internal states do not have to be related to states of the modelled system. In the case of unknown or unmeasurable system states, this is a great advantage of the network model. Identifying state variables of a complex system is in most cases impossible. However, although artificial neural model states can still correctly describe the impact of system state variable dynamics on the system output (Zamarreno and Pastora, 1998; Kulawski and Brdyś, 2000). This vastly improves the ability of the model to approximate unknown system input-output dynamics. The SSWN input-output relationship can be written as

$$x_i(k+1) = \sum_{j=1}^L w_{N+i,j} \Psi_j(x(k), u(k)), \quad i = 1, \dots, M, \quad (1)$$

$$y_i(k) = \sum_{j=1}^L w_{i,j} \Psi_j(x(k), u(k)), \quad i = 1, \dots, N, \quad (2)$$

where  $w_{i,j}, i = 1, \dots, N + M, j = 1, \dots, L$  are network weights to be determined and  $u(k), x(k)$  are network input and state vectors at the time instant  $k$ , respectively.

Network nodes that process the input information at time instants  $k$  are Multidimensional Radial Wavelons (MRWs) (Zhang, 1992). MRW processing mappings are denoted by  $\Psi_j, j = 1, \dots, L$ . Feedforward networks with wavelet based processing nodes were introduced in (Zhang and Beneveniste, 1992). The MRW input information processing structure is illustrated in Fig. 2. The mapping  $R$  in the Fig. 2 is defined as

$$z(k) = [x(k), u(k)], \quad (3)$$

$$d_j = [d_{1,j}, \dots, d_{K+M,j}], \quad (4)$$

$$t_j = [t_{1,j}, \dots, t_{K+M,j}], \quad (5)$$

$$A(z(k), d_j, t_j) = \text{diag}(d_j)(z(k) - t_j)^T, \quad (6)$$

$$R(z(k), d_j, t_j) = a_j(k) = [A^T(z(k), d_j, t_j)A(z(k), d_j, t_j)]^{1/2}, \quad (7)$$

where the vectors  $d_j, t_j, j = 1, \dots, L$  are composed of the  $j$ -th MRW parameters. Solving Eqns. (3)–(7) yields

$$a_j(k) = [(d_{1,j}(x_1 - t_{1,j}))^2 + \dots + (d_{M,j}(x_M - t_{M,j}))^2 + (d_{M+1,j}(u_1 - t_{M+1,j}))^2 + \dots + (d_{M+K,j}(u_K - t_{M+K,j}))^2]^{1/2}. \quad (8)$$

From (8) it can now be seen that the components of the parameter vector  $d_j$  are scaling coefficients for the network inputs, while the components of the parameter vector  $t_j$  perform translations of the inputs. Finally, the

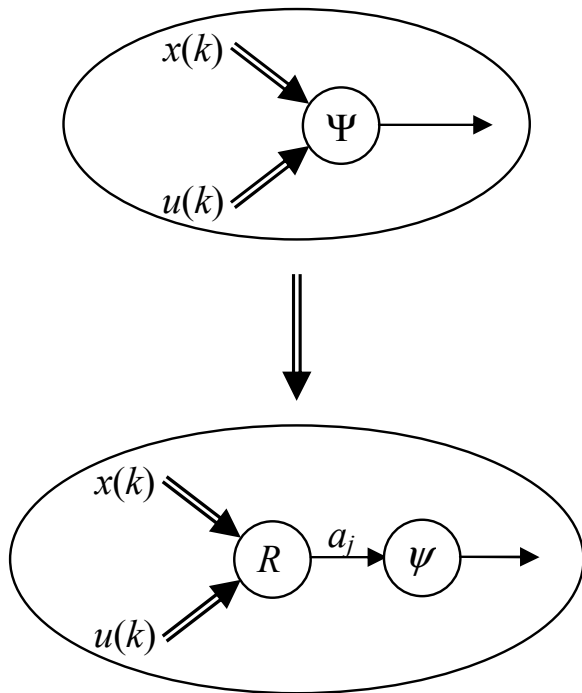


Fig. 2. Multidimensional radial wavelon.

mapping  $\psi$  in Fig. 2 is a one-dimensional Morlet wavelet function (Grossmann and Morlet, 1984):

$$\psi(a_j) = \exp(-a_j^2/2) \cos(5a_j). \quad (9)$$

Equations (1)–(9) define an input-output model in a state space form with the weights  $w_{i,j}$ , scaling and translation factors  $d_{i,j}$ ,  $t_{i,j}$ , respectively, that are continuously varying model parameters to be determined.

In order to design the Stock Exchange Index predictor based on the described SSWN, not only suitable parameter values must be determined. The SSWN structure depends on the number of wavelons and states  $L$  and  $M$ , respectively. These are structure parameters to be determined. Finally, it is very important to identify a vector of all essential input variables  $u$ . In this application, the latter is difficult and requires at least qualitative knowledge regarding functioning mechanisms of the modelled Stock Exchange. Unfortunately, such knowledge is still limited. However, the SSWN applied is supposed to be built based on principles of artificial intelligence. Therefore, it is expected that with a properly chosen structure, which implies the SSWN sufficient approximation capability to get it accurately trained based on available knowledge, the trained SSWN will manage to properly respond also under different data than used during its training.

The paper is organised as follows: The Warsaw Stock Exchange and the key WIG20 index are presented in Section 2. The input and output structures of SSWN based models of the index for the short term and longer term index prediction are developed in Section 3. Section 4

presents principles of the SSWN off-line training and formulates the training data structure and performance function. An optimisation solver is described in Section 5, and it is applied to determine off-line predictor parameters based on historical data records. A mechanism for on-line network parameter update is introduced in Section 6 to produce adaptive index predictors. The adaptive predictors are validated in Section 7. Section 8 concludes the paper.

## 2. Presentation of the Warsaw Stock Exchange and the WIG20 index

The tradition of capital market in Warsaw goes back to the 19-th century, when the first mercantile exchange was founded in 1817. The Warsaw Stock Exchange (WSE) in the current form was established in 1991 after the fall of the communism in Poland. At the beginning, there were only five companies on the market. Comparing with today's three hundred companies, the WSE has rapidly grown over last 16 years and is at present the most important stock market in Central and Eastern Europe, and it is still growing. The capitalisation at the end of August 2007 was more than 300 billion US dollars (<http://www.gpw.pl>, 20-th October 2007).

At present, the following instruments are traded: shares, bonds, subscription rights, allotment certificates, investment certificates, and derivative instruments such as futures, options and index participation units.

There are two main markets within the WSE: main and parallel. The capitalisation and stability of companies operating on the main market is rather high. An average capitalisation and stability of companies operating on the parallel market is smaller, but an average investment rate of shares of small companies is rather high. WSE development in time is illustrated in Figs. 3 and 4.

There are more than ten indices on the WSE, but the major indices on the market are WIG, WIG20, mWIG40, sWIG80. The WIG index includes prices of all shares on the market, while the WIG20 index includes shares of twenty leading companies. The mWIG40 index includes shares of forty medium-size companies, and the sWIG80 index includes shares of the smallest companies. There is also the New Connect index, established in August 2007, which includes shares of small and innovative businesses but comparing with other indices is not regulated.

The Warsaw Stock Exchange Price Index (WIG20) numerical value is an aggregated price change of twenty, mostly domestic, stocks, which are best ranked. The formula for determining the WIG20 index score draws upon both the turnover and market capitalisation during the month preceding the quarterly revision of the index sample in a proportion of 60:40. In addition, in order to be qualified as a new stock in the index sample, a stock must have ranged among the twenty leaders in at least two of

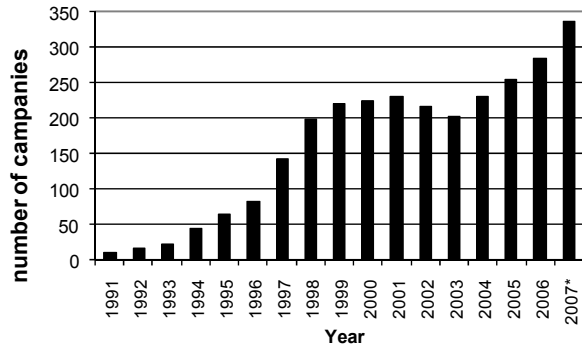


Fig. 3. Number of companies on the WSE (\* forecast made after nine months).

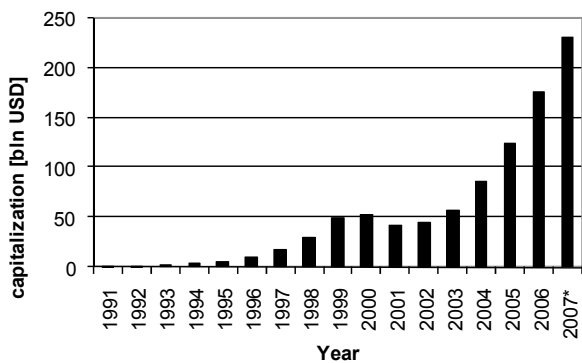


Fig. 4. Market value (\* forecast made after nine months).

the last three months in terms of the turnover. The WIG20 index is also used as the base instrument in calculating and valuating terminal contracts. Its value is announced at the beginning and the end of the trading session to produce the opening and the closing prices, respectively. In addition, the index value is calculated on-line during the session every thirty seconds. The index has been calculated since the 16-th of April 1994 and all records ever are available. The value of the WIG20 index is strongly correlated with trading results of worldwide stocks and, in particular, the stocks in Europe and North America.

### 3. Model structure for WIG20 index forecasting

The SSWN model well captures such features of stock exchange trading dynamics as nonlinearities, uncertainty in inputs and structure, and different time scales. However, significant structure changes of the stock exchange due to bankruptcy and/or the emerging of new major investors, for example, requires on-line adaptation of the model structure (Qi and Brdyś, 2005; 2008).

Correct selection of essential inputs to the WIG20 index system is a mile stone in designing its SSWN model. As relations between economic indicators on stock exchange markets are extremely complex and almost impossible to be measured or estimated, it is impossible to

Table 1. Results of correlation analysis.

Indicator symbol	Correlation coefficient	Description of indicator
WIG20cl	1,00	Value of WIG20 index at the end of session
WIG20vol	0,60	Session volume of WIG20 index
WIGcl	1,00	Value of WIG index at the end of session
sWIG80cl	0,86	Value of sWIG80 index at the end of session
EUROcl	-0,78	Exchange rate EURO/PLN
USDcl	-0,74	Exchange rate USD/PLN
FTSE100cl	0,97	Value of FTSE100 index at the end of session
S&P500cl	0,92	Value of S&P500 index at the end of session
DAXcl	0,97	Value of DAX index at the end of session
DJIAcl	0,81	Value of DJIA index at the end of session
CAC40cl	0,97	Value of CAC40 index at the end of session

choose all the factors that influence the index considered. Therefore, it is attempted to choose only the most important factors that influence the predicted index values. Unknown, complex and nonlinear relations between inputs and outputs of the SSWN model are estimated during the process of training, which is discussed in the sequel.

A common approach to selecting input variables is to build, based on qualitative knowledge, a list of potential measurable inputs and to apply a standard data correlation analysis to calculate correlation coefficients between the input and the output considered. The final input selection is then based on correlation coefficient values. The larger the coefficient, the higher selection priority to the corresponding input is assigned.

The results of correlation analysis based on preselected input variables and future values of the WIG20 index are shown in Table 1. Correlation analysis is strictly valid only for linear relationships between the predicted index and economic indicators that are the inputs. In re-

ality, these relationships are typically heavily nonlinear. Hence, the analysis should be seen as qualitative. The final selection of inputs needs to be done within an iterative process where different inputs are tried, the predictor is validated and, based on the validation results, new inputs are produced. The process stops when the required prediction accuracy is reached.

About 20 input variables were preliminarily selected when designing the SSWN for one day prediction. Those included the Polish stock indices, the key American and European stock indices, terminal contracts, currency exchange rates, changes of WIG20 index over one session.

Apart from the above input-output correlation analysis, mutual input correlation analysis was performed in order to reduce the number of inputs. Namely, two inputs with a high mutual correlation coefficient such as currency exchange rates, for example, were replaced by one of them.

Let us consider trading sessions during the day  $k - 1$  and  $k$ . Let  $k - 1$  be a discrete time instant located between the session  $k - 1$  closing time and the session  $k$  opening time. For the  $k$ -th session, let  $WIG20_{cl}(k)$  and  $WIG20_{op}(k)$  denote the closing and opening index values, respectively. The following seven inputs are selected for the SSWN to produce the  $k$ -th session (daily) prediction  $WIG20_{cl}(k|k - 1)$  of  $WIG20_{cl}(k)$  performed at instant  $k - 1$ , that is, after closing the session  $k - 1$  and at the same time before opening the session  $k$ :

$u_1(k) \triangleq WIG20_{cl}(k - 1)$  is the index value at the end of the previous trading session;

$u_2(k) \triangleq \frac{WIG20_{cl}(k-1) - WIG20_{cl}(k-2)}{WIG20_{cl}(k-2)}$  is the daily relative change of the index WIG20 value over the previous day, that is, the day  $k - 1$ ;

$u_3 \triangleq WIG20_{cl}(k - 1) - WIG_{op}(k - 1)$  is the session index change over the previous trading session that is, the session  $k - 1$ ; let us notice that  $WIG20_{cl}(k - 2) \neq WIG20_{op}(k - 1)$ ;

$u_4$  is the difference between the overall gain and overall loss on the WSE;

$u_5(k) \triangleq \frac{FTSE100_{cl}(k-1) - FTSE100_{cl}(k-2)}{FTSE100_{cl}(k-2)}$  is the daily relative change of the British index FTSE100 value over the previous day, that is, the day  $k - 1$ ;

$u_6(k) \triangleq \frac{S\&P500_{cl}(k-1) - S\&P500_{cl}(k-2)}{S\&P500_{cl}(k-2)}$  is the daily relative change of the American index S&P500 value over the previous day, that is, the day  $k - 1$ ;

$u_7(k)$  is an integrated PLN exchange rate with regard to EUR and USD calculated as an unweighted average of the two based on their daily average exchange rates over the days  $k - 1$  and  $k - 2$ , measured in the GMT+1 time.

Due to SSWN training properties, the input variable values are scaled to the intervals  $[-1, 1]$  for the variables  $u_2, \dots, u_7$  that can take both positive and negative values and  $[0, 1]$  for  $u_1$ , which is positively valued. At the instant  $k - 1$ , the SSWN network operates as follows:

first the new state at  $k$  is calculated from (1) as  $x_i(k) = \sum_{j=1}^L w_{N+i,j} \Psi_j(x(k - 1), u(k - 1))$ ,  $i = 1, \dots, M$ , and then the network output is calculated according to (2) as  $y(k) \triangleq y_1(k) = \sum_{j=1}^L w_{1,j} \Psi_j(x(k), u(k))$ ,  $i = 1, \dots, N$ . Let us notice that both the inputs  $u(k - 1)$  and  $u(k)$  are well defined from the past measurement (observation) data.

Let us consider now the input structure for the five sessions (days) ahead prediction of the index WIG20 value, that is, the prediction  $WIG20_{cl}(k + 4|k - 1)$  of  $WIG20_{cl}(k + 4)$ , which is performed after closing the session  $k - 1$  and at the same time before opening the session  $k$ , that is, at the time instant  $k - 1$ . The one session ahead predictor would be taken as the basis for designing the five days ahead predictor. Indeed, a standard approach would be to consecutively apply, starting from  $k - 1$ , the one session ahead predictor. However, only the inputs at  $k - 1$  could be used exactly as required. For  $k, k + 1, \dots, k + 4$ , the required inputs would have to be replaced by their predictions available at  $k - 1$ . It has been verified that the accuracy of these predictions is not sufficient to obtain in this manner a desired prediction accuracy of the index value at  $k + 4$ . Hence, the one session prediction approach to produce the five session ahead prediction must be discarded and suitable inputs must be selected again. The following inputs are finally selected as the result of extensive analysis performed as described before:

$$u_1(k) \triangleq \frac{WIG20_{cl}(k - 1) - WIG20_{cl}(k - 6)}{WIG20_{cl}(k - 6)},$$

$$u_2(k) \triangleq \frac{\frac{1}{5} \sum_{i=1}^5 WIG20_{cl}(k - i) - \frac{1}{5} \sum_{i=6}^{10} WIG20_{cl}(k - i)}{\frac{1}{5} \sum_{i=6}^{10} WIG20_{cl}(k - i)}$$

$$= \frac{\sum_{i=1}^5 WIG20_{cl}(k - i) - \sum_{i=6}^{10} WIG20_{cl}(k - i)}{\sum_{i=6}^{10} WIG20_{cl}(k - i)},$$

relative average change of the index value over the five-day period, and

$$u_3(k) \triangleq \frac{WIG_{cl}(k - 1) - WIG_{cl}(k - 6)}{WIG_{cl}(k - 6)},$$

$$u_4(k) \triangleq \frac{FTSE100_{cl}(k - 1) - FTSE100_{cl}(k - 6)}{FTSE100_{cl}(k - 6)},$$

$$u_5(k) \triangleq \frac{S\&P500_{cl}(k - 1) - S\&P500_{cl}(k - 6)}{S\&P500_{cl}(k - 6)}.$$

As previously, the inputs are appropriately scaled so that their values belong to the intervals  $[-1, 1]$  and  $[0, 1]$ .

There are other qualitative factors having significant impact on the predicted index values such as political cli-

mate or investor moods, which are not on the input variable list as they are not numerically quantifiable. In order to incorporate them into a set of inputs, the SSWN would have to be augmented by introducing fuzzy logic components, for example. However, this is still under research.

Let us notice that most of the inputs are of an incremental type. Therefore, the same structure of the SSWN output is assumed. Hence,

$$\begin{aligned}
 y(k) &\triangleq \Delta \text{WIG20}(k|k-1) \\
 &\triangleq \frac{\text{WIG20}_{cl}(k|k-1) - \text{WIG20}_{cl}(k-1|k-1)}{\text{WIG20}_{cl}(k-1|k-1)} \\
 &= \frac{\text{WIG20}_{cl}(k || k-1) - \text{WIG20}_{cl}(k-1)}{\text{WIG20}_{cl}(k-1)},
 \end{aligned}$$

for one session ahead prediction, and

$$\begin{aligned}
 y(k) &\triangleq \Delta \text{WIG20}(k+4|k-1) \\
 &\triangleq \frac{\text{WIG20}_{cl}(k+4|k-1) - \text{WIG20}_{cl}(k-1|k-1)}{\text{WIG20}_{cl}(k-1|k-1)} \\
 &= \frac{\text{WIG20}_{cl}(k+4|k-1) - \text{WIG20}_{cl}(k-1)}{\text{WIG20}_{cl}(k-1)},
 \end{aligned}$$

for prediction five sessions ahead.

#### 4. Training of the SSWN

Having defined SSWN inputs and outputs, it remains to determine suitable values of network parameters, which are  $w_{i,j}$ ,  $i = 1, \dots, M + 1$ ,  $j = 1, \dots, L$  (see (1) and (2)),  $d_j, t_j$ ,  $j = 1, \dots, L$  (see (4) and (5)). This is done by using historical data and searching for parameters values such that the corresponding prediction error is minimal. The procedure is called network training (Zhang *et al.*, 2007; Borowa *et al.*, 2007). The SSWN structure and its optimised parameters for one session ahead prediction are illustrated in Fig. 5. The resulting SSWN is then validated by using different data sets in order to assess its generalisation properties. The parameter search is performed by solving an appropriately formulated optimisation problem. As the parameters are mixed-integer and the SSWN is described by a nonlinear mapping, solving the optimisation problem is a very challenging task for any known optimisation solver. Hence, we shall separate determining the number of states  $M$  and the number of wavelons  $L$ , which are the integer valued SSWN structure parameters, from determining the continuously valued parameters of the SSWN, which are  $W, d$  and  $t$ . The former will be presented later, and for the time being  $M$  and  $L$  are assumed known. The latter will be determined by applying a global stochastic optimiser being a further development of an algorithm presented in (Borowa *et al.*, 2007).

When preparing the training data, special attention needs to be paid to the impact of the network initial state

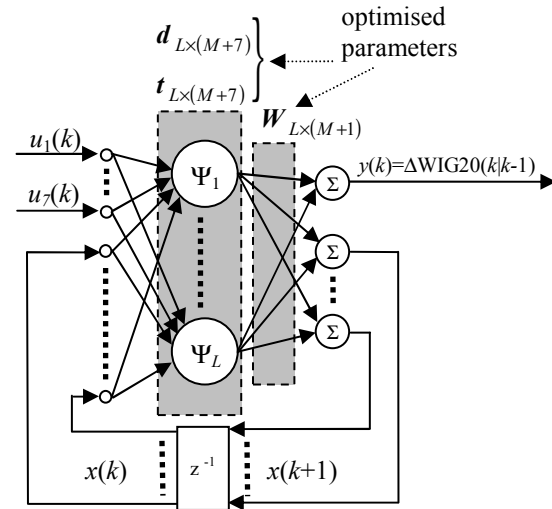


Fig. 5. Structure and optimised parameters of the SSWN for one session ahead prediction.

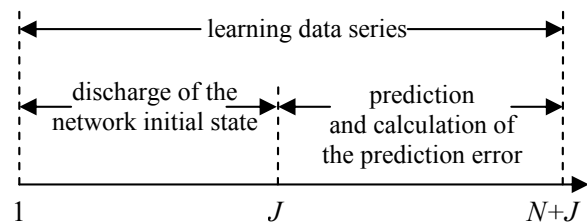


Fig. 6. Training data structure.

on its output. The SSWN state is artificial and there is no physical counterpart of it. It is used entirely to improve network approximation properties. Hence, the network output can be considered valid for prediction purposes only after a period of time needed to get the initial state discharged so that its impact on the output can be neglected. This requires that the SSWN be input to state stable (Khalil, 2002). Stability conditions expressed in terms of network parameters were derived for neural networks in, e.g., (Sanchez and Perez, 1999; Kulawski and Brdyś, 2000; Nguyen and Brdyś, 2006). This is still under research for our SSWN, for which stability was demonstrated by simulation for a wide range of the parameter values in (Borowa *et al.*, 2007). Let us denote the initial state discharge time by  $J$ . It is determined by simulation performed for representative network parameter sets. The training data series is then composed of historical trading sessions used to calculate the prediction errors and initial sessions during which the initial network state discharges, as illustrated in Fig. 6.

Let us now consider the training of the SSWN performing one session ahead prediction. Given the network parameters  $w, t, d$ , the prediction performance is evaluated over  $N$  consecutive sessions according to the follow-

ing performance function:

$$E(w, t, d) = \sum_{i=J+1}^{N+J} e_i^2(w, t, d), \quad (10)$$

$$e_i = y(i) - \frac{\text{WIG20}_{cl}(i) - \text{WIG20}_{cl}(i-1)}{\text{WIG20}_{cl}(i-1)}, \quad (11)$$

where  $i$  is a session number,  $e_i$  stands for the prediction error for session  $i$ , and  $E$  denotes the total prediction error over  $N$  sessions.

Optimising the performance function  $E(w, t, d)$  with respect to the parameters is performed by a simulated annealing solver, presented in the following section. The solver turned out to be efficient in training SSWN for modelling a wastewater treatment plant (Borowa *et al.*, 2007). Each iteration of the solver starts from discharging the network state initial condition for the actual parameter values. This is done by running the network over the first  $J$  sessions to discharge the initial condition applied at the beginning of Session 1 and determine the initial state corresponding to the actual parameters with which the prediction error over the next  $N$  session is evaluated. The same approach is applied to the training of the SSWN, which performs five sessions ahead prediction.

## 5. Application of simulated annealing to SSWN training

**5.1. Presentation of the simulated annealing solver.** Simulated Annealing (SA) was inspired by thermodynamics (statistical mechanics). The algorithm was motivated by the growing mechanism of a single crystal from a melt (Kirkpatrick *et al.*, 1983). It was found that slow cooling (annealing) of melted metal goes to a low energy state while fast cooling does not. The main part of the SA algorithm is the Metropolis Monte Carlo search, which was proposed in (Metropolis *et al.*, 1953). The annealing schedule was added in (Kirkpatrick *et al.*, 1983) in order to formalise the SA algorithm. In the process of optimisation, only one chain of atoms is used because SA convergence is not initial conditions dependent due to the exploring nature of the method.

**5.1.1. Metropolis Monte Carlo.** As opposed to many nonlinear programming algorithms, the Metropolis Monte Carlo (MMC) algorithm, as well as genetic algorithms, allows incorporating a worse solution in the search for a better solution. This algorithm works on a chain of atoms ( $S$ ), which is a chromosome equivalent in genetic algorithms. The chain of atoms is illustrated as

$$S = [s_1, s_2, \dots, s_q], \quad (12)$$

where  $s_i$  is a single atom and  $q$  is the number of atoms.

The MMC method is a modified Monte Carlo scheme. Whilst in the latter scheme a new solution is chosen fully randomly, in the former the new solution (new chain of atoms) is chosen randomly but from a bounded area and is compared with the old solution (Metropolis *et al.*, 1953). In each step only one, randomly chosen from the chain, atom ( $s_i$ ) is disturbed. Disturbing the single atom is performed according to

$$s_i \rightarrow s_i + \alpha\xi, \quad (13)$$

where  $\alpha$  is the maximum magnitude of the allowed disturbance and  $\xi$  is random number from the interval  $[-1, 1]$ .

After the new chain ( $S'$ ) has been chosen, its energy (cost function)  $E(S')$  is calculated. A difference  $\Delta E$  between the primary chain  $S$  and disturbed chain  $S'$  energies is calculated according to

$$\Delta E = E(S') - E(S). \quad (14)$$

If  $\Delta E < 0$ , then the new chain  $S'$  is accepted. Otherwise, the new chain is conditionally accepted with a probability given by the Boltzmann probability factor

$$P(\Delta E) = \exp\left(\frac{-\Delta E}{k_B T}\right), \quad (15)$$

where  $k_B$  is a Boltzmann constant and  $T$  is a temperature factor.

Whilst  $T$  is not real temperature and  $k_B$  is a constant, the product  $k_B T$  may be replaced for practical implementation with a single factor  $TP(\Delta E)$ , which is now compared against a randomly chosen number from the interval  $[0, 1]$ . If it is less than  $P(\Delta E)$ , then the configuration is retained. Otherwise the original chain is restored to start a new step.

**5.1.2. Simulated annealing algorithm.** MMC is an integral part of the simulated annealing algorithm. It follows from (15) that the probability of worse solution acceptance is a function of  $\Delta E$  as well as  $T$ . The factor  $T$  is called temperature. Kirkpatrick *et al.* (1983) proposed to first melt the system at a high temperature and then to optimise it at a given temperature. The temperature ought to be lowered slowly as long as the system is not frozen. A dedicated annealing schedule is used for lowering the temperature. The SA algorithm illustrated in Fig. 7 works iteratively as follows: for a given temperature ( $T_0$ ), the Metropolis Monte Carlo (MMC) method is applied. If the chain of atoms is said to be stable (optimised at a given temperature) then new  $T$  is computed and MMC is applied again; this procedure goes on until the temperature reaches zero or the energy of the system reaches an optimum. The temperature is updated according to the annealing schedule,

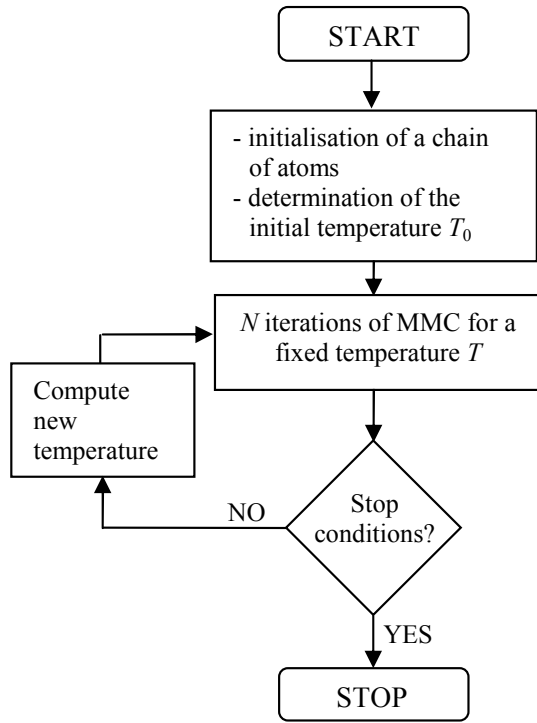


Fig. 7. General information structure of simulated annealing.

$$T(n) = f(T_0, n), \quad (16)$$

$$T(n + 1) < T(n) \quad (17)$$

is satisfied, where the condition. Here  $T_0$  is the initial temperature and  $n$  is the SA iteration counter (Kirkpatrick et al., 1983).

Kirkpatrick (1983) proposed a cooling schedule given by

$$T(n) = \left(\frac{T_1}{T_2}\right)^n \cdot T_0, \quad (18)$$

where  $T_0$  and  $T_1$  are cooling parameters. In (Hajek, 1988), a logarithmic cooling schedule was proposed as in

$$T(n) = \frac{T_0}{\log(1 + n)}, \quad (19)$$

and convergence conditions were derived for the SA algorithm. Few of the commonly used are also the cooling schedules

$$T(n + 1) = \gamma \cdot T(n), \quad T(1) = T_0, \quad (20)$$

$$T(n) = T_0 \cdot \exp(-\gamma n), \quad (21)$$

where  $\gamma$  is a cooling factor, and

$$T(n) = \frac{T_0}{1 + \ln(n)}, \quad (22)$$

cf. (Jacobson et al., 2005; Karafyllidis, 1999). The cooling parameters ( $T_0, T_1$  or  $\gamma$ ) must be selected appropriately to ensure the convergence conditions (Hajek, 1988; Locatelli, 2000).

Practically, the stability of a chain of atoms is verified by checking if the MMC iteration counter has reached a number specified in advance.

**5.1.3. Implementation of the simulated annealing solver.** The chain of atoms is composed of all network optimised parameters, that is

$$S = [w_1, \dots, w_{N_w}, d_1, \dots, d_{N_d}, t_1, \dots, t_{N_t}], \quad (23)$$

where  $d_i, i = 1, \dots, N_d$  are scaling parameters,  $t_i, i = 1, \dots, N_t$  are translation parameters, and  $w_i, i = 1, \dots, N_w$  are network output weights.

During training, a single atom is randomly selected at each step of the MMC algorithm with regard to constraints. The cooling is implemented according to

$$T(n) = \frac{T_0}{1 + \ln(n)}. \quad (24)$$

The initial temperature  $T_0$  was chosen to be equal to 0.005. MMC performs only one iteration between two temperature updates.

**5.2. Determining SSWN parameters.** Structure parameters are determined by directly searching over their value spaces. Giving the structure parameter value, the SA algorithm is applied to calculate optimised weights and wavelon parameters for selected training data. The graph in Fig. 8 illustrates the optimised training error in terms of the number of wavelons  $L$  under a fixed number of state variables  $M$ .

Training weights and wavelon processing parameters is long, and this needs to be considered by the predictor designer as parameter updates need to be performed in order to quickly respond to significant structural changes on the stock exchange. Therefore, the SSWN structure should not be too large and the training data time series length should not be too long. For example, an increase in the wavelon number implies an exponential increase in the training time. Comprehensive simulation trials have shown that a satisfactory trade-off between the training time and the resulting SSWN accuracy is achieved for the wavelon number between seven and twelve. A further increase in the number of wavelons does not improve the network accuracy but only increase the training time. Applying the same methodology, the number of states was determined. A good compromise between the training time and the network accuracy is achieved with the structure parameters as in Table 2. With these parameters, the SA solver finds a solution under the prescribed network accuracy for the one session ahead predictor in 70,000 iterations. This takes 120 minutes of computing time on an AMD Athlon 1900 Mhz processor. For five sessions ahead predictor, the training time is shorter due to a smaller structure of the predictor. The final structures of the one



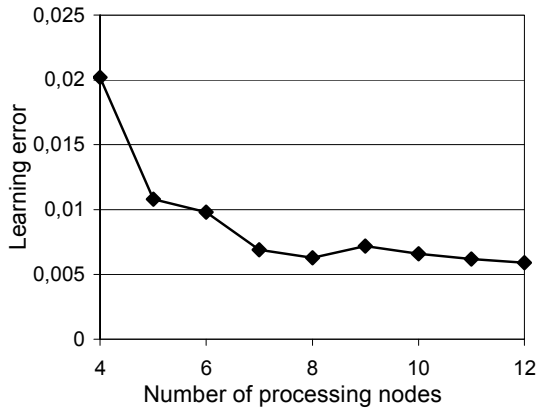


Fig. 8. Optimised training error vs. the number of wavelons under a fixed number of SSWN states.

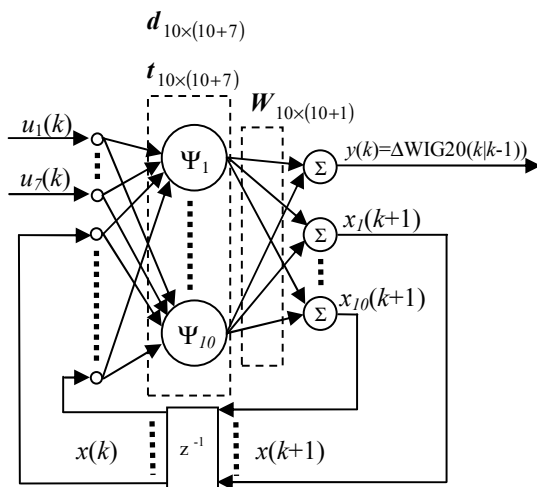


Fig. 9. Structure of the SSWN for one session ahead prediction of the WIG20 index.

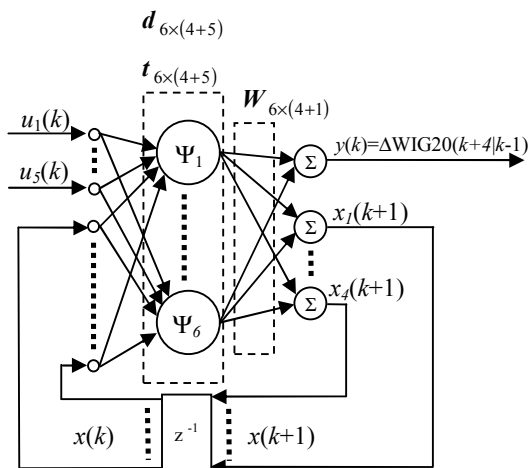


Fig. 10. Structure of the SSWN for five sessions ahead prediction of the WIG20 index.

Table 2. Selected structure parameters of the SSWN.

SSWN parameters	WSE-WIG forecast	
	1-session ahead	5-sessions ahead
Number of inputs	7	5
Number of wavelons	10	6
Number of states	10	4

step ahead predictor and five step ahead predictor are illustrated in Figs. 9 and 10, respectively.

### 6. Adaptive prediction

Preliminary validation of predictors on data different than those used for training have shown not entirely satisfactory results. This is mainly due to not including certain variables as the inputs, which have influence on the predicted index value. Some of them are not included as they are not measurable; the others have not been identified. The predictor trained on a selected data set accommodates these uncertainties in parameter values. If the uncertain inputs remain constant or they slowly vary, the predictor still performs well on different data sets. Otherwise, the parameters need to be updated on-line during the predictor operation. This leads to an adaptive predictor where initially the SSWN is trained off-line based on longer data sets as described earlier. The same training scheme is then applied on-line to update network parameters to actual values of the uncertain variables. However, the training performance function on  $E_k(w, t, d)$  at the instant  $k$  is now modified by introducing the weights with which the prediction errors during the previous sessions  $k, k = 1, \dots, k - N$  contribute to an overall prediction error over the last  $N$  trading sessions. Namely,

$$E_k = (w, t, d) = \sum_{i=k-N}^k \omega(i) e_i^2(w, t, d), \quad (25)$$

where

$$\omega(i) = \frac{2i}{N}, \quad (26)$$

and the  $i$ -th session  $e_i$  is defined as in (11)

The weights  $\omega(i)$  are linearly growing in time reaching the highest value for the last prediction error. Hence, the actual uncertainty input values are best accommodated into the resulting network parameter values  $w(k), t(k), d(k)$  obtained at the instant  $k$ . The optimisation solver starts from the parameters  $w(k-1), t(k-1), d(k-1)$  determined at the last time instant  $k-1$ . Hence, the on-line training time reduces. For the structure parameters as in Table 2, the number of iterations decreases from 70,000 to 15,000 and the computing time from 120 minutes to 25 minutes, for the one session ahead predictor. Hence, on-line parameter update is perfectly feasible to be performed

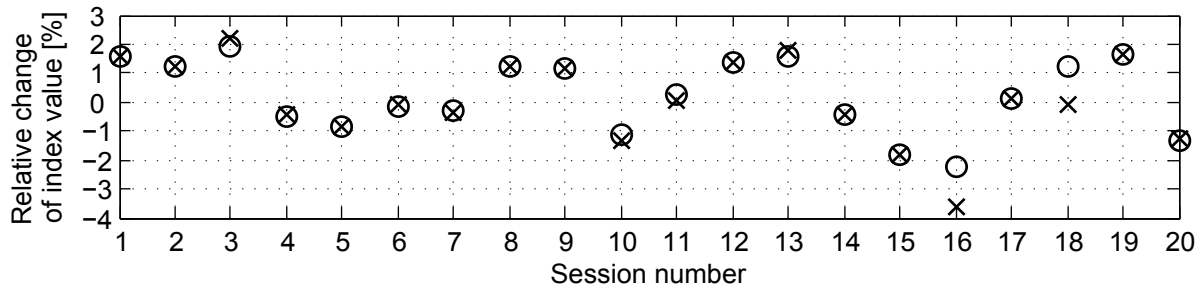


Fig. 11. Results of the validation of the one session ahead predictor in an incremental scale: circle—forecast; cross (x)—real value.

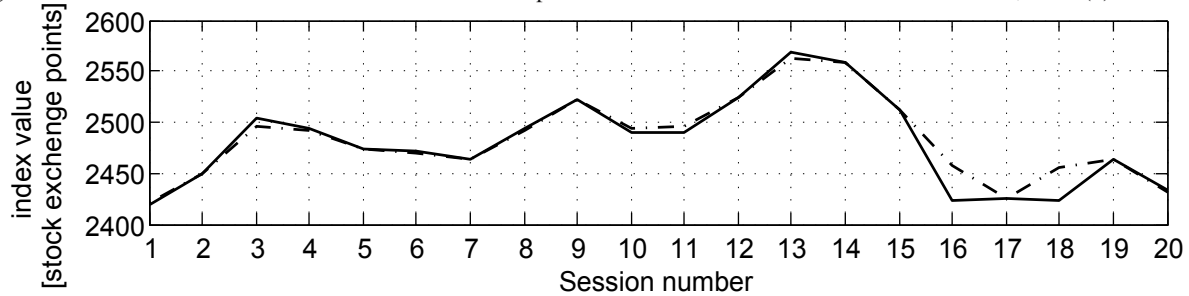


Fig. 12. Results of the validation of the one session ahead predictor in an absolute scale: solid line—real value; dashed line—forecast.

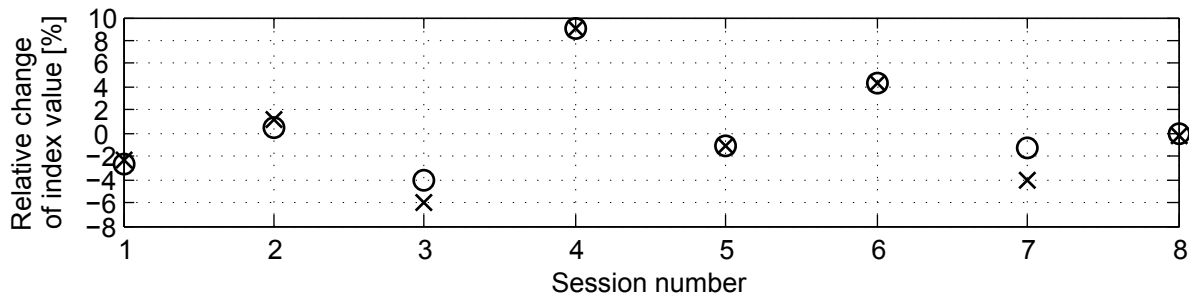


Fig. 13. Results of the validation of the five session ahead predictor in an incremental scale: circle—forecast; cross (x)—real value.

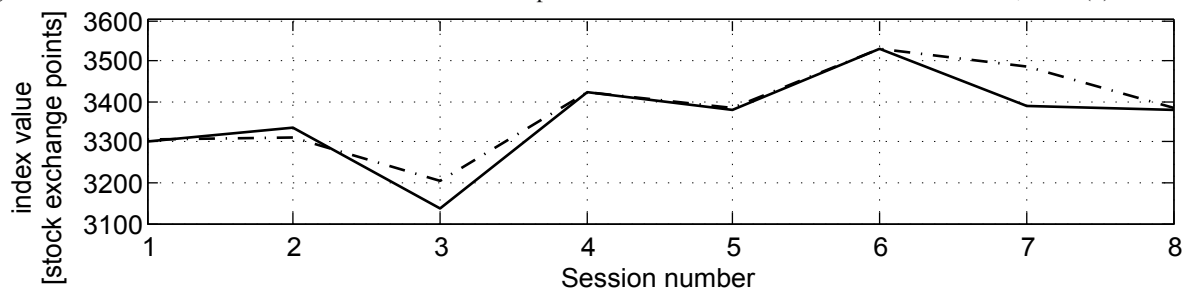


Fig. 14. Results of the validation of the five session ahead predictor in an absolute scale: solid line—real value; dashed line—forecast.

during the time periods remained between two trading sessions. Similar results were obtained for the five sessions ahead predictor.

### 7. Validation results

The designed SSWN based predictors were validated based on real data records from the Warsaw Stock Exchange.

The one session ahead predictor was validated based on data records composed of 20 subsequent sessions and gathered over the period of 15 September to 20 October 2005. The results obtained are illustrated in Figs. 11 and 12 by applying the incremental and absolute index value scales, respectively, and are summarised in Table 3.

The validation results show that the predictor can be effectively used to perform on-line one session ahead pre-

Table 3. Validation results of the accuracy of the one session ahead predictor.

Prediction accuracy measure	Units	
	[stock points]	[%]
Average prediction error	2.2	0.13
Maximal prediction error	7.8	0.36
Real average volatility of the index WIG20 over a session	23.3	1.04

Table 4. Validation results of the accuracy of the five session ahead predictor.

Prediction accuracy measure	Units	
	[stock points]	[%]
Average prediction error	38.6	1.30
Maximal prediction error	108.6	3.38
Real average volatility of the index WIG20 over 5 session	109.6	3.35

diction of the WIG20 index value. Notice that the average prediction error is one order smaller than the predicted index average change over one session.

The five sessions ahead predictor was validated based on data records gathered over the period of 24 July 2005 to 9 September 2006. The results obtained are illustrated in Figs. 13 and 14 in the incremental and absolute index value scales, respectively, and are summarised in Table 4.

Comparing the one session ahead and five session ahead prediction results, it can be seen that the average prediction error is greater for the latter. This is not surprising as the prediction horizon is longer in that case. However, the average prediction error is about three times smaller than the predicted index average change over five sessions. Hence, the predictor is still useful in performing on-line five sessions ahead prediction of the WIG20 index value. It seems that, first of all, introducing new inputs into the SSWN but not augmenting the network structure by new states and wavelons would certainly improve the prediction accuracy. However, identifying such inputs that are long term informative from the predicted index point of view and at the same time measurable (quantifiable) is much more difficult than in the one step ahead prediction case.

## 8. Conclusions

The paper has considered the prediction of the Warsaw Stock Exchange Price Index WIG20 price by using a newly developed artificial dynamic state space wavelet network model of the index price. The approach can be applied to developing tools for predicting changes of other economic indicators, especially stock exchange indices. Short and longer term predictors were designed for one

trading sessions ahead and five trading sessions ahead prediction. An adaptive mechanism was introduced into the off-line designed predictor in order to better incorporate into network parameters the uncertainty in market variables not included in the set of predictor inputs. The computing time due to parameter adaptation is not excessive so that on-line operation of predictors is perfectly feasible. Incorporating heterogenous knowledge into predictor inputs by using a fuzzy logic interface is under research.

## References

- Borowa, A., Brdyś, M.A. and Mazur, K. (2007). Modeling of wastewater treatment plant for monitoring and control purposes by state-space wavelet networks, *International Journal of Computers, Communications & Control* **II**(2): 121–131.
- Brdyś, M.A., Grochowski, M., Gminski, T., Konarczak, K. and Drewa, M. (2008). Hierarchical predictive control of integrated wastewater treatment systems, *Control Engineering Practice* **16**(6): 751–767.
- Grossmann, A. and Morlet, J. (1984). Decomposition of Hardy functions into square integrable wavelets of constant shape, *SIAM Journal on Mathematical Analysis* **15**(4): 723–736.
- Hajek, B. (1988). Cooling schedules for optimal annealing, *Mathematics of Operations Research* **13**(2): 311–329.
- Jacobson, S.H., Hall, S.N., Mclay, L.A. and Orosz, J.E. (2005). Performance analysis of cyclical simulated annealing algorithms, *Methodology and Computing in Applied Probability* **7**(2): 183–201.
- Karafyllidis, I. (1999). A simulator for single-electron tunnel devices and circuits based on simulated annealing, *Superlattices and Microstructures* **25**(4): 567–572.
- Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. (1983). Optimization by simulated annealing, *Science* **220**: 671–680.
- Khalil, H.K. (2002). *Nonlinear Systems*, Prentice Hall.
- Kulawski, G.J. and Brdyś, M.A. (2000). Stable adaptive control with recurrent networks, *Automatica* **36**(1): 5–22.
- Kuo, R.J., Chen, C.H. and Hwang, Y.C. (2001). An intelligent stock trading support system through integration of genetic algorithm based fuzzy neural network and artificial neural network, *Fuzzy Sets and Systems* **118**(1): 21–45.
- Locatelli, M. (2000). Convergence of a simulated annealing algorithm for continuous global optimization, *Journal of Global Optimization* **18**(3): 219–234.
- Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E. (1953). Equations of state calculations by fast computing machines, *Journal of Chemical Physics* **21**: 1087–1092.
- Mun, J. (2006). *Modeling Risk, Applying Monte Carlo Simulation, Real Options Analysis, Forecasting and Optimisation Techniques*, John Wiley & Sons, Inc.
- Nguyen, D.T. and Brdyś, M.A. (2006). Dynamic neural network identification and control under unmeasurable plant states, *Proceedings of the International Control Conference UKAC 2006, Glasgow, UK*.

- Qi, R. and Brdyś, M.A. (2005). Adaptive fuzzy modelling and control for discrete-time nonlinear uncertain systems, *Proceedings of the American Control Conference, ACC 2005, Portland, OR, USA*.
- Qi, R. and Brdyś, M.A. (2008). Stable indirect adaptive control based on discrete-time T-S fuzzy model, *Fuzzy Sets and Systems* **159**(8): 900–925.
- Sanchez, E.N. and Perez, J.P. (1999). Input-to-state stability analysis for dynamic neural networks, *IEEE Transactions on Circuits and Systems—I: Fundamental Theory and Applications* **46**(11): 1395–1398.
- Tsang, P.M., Kwok, P., Choy, S.O., Kwan, R., Ng, S.C., Mak, J., Tsang, J., Koong, K. and Wong, T. (2007). Design and implementation of NN5 for Hong Kong stock price forecasting, *Engineering Application of Artificial Intelligence* **20**(4): 453–461.
- Zamarreno, J.M. and Pastora, V. (1998). State space neural network. Properties and applications, *Neural Networks* **11**(6): 1099–1112.
- Zhang, G., Patuwo, B.E. and Hu, M.Y. (1998). Forecasting with artificial neural networks: The state of the art, *International Journal of Forecasting* **14**(1): 35–62.
- Zhang, Q. and Beneveniste, A. (1992). Wavelet networks, *IEEE Transactions on Neural Networks* **3**(6): 889–898.
- Zhang, Q. (1992). Wavelet network: The radial structure and an efficient initialization procedure, *Technical Report LiTH-isy-I-1423*, Linköping University.



**Mietek A. Brdyś** received the M.Sc. degree in electronic engineering, and the Ph.D. and D.Sc. degrees in control systems from the Institute of Automatic Control of the Warsaw University of Technology in 1970, 1974 and 1980, respectively. From 1974 to 1983, he held the posts of an assistant professor and an associate professor at the Warsaw University of Technology. In 1992 he became a full professor of control systems. Between 1978 and 1995, he held various

visiting faculty positions at the University of Minnesota, City University, De Montfort University and the University Polytechnic of Catalunya. In 1989, he became a senior lecturer at the School of Electronic, Electrical and Computer Engineering of the University of Birmingham. In 2001, he became a full professor of control systems at the Department of Automatic Control of the Gdańsk University of Technology. In 2008, he founded the Department of Control Systems Engineering at that University and became its head. He has served as a consultant for the Honeywell Systems and Research Centre in Minneapolis, GEC Marconi and Water Authorities in the UK, France, Spain, Germany and Poland. He is the head of the Interdisciplinary Research Network on Decision Support and Control Systems at the University of Birmingham and the head of the Intelligent Decision Support and Control System Group at the Gdańsk University of Technology. His research is supported by UK and Polish Research Councils, industry, and the European Commission. He is an author or co-author of about 220 refereed papers and six books.

His current research covers intelligent decision support and the control of complex uncertain systems, robust monitoring and control, softly switched robustly feasible model predictive control. The applications include environmental systems, technological processes, autonomous intelligent vehicles, and defence systems. He is a Chartered Engineer, a member of the IEE, a Senior Member of the IEEE, a Fellow of the IMA and the chair of the IFAC Technical Committee on Large Scale Complex Systems. He was the IPC chair of the 11th IFAC Symposium on Large Scale Complex Systems, held in Gdańsk in 2007.



**Adam Borowa** received his M.Sc. degree in control engineering in 2002 from the Electrical and Control Engineering Department of the Gdańsk University of Technology. Soon after that he became a Ph.D. student at this Department. During the period of 2001–2002 he served his apprenticeship at the Wastewater Treatment Plant in Swarzewo, Poland. Since 2002 he has been a member of the Intelligent Decision Support and Control System Group at

the Gdańsk University of Technology. He was a member of the organizing committee of the 11th IFAC Symposium on Large Scale Complex Systems, held in Gdańsk in 2007. He is a co-author of eight publications and four book chapters. He mainly focuses on the modelling and monitoring of large scale systems, especially processes with many time scales. In 2008 he received the Ph.D. degree in automatic control and robotics.



**Piotr Idźkowiak** received his M.Sc. degree in control engineering from the Electrical and Control Engineering Department of the Gdańsk University of Technology in 2007, and another M.Sc. degree in marketing and management from the Faculty of Management and Economics of the same University in 2008. After the studies he focused on improving production processes in a commercial company.



**Marcin T. Brdyś** received a B.Sc. degree in economics and an MBA degree from the University of Wales Cardiff Business School in 1997 and 1999, respectively. Since 2000 he has gained extensive experience working at European financial hubs, notably in the City of London, Frankfurt and Warsaw. The experience gained was working for some of the industry most revived financial institutions, including Schroder Salomon Smith Barney, European

Central Bank, Fimat Societe Generale International Banque SA and DZ Bank Polska SA. His specialising is in risk management as well as pricing the interest rate and equity derivatives. At the beginning of 2008 he received the internationally recognized Financial Risk Manager Certified qualification.

Received: 9 May 2008

Revised: 14 September 2008