Computer Intelligence in Modeling, Prediction, and Analysis of Complex Dynamical Systems

Ivan Zelinka, Ajith Abraham, Otto Rössler, Mohammed Chadli, René Lozi

Ivan Zelinka, Ajith Abraham, Otto Rössler, Mohammed Chadli, René Lozi. Computer Intelligence in Modeling, Prediction, and Analysis of Complex Dynamical Systems. Ivan Zelinka, Ajith Abraham, Otto Rossler, Mohammed Chadli, and Rene Lozi. Hindawi Publishing Corporation, 62 p., 2015, The Scientific World Journal, 10.1155/2015/948512 . hal-01338674

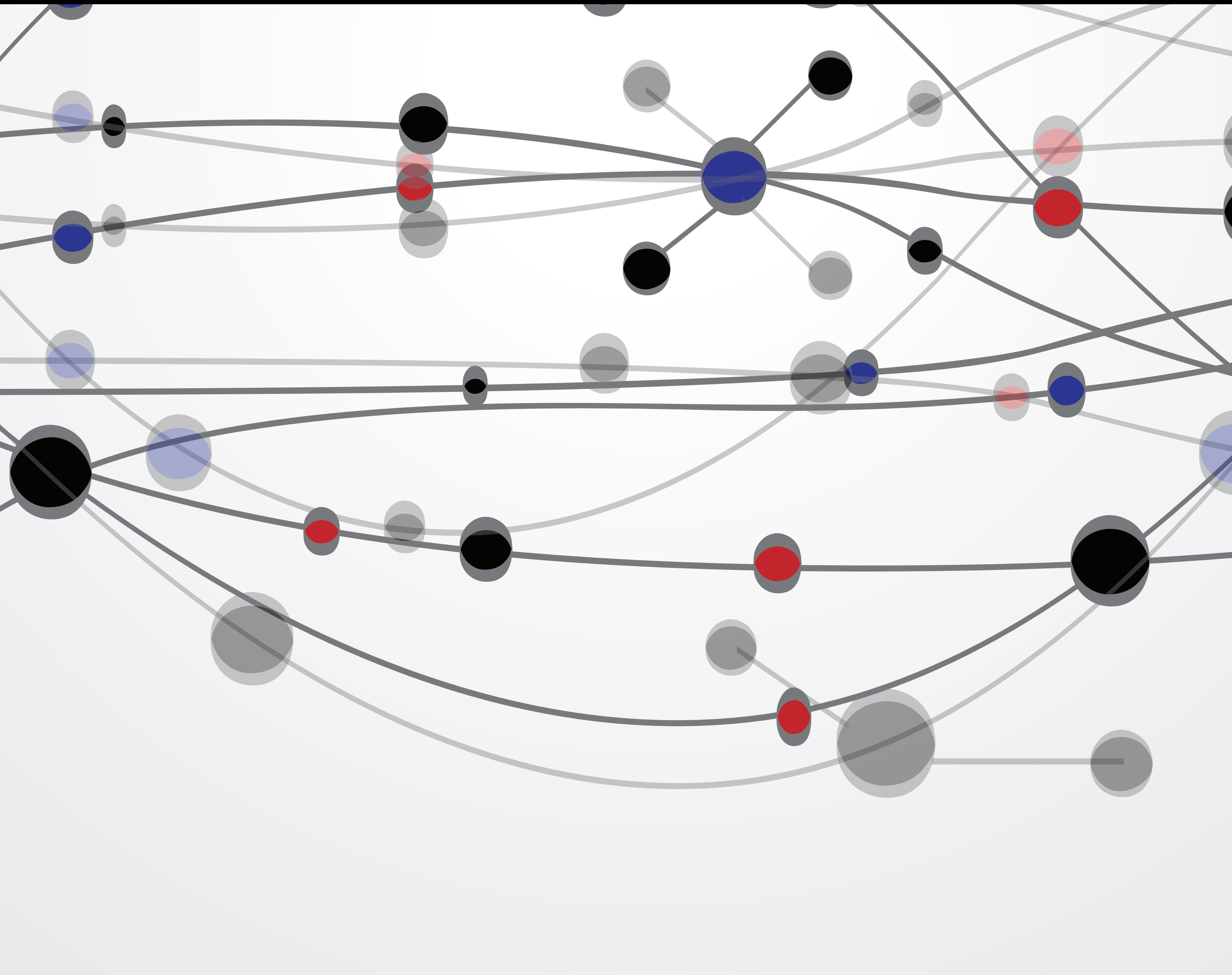
HAL Id: hal-01338674
https://hal.science/hal-01338674v1

Submitted on 15 Mar 2017

The Scientific World Journal

Computer Intelligence in Modeling, Prediction, and Analysis of Complex Dynamical Systems

Guest Editors: Ivan Zelinka, Ajith Abraham, Otto Rossler,
Mohammed Chadli, and Rene Lozi



Computer Intelligence in Modeling, Prediction, and Analysis of Complex Dynamical Systems

The Scientific World Journal

Computer Intelligence in Modeling, Prediction, and Analysis of Complex Dynamical Systems

Guest Editors: Ivan Zelinka, Ajith Abraham, Otto Rossler,
Mohammed Chadli, and Rene Lozi

Contents

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 948512, 1 page
http://dx.doi.org/10.1155/2015/948512

Editorial

Computer Intelligence in Modeling, Prediction, and Analysis of Complex Dynamical Systems

Ivan Zelinka,[1] Ajith Abraham,[1] Otto Rossler,[2] Mohammed Chadli,[3] and Rene Lozi[4]

[1] *VŠB-Technical University of Ostrava, Ostrava, Czech Republic*
[2] *University of Tubingen, Tubingen, Germany*
[3] *University of Picardie Jules Verne, Amiens, France*
[4] *Laboratoire J. A. Dieudonné, Université de Nice Sophia-Antipolis, Nice, France*

Correspondence should be addressed to Ivan Zelinka; ivan.zelinka@vsb.cz

Received 16 June 2015; Accepted 16 June 2015

Our technological civilization has had to confront numerous technological challenges such as finding the optimal solution of various problems including control technologies, power sources construction, and energy distribution amongst others. Analysis and prediction of complex system behavior such as stock exchange or other complex engineering structures and devices. Technology development of those and related areas has had and continues to have a profound impact on our civilization and lifestyle.

The topics discussed in this special issue belong to these mentioned areas and are mutually joined into a comprehensive text, which while discussing the specific selected topics gives a deeper insight to the interdisciplinary fusion of those modern and promising areas of emerging algorithms and technologies. This special issue discusses the mutual intersection of interesting fields of research, as artificial intelligence, unconventional algorithm, complex system behavior like chaos, soft computing, simulators, and software engineering amongst others. Novel techniques are also discussed in this special issue, which are able to handle tasks such as model and control of various systems, optimization by means standard and novel methods. Together with many interesting emerging technologies, a reader will also find in the special issue various mathematical and algorithmical methods used for proposed applications.

Therefore, this special issue is a timely volume to be welcome by the community focused on above mentioned techniques and beyond. This special issue is devoted to the studies of common and related subjects in intensive research fields of modern algorithms and their applications. For these reasons, we believe that this special issue will be interesting to scientists and engineers working in the above mentioned fields of research and applications.

Ivan Zelinka
Ajith Abraham
Otto Rossler
Mohammed Chadli
Rene Lozi

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 573068, 7 pages
http://dx.doi.org/10.1155/2015/573068

Research Article

Pattern Recognition Methods and Features Selection for Speech Emotion Recognition System

Pavol Partila, Miroslav Voznak, and Jaromir Tovarek

Department of Telecommunications, Faculty of Electrical Engineering and Computer Science, VSB-Technical University of Ostrava, 17 Listopadu 15, 70833 Ostrava, Czech Republic

Correspondence should be addressed to Miroslav Voznak; miroslav.voznak@vsb.cz

Received 27 August 2014; Accepted 27 October 2014

Academic Editor: Ivan Zelinka

The impact of the classification method and features selection for the speech emotion recognition accuracy is discussed in this paper. Selecting the correct parameters in combination with the classifier is an important part of reducing the complexity of system computing. This step is necessary especially for systems that will be deployed in real-time applications. The reason for the development and improvement of speech emotion recognition systems is wide usability in nowadays automatic voice controlled systems. Berlin database of emotional recordings was used in this experiment. Classification accuracy of artificial neural networks, k-nearest neighbours, and Gaussian mixture model is measured considering the selection of prosodic, spectral, and voice quality features. The purpose was to find an optimal combination of methods and group of features for stress detection in human speech. The research contribution lies in the design of the speech emotion recognition system due to its accuracy and efficiency.

1. Introduction

The development of applications and services is trying to deploy natural interaction between man and computer. Specifying commands by voice and movements is very popular nowadays. The majority of information is extracted from human speech with rather good accuracy. Human speech also includes secondary information, which holds properties of the speaker. Age, gender, emotional state, speech error, and other features are contained in human speech. The mentioned source characteristics are highly valuable, because speech features can be simulated only by person with good acting skills. As the title suggests, this paper describes a system for classifying emotional state of human speech. Emotion is one of the characteristics of human which describes his mental condition affecting physiological changes in the human body. These changes are also reflected in the human speech. Information about the emotional state is requested in many fields. Statistical evaluation of customer satisfaction and his interest in the products is evaluated by affected emotional state. This information is a direct response to any stimulus. Call center agents can be evaluated with regard to their work and access to the customer. There is a chance to train new agents and teach them to correct the procedure of communication with the customer. Human body influenced by stronger emotions is getting stressed. Sectors such as police, firemen, and especially military generate the greatest emotional pressure on employees. Dispatching orders can be directly influenced by information from speech emotion recognition system. Speech signal can serve as an authorization key in access systems. Speech is affected by physiological changes caused by changing emotions. An authorized user can be denied because authorization unit recognizes the stress speech as a wrong key. These are just the first examples of utilizations for speech emotion recognition systems. It is obvious that the system will have great application in human-machine interaction. Therefore it is appropriate to identify a classification ability of different classifiers for different emotional states. One of the related works, but a more extensive research summary from Mr. El Ayadi et al., is published in the article "survey on speech emotion recognition: features, classification schemes, and databases," which is mentioned in [1–4].

2. Speech Emotion Recognition System

System design consists of several blocks, which it distributed to major functions. Input values are represented by audio signals from created database and used for training and testing. Block diagram of the system is shown in Figure 1.

The quality of the input data, the audio signal in this case, has a direct impact on the classification accuracy. For this reason, the Berlin database containing over 500 recordings of actors consisting of men and women is used. The database contains 10 sentences in the seven emotional states. This corpus of recordings is considered as a high-quality database, because it was created by professional actors in the sound studio. Blocks *a*, *b*, and *c* represent point of view for emotion recognition. The system can be designed and used for detecting the stress of the speaker (option *a*), for recognition of all emotional states, as in the case of the Berlin database in which they are seven (option *b*). Other approaches to the problem are represented by block *c*. As mentioned, the speech signal has to be modified by routine preprocessing operations such as removing the DC component, preemphasis, and segmentation stochastic signal into quasiperiodic frames.

Speech recognition system is context independent, that is, taking into account only signal parameters, not content information. These parameters are the training and testing vectors for classifiers [5–7].

The calculation parameters are represented by the features extraction block that extracts the following:

(i) 39 Mel-frequency cepstral coefficients (MFCC) and dynamic parameters (first and second derivative of MFCC),

(ii) 12 linear prediction coefficients (LPC),

(iii) 12 linear spectral pairs (LSP),

(iv) 8 prosodic features (RMS energy, log-energy, zero crossing rate (ZCR), mean crossing rate (MCR), position of maximum, maximum, minimum, and harmonic-to-noise ratio (HNR) [8].

3. Classifiers

Individual research shows that it cannot be said which classifier for emotion recognition is the best. Each classifier or their combination achieved some results accuracy, which depends on several factors. The success of classifier is directly dependent on the data. This is derived from the fact that the accuracy varies with the data character such as the quantity, density distribution of each class (emotions), and the language too. One classifier has different results with acted database, where the density of each emotion is equitable and different with real data from call center where normal (calm) emotion state occupies 85 to 95 percent of all data. Appropriate choice of parameters has a considerable effect on the accuracy of these classifiers. The following subsections describe the used classification methods.

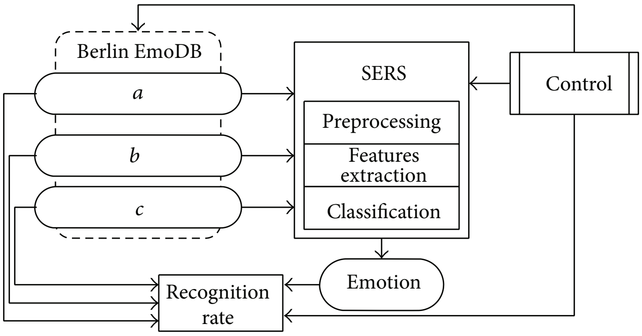


Figure 1: Block diagram of speech emotion recognition system (SERS). The system consists of a database that is used for training and testing and other blocks that describe functions of the algorithm. Different views on the issue are represented by blocks. (*a*) Stress versus neutral state classification, (*b*) each kind of emotion, and (*c*) other approaches. Scenario option is represented by control block.

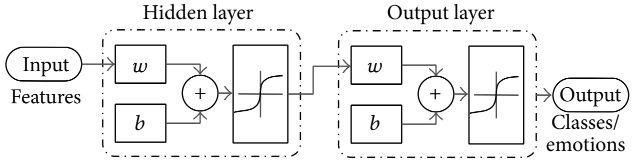


Figure 2: Artificial neural network architecture with hidden layers and output classes.

3.1. Artificial Neural Network. Our emotional state classification problem with high number of parameters can be considered as a pattern-recognition problem. In this case, two-layer feedforward network can be used. A two-layer feedforward network, with sigmoid hidden and output neurons, can classify vectors arbitrarily well, given enough neurons in its hidden layer. The network is trained with scaled conjugate gradient (SCG) backpropagation.

We shall denote the input values to the network by x_i where $i = 1, \ldots, d$. The first layer of network forms M linear combinations of these inputs to give a set of intermediate activation variables $a_j^{(1)}$:

$$a_j^{(1)} = \sum_{i=1}^{d} w_{ij}^{(1)} x_i + b_j^{(1)}, \quad j = 1, \ldots, M, \tag{1}$$

with one variable $a_j^{(1)}$ associated with each hidden unit. Here $w_{ij}^{(1)}$ represents the elements of first-layer weight matrix and $b_j^{(1)}$ is the *bias* parameters associated with the hidden units. Demonstration of such a network with speech features as an input, 5 hidden layers, and two output classes is shown in Figure 2.

SCG training implements mean squared error $E(w)$ associated with gradient ∇E and avoids the line search per learning iteration by using Levenberg-Marquardt approach in order to scale the step size. A weight in the network will be expressed in vector notation:

$$w = \left(\ldots, w_{ij}^{(1)}, w_{i+1j}^{(1)}, \ldots, w_{N_1 j}^{(1)} \theta_j^{(l+1)}, w_{ij+1}^{(1)}, w_{i+1j+1}^{(1)}, \ldots\right). \quad (2)$$

The vector-delta E points in the direction in which $E(w)$ will decrease at the fastest possible rate. Weight update equation is shown below, where c is suitable constant:

$$w(k+1) = w(k) - c\nabla E. \quad (3)$$

The gradient descent method for optimization is very simple and general. Only local information, for estimating a gradient, is needed for finding the minimum of the error function [9, 10].

3.2. k-Nearest Neighbour. The k-NN is a classification method on the principle of analogies learning. Samples from the training set are n numeric attributes, and each sample represents a point in N-dimensional space. This space of training samples is scanned by the classifier due to determining the shortest distance between training and unknown samples. Euclidean and other distances can be computed. In other words, an object is classified by a majority vote of its neighbours, with the object being assigned to the class most common amongst its k nearest neighbours (k is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of its nearest neighbour. The various distances between the vectors x_i and y_i are as follows:

$$d(X, Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}. \quad (4)$$

The neighbourhood distance is calculated through Euclidean metric. Given an m-by-n data matrix X, it is treated as m (1-by-n) row vectors $x_1, x_2, \ldots, x_m$.

3.3. Gaussian Mixture Model. A Gaussian mixture model is a parametric probability density function represented as a weighted sum of Gaussian component densities. GMMs are commonly used as a parametric model of the probability distribution of continuing measurements or features in biometric system, such as vocal tract, in speaker recognition systems as well. Probability distribution of the parameter vectors derived from human speech can be described using GMM:

$$p(o \mid \lambda^s) = \sum_{i=1}^{M^s} w_i^s p_i^s(o), \quad (5)$$

where M is the number of components for s class, w_i, $i = 1, \ldots, M$ are weights of components complying the condition that sum of all weights is 1, and p means the probability density of the components represented by the mean value and covariance matrix C_i. Gaussian model for class "s" is defined by (6)

$$\lambda^s = \{w_i^s, \mu_i^s, C_i^s\}, \quad i, \ldots, M^s. \quad (6)$$

The criterion of maximum likelihood depends on the probability density p and sequence parameters $O = (o_1, o_2, \ldots, o_n)$, as seen bellow [11, 12]:

$$\lambda^s = \arg\max p(o \mid \lambda^s). \quad (7)$$

4. Experiment

The aim of the experiment was to clarify the significance of chosen groups of features, as well as classification ability of selected classification methods for speech emotion recognition system. Samples of examination were formed from recordings of human speech with various emotional characters. The following settings and features were used in the experiment:

(i) input samples—Berlin database of emotional utterances:

  (a) 10 different sentences recorded by 10 different actors (both genders),
  (b) over 530 samples consisting of 7 emotions: *anger, boredom, disgust, fear, happiness, sadness, neutral state*;

(ii) feature extraction—computing of input vectors (speech parameters):

  (a) 13 MFCC coefficients $c_m = [c_m(0), \ldots, c_m(12)]$, 13 dynamic Δc_m, and 13 acceleration coefficients of MFCC $\Delta^2 c_m$,
  (b) 12 LPC coefficients,
  (c) 12 LSP coefficients,
  (d) 8 prosodic features;

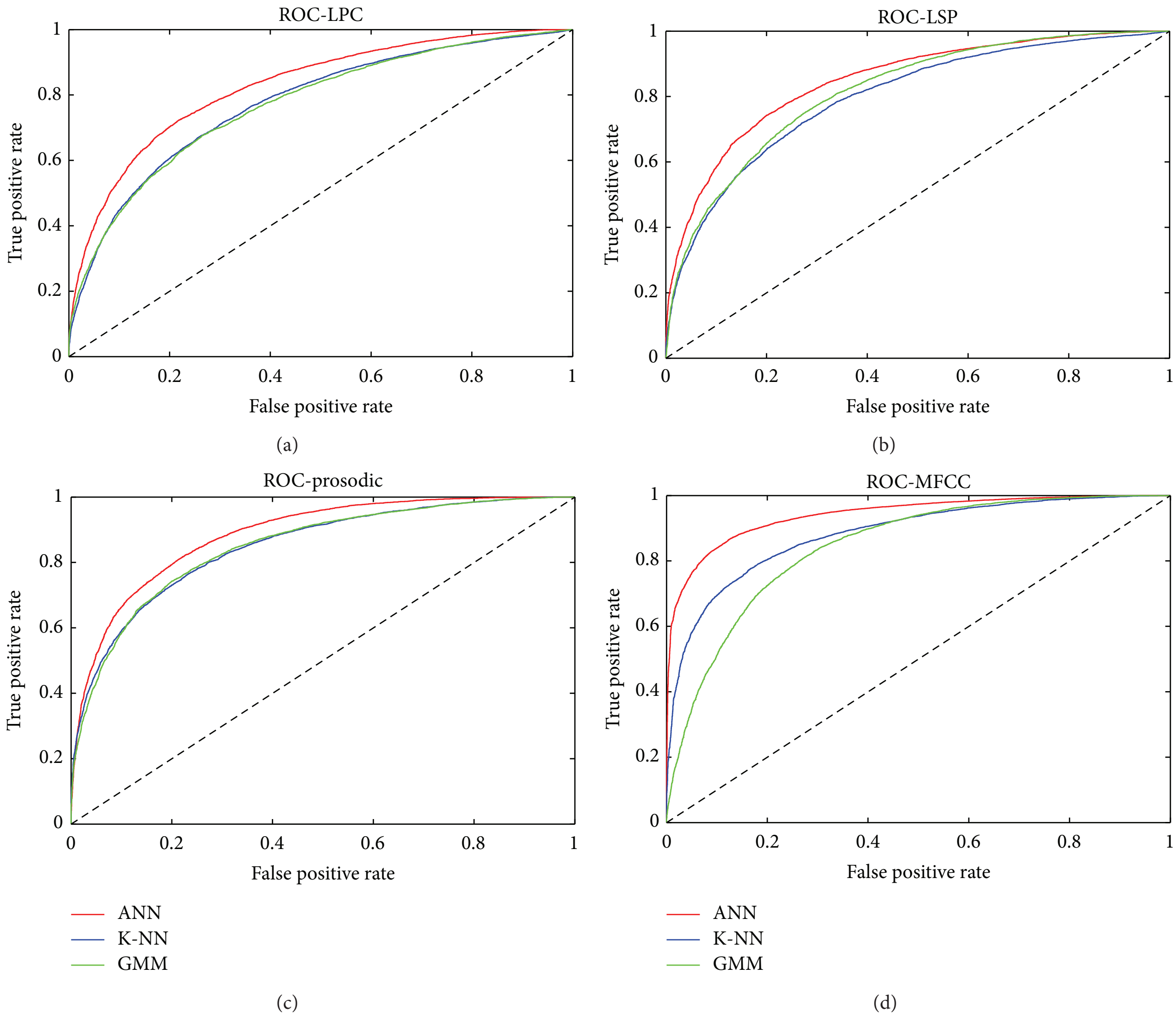
(iii) emotion classification:

  (a) GMM—64 mixture components,
  (b) k-nearest neighbours (set up 5 neighbours);
  (c) artificial neural network—feedforward backpropagation.

5. Result

One of the points of view is recognizing the stressed-out person, which means recognizing deviations from the neutral state. This state is not defined in the Berlin database. Therefore, it was necessary to assemble a set of data, the so-called "stress cocktail" from defined emotional states.

The stress of a person can be assembled from emotional states, other than neutral state. Emotions, anger and fear, were used to compile the stress data set with 50/50 ratio that these emotional states are reflected most often when a person is exposed to stressful situations. Fear and anger were selected because of the major sound differences from neutral state. Confusion matrices for each set of features are shown in Tables 2, 3, 4, and 5. The meaning of cells describes Table 1. True positive (TP) represents correctly classified first class



(a) (b) (c) (d)

Figure 3: Receiver operating characteristic of GMM, *k*-NN, and ANN classifier in neutral state versus stress recognizing for LPC, LSP, prosodic, and MFCC features.

Table 1: Confusion matrix: description of cells.

	Classifier		
Output classes	True positive	False positive	Positive predictive value
	False negative	True negative	Negative predictive value
	Sensitivity	Specificity	Precision
	Target classes		

(neutral) and true negative (TN) correctly classified second class (stress).

The GMM, *k*-NN, and the ANN were used to classify the stress versus neutral state. Results for all three classifiers are shown in Figure 3. The receiver operating characteristic (ROC) is applied for better system understanding. ROC curve is a tool for the evaluation and optimization of binary classification system (test), which shows the relationship between the sensitivity and specificity of the test or the detector for all possible threshold values [13].

The results in Tables 6, 7, and 8 describe classification accuracy for a particular type of classifier that has been trained by best-scored MFCC features of emotion pair. The classifier was trained by pair of emotions and values in the tables show tested ability of recognizing emotional state (left table header). All three classifiers showed the best recognition ability for the emotional state of anger. Emotional state of sadness was recognized with the evaluation very well. On the other hand, the worst-recognition ability of the system was the emotional state of fear (GMM and ANN) and disgust (ANN).

6. Conclusion

Neutral state versus stress scenario has been used for evaluating the accuracy of classification methods and features. The

Table 2: Performance results of classifiers for LPC features.

LPC	ANN			*k*-NN			GMM		
Neutral state	1718 5.1%	920 2.7%	65.1%	2233 6.6%	3241 9.6%	40.8%	5240 15.6%	6023 17.9%	46.5%
Stress	7542 22.5%	23410 69.7%	75.6%	7027 20.9%	21089 62.8%	75.0%	4020 12.0%	18307 54.5%	81.9%
	18.6%	96.2%	74.8%	24.1%	86.7%	69.4%	56.6%	75.2%	73.4%
	Neutral state	Stress		Neutral state	Stress		Neutral state	Stress	

Table 3: Performance results of classifiers for LSP features.

LSP	ANN			*k*-NN			GMM		
Neutral state	3466 13.4%	1814 7.0%	66.1%	3409 13.2%	3369 13.0%	50.3%	5778 17.2%	5452 16.2%	51.5%
Stress	5794 22.4%	14820 57.2%	71.7%	5851 22.6%	13265 51.2%	69.4%	3482 10.4%	18818 56.2%	84.4%
	37.4%	89.1%	70.6%	36.8%	79.7%	64.4%	62.4%	77.6%	73.4%
	Neutral state	Stress		Neutral state	Stress		Neutral state	Stress	

Table 4: Performance results of classifiers for prosodic features.

LSP	ANN			*k*-NN			GMM		
Neutral state	4470 13.3%	1699 5.1%	72.5%	3953 11.8%	2503 7.5%	61.2%	6483 19.3%	4774 14.2%	57.6%
Stress	4790 14.3%	22629 67.4%	82.5%	5307 15.8%	21825 65.0%	80.4%	2777 8.3%	19556 58.2%	87.6%
	48.3%	93.0%	80.7%	42.7%	89.7	76.7%	70.0%	80.4%	70.1%
	Neutral state	Stress		Neutral state	Stress		Neutral state	Stress	

Table 5: Performance results of classifiers for MFCC (dynamic and acceleration coefficients too) features.

MFCC	ANN			*k*-NN			GMM		
Neutral state	7445 22.1%	1901 5.7%	79.7%	4919 14.6%	2209 6.6%	69.0%	6305 18.8%	3507 10.4%	63.3%
Stress	1684 5.0%	22587 67.2%	93.1%	4210 12.5%	22279 66.3%	84.1%	2824 8.4%	20981 62.4%	88.1%
	81.6%	92.2%	89.3%	53.9%	91.0%	80.9%	69.1%	85.7%	81.2%
	Neutral state	Stress		Neutral state	Stress		Neutral state	Stress	

Table 6: Gaussian mixture model classification accuracy for different combinations of emotions [%].

Train 2/test	Train 1						
	Anger	Boredom	Disgust	Fear	Happiness	Sadness	Neutral state
Anger	—	91.7	85.7	83.4	70	96.4	90.8
Boredom	76.3	—	64.9	66.2	71.9	65.3	59
Disgust	59.1	64.3	—	62.5	56	78.5	64.7
Fear	52.4	72.6	59.7	—	47.2	81.8	70.4
Happiness	42.7	83.9	73.7	73.9	—	90.5	82.4
Sadness	87.1	67	75.5	73	85.5	—	75
Neutral state	78.6	53.3	69.1	63.9	74	65.7	—

Table 7: *K*-nearest neighbours classification accuracy for different combinations of emotions [%].

Train 2/test	Train 1						
	Anger	Boredom	Disgust	Fear	Happiness	Sadness	Neutral state
Anger	—	91.2	90.3	89.8	81.1	94.7	92.4
Boredom	92	—	70.4	67.7	75.9	61.6	64.6
Disgust	49	55	—	59.7	51.8	71.2	56.6
Fear	46.9	60.3	56.5	—	47.8	75.1	68.2
Happiness	31.9	78	72.7	73.1	—	86.1	78.7
Sadness	89.2	70.5	84.8	79.7	89.6	—	81.1
Neutral state	79.8	40.5	70.6	66	77.4	65.7	—

Table 8: Feed forward backpropagation neural network classification accuracy for different combinations of emotions [%].

Train 2/test	Train 1						
	Anger	Boredom	Disgust	Fear	Happiness	Sadness	Neutral state
Anger	—	95.8	93.7	92.4	87.9	98.1	96.7
Boredom	92	—	87.6	83.4	91.2	75.1	77.1
Disgust	79.8	79.1	—	68.2	77.7	86.4	77.6
Fear	69.6	70.6	73.7	—	68.2	82.2	76.2
Happiness	32.3	88.3	79.8	83.9	—	95	88.9
Sadness	97.9	81.5	92.6	95.3	97.5	—	85.2
Neutral state	93	49.9	86	85.1	88.2	52.4	—

results show that the most precise method for recognizing speech of the human stress state is an artificial neural network, which achieved the best results for all sets of parameters (90% for MFCC). The most significant feature for emotion classification is MFCC. This fact demonstrates accuracies of all the classifiers and the ratio of the sensitivity and specificity of the ROC curve shown in Figure 3. One of the reasons is the individuality of MFCC coefficients, which are not mutually correlated.

This experiment shows that these classification methods can be used on the recognition of emotional state. At the same time, the question arises: what emotional states will characterize stress. The answer will probably depend on which system would be applied. Another fact is that we cannot determine the intensity of emotionally stimulated Berlin database. One of the main tasks will be to compare these results with the emotional recordings of the realistic environmental conditions.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] M. Zarkowski, "Identification-driven emotion recognition system for a social robot," in *Proceedings of the 18th International Conference on Methods and Models in Automation and Robotics (MMAR '13)*, pp. 138–143, August 2013.

[2] S. Bakhshi, D. Shamma, and E. Gilbert, "Faces engage us: photos with faces attract more likes and comments on Instagram," in *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*, pp. 965–974, ACM, New York, NY, USA, 2014.

[3] M. A. R. Ahad, *Motion History Images for Action Recognition and Understanding*, Springer, London, UK, 2013.

[4] M. El Ayadi, M. S. Kamel, and F. Karray, "Survey on speech emotion recognition: features, classification schemes, and databases," *Pattern Recognition*, vol. 44, no. 3, pp. 572–587, 2011.

[5] S. G. Koolagudi and K. S. Rao, "Emotion recognition from speech: a review," *International Journal of Speech Technology*, vol. 15, no. 2, pp. 99–117, 2012.

[6] M. Voznak, F. Rezac, and J. Rozhon, "Speech quality monitoring in Czech national research network," *Advances in Electrical and Electronic Engineering*, vol. 8, no. 5, pp. 114–117, 2010.

[7] P. Partila, M. Voznak, M. Mikulec, and J. Zdralek, "Fundamental frequency extraction method using central clipping and its importance for the classification of emotional state," *Advances in Electrical and Electronic Engineering*, vol. 10, no. 4, pp. 270–275, 2012.

[8] F. Eyben, F. Weninger, M. Wollmer, and B. Schuller, open-SMILE—the Munich open Speech and Music Interpretation by Large Space Extraction toolk it, TU Munchen, 2013, http://opensmile.sourceforge.net/.

[9] T. Neuberger and A. Beke, "Automatic laughter detection in spontaneous speech Using GMM-SVM method," in *Text, Speech, and Dialogue*, vol. 8082 of *Lecture Notes in Computer Science*, pp. 113–120, 2013.

[10] J. Krajewski, S. Schnieder, D. Sommer, A. Batliner, and B. Schuller, "Applying multiple classifiers and non-linear dynamics features for detecting sleepiness from speech," *Neurocomputing*, vol. 84, pp. 65–75, 2012.

[11] S. Ntalampiras and N. Fakotakis, "Modeling the temporal evolution of acoustic parameters for speech emotion recognition," *IEEE Transactions on Affective Computing*, vol. 3, no. 1, pp. 116–125, 2012.

[12] H. Hu, M.-X. Xu, and W. Wu, "GMM supervector based SVM with spectral features for speech emotion recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '07)*, pp. IV-413–IV-416, Honolulu, Hawaii, USA, April 2007.

[13] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proceedings of the 23rd International Conference on Machine Learning (ICML '06)*, pp. 233–240, 2006.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 729080, 9 pages
http://dx.doi.org/10.1155/2015/729080

Research Article

Time Evolution of Initial Errors in Lorenz's 05 Chaotic Model

Hynek Bednář, Aleš Raidl, and Jiří Mikšovský

Department of Meteorology and Environment Protection, Faculty of Mathematics and Physics, Charles University in Prague, V Holešovičkách 2, 180 00 Prague, Czech Republic

Correspondence should be addressed to Aleš Raidl; ales.raidl@mff.cuni.cz

Received 11 August 2014; Accepted 24 September 2014

Academic Editor: Ivan Zelinka

Initial errors in weather prediction grow in time and, as they become larger, their growth slows down and then stops at an asymptotic value. Time of reaching this saturation point represents the limit of predictability. This paper studies the asymptotic values and time limits in a chaotic atmospheric model for five initial errors, using ensemble prediction method (model's data) as well as error approximation by quadratic and logarithmic hypothesis and their modifications. We show that modified hypotheses approximate the model's time limits better, but not without serious disadvantages. We demonstrate how hypotheses can be further improved to achieve better match of time limits with the model. We also show that quadratic hypothesis approximates the model's asymptotic value best and that, after improvement, it also approximates the model's time limits better for almost all initial errors and time lengths.

1. Introduction

Forecast errors in numerical weather prediction models (NWPMs) grow in time because of the inaccuracy of the initial state, chaotic nature of the weather system itself, and the model imperfections. Due to the nonlinear terms in the governing equations, the forecast error will saturate after some time. Time of saturation or *the limit of predictability of deterministic forecast* in NWPM is defined by [1] as time when the prediction state diverges as much from the verifying state as a randomly chosen but dynamically and statistically possible state. Forecasters also use other time limits to measure the error growth. *Forecast-error doubling time* τ_d is time when initial error doubles its size. $\tau_{95\%}$, $\tau_{71\%}$, $\tau_{50\%}$, and $\tau_{25\%}$ are times when the forecast error reaches 95%, 71%, 50%, and 25% of the limit of predictability. The time limit $\tau_{71\%}$ is the time when the forecast error exceeds $1/\sqrt{2}$ of the saturation or asymptotic value (AV) and, by [2], it corresponds to the level of climatic variability. Lorenz [3] calculated forecast error growth of NWPM by comparing the integrations of model, starting from slightly different initial states. Present-day calculations use the approach developed by Lorenz [4], where we can obtain two types of error growth. The first is called *lower bound* and is calculated as the root mean-square error (RMSE) between forecast data of increasing lead times and analysis data valid at the same time. The second is called *upper bound* and is calculated as the root mean-square (RMS) difference between pairs of forecasts, valid at the same time but with times differing by some fixed time interval. For example, if this interval is one day, the analysis for a given day is compared with one day forecast valid for the same day, and then this one day forecast is compared with two days forecast valid for the same day and so on. This second method compares only model equations and therefore it represents growth without model error. The innovation to upper bound, that is also used, is calculated as the RMS difference between forecast and control forecast with higher resolution of the model (*perfect model framework*).

Quadratic hypothesis (QH) was the first attempt that was made by Lorenz [3] to quantify the error growth. QH is based on the assumption that, if the principal nonlinear terms in the atmospheric equations are quadratic, then the nonlinear terms in the equations governing the field of errors are also quadratic. Dalcher and Kalney [5] added a model error to Lorenz's QH. A version that is used by recent researchers is the Simmons's et al. modification [6] of [5]. The Lorenz's QH is therefore suitable for upper bound of error growth and the Simons's et al. modification for lower bound. Trevisan et al. [7] came out with idea that logarithmic term is more valid than quadratic and linear term in the equations governing the field of errors, but this *logarithmic hypothesis* (LH) has never been used in NWPM computations.

Ensemble prediction systems (EPS) are used in order to estimate forecast uncertainties. They consist of a given number of deterministic forecasts where each individual forecast starts from slightly different initial states. EPS also includes a stochastic scheme designed to simulate the random model errors due to parameterized physical processes. Recent studies of predictability and forecast error growth (e.g., [8–11]) are mostly done by models of European Centre for Medium Range Weather Forecasts (ECMWF) and the Global Ensemble Forecast System (GEFS) from the National Centers for Environmental Prediction (NCEP). They include deterministic and ensemble forecast with 1 to 70 members. Operational model of ECMWF uses 50 members plus control forecast. More detailed study [10] uses 5 members plus control forecast. The initial conditions of ensemble members are defined by linear combination of the fastest singular vectors. Horizontal resolution with spectral truncation varies from T95 to T1279 and the number of vertical levels varies from 19 to 91 (analyses use higher resolution than forecasts). The output data are interpolated to 1° latitude × 1° longitude or 2.5° latitude × 2.5° longitude resolution separately for the Northern Hemisphere (20°, 90°) and Southern Hemisphere (−90°, −20°). Forecast is usually run for 90 days at winter (DJF) or summer (JJA) season with 0 (analysis) to 10, 15 (ECMWF), or 16 days (NCEP) of *forecast length* (FL) at 6 or 12 hours intervals. The most often used variable for analyzing the forecast error is geopotential height at 500 hPa level (Z500). Others are geopotential height at 1000 hPa level (Z1000) and the 850 hPa temperature (T850). To describe the forecast error growth over the calculated forecast length, the Simmons et al.'s modification [6] of Lorenz's QH [3] is used.

The questions that have arisen from studies of predictability and forecast error growth and that represent the key issues addressed in this work are: Is the LH [7] better approximation of initial error growth than QH [3]? Is there a possible modification of LH and QH that better approximates model data? If so, how much difference it creates in time limits that measure the forecast error growth? How precisely do the approximations describe forecast error growth over the FL (10, 15 or 16 days)? How do the approximations obtained from model values with various number of ensemble members differ from each other? Lorenz's chaotic atmospheric model (L05II) [12] will be used. For a more comprehensive introduction to the problem of weather predictability, we refer reader to the book by Palmer and Hagedorn [13]. After this introduction, Section 2 describes the model and experimental design, Section 3 describes ensemble prediction method, Section 4 introduces quadratic and logarithmic hypotheses, and Section 5 sets experimental designs. Section 6 presents the results and their discussion and Section 7summarizes the conclusions.

2. Model

Because of the limitations of NWPMs and because we want to derive the impact of initial error (perfect model framework), we use modification [13] of low-dimensional atmospheric model (L96). L96 [14] is a nonlinear model, with N variables $X_1, \ldots, X_N$ connected by governing equations:

$$\frac{dX_n}{dt} = -X_{n-2}X_{n-1} + X_{n+1}X_{n-1} - X_n + F. \qquad (1)$$

$X_{n-2}, X_{n-1}, X_n, X_{n+1}$ are *unspecified (i.e., unrelated to actual physical variables) scalar meteorological quantities*, F is a constant representing external forcing, and t is time. The index is cyclic so that $X_{n-N} = X_{n+N} = X_n$ and variables can be viewed as existing around a circle. Nonlinear terms of (1) simulate advection. Linear terms represent mechanical and thermal dissipation. The model quantitatively, to a certain extent, describes weather systems, but, unlike the well-known Lorenz's model of atmospheric convection [15], it cannot be derived from any atmospheric dynamic equations. The motivation was to formulate the simplest possible set of dissipative chaotically behaving differential equations that share some properties with the "real" atmosphere. NWPMs interpolate the output data mostly to 1° latitude × 1° longitude grid. In L96, it means $N = 360$. Such a high resolution would create large number of waves with similar maxima "pressure highs" and minima "pressure lows"; however, to share some properties with the "real" atmosphere, we would rather have 5 to 7 main highs and lows that correspond to planetary waves (Rossby waves) and a number of smaller waves that correspond to synoptic-scale waves. Therefore, we introduce spatial continuity modification (L05II) [12] of L96. Equation (1) is rewritten to the form:

$$\frac{dX_n}{dt} = [X, X]_{L,n} - X_n + F, \qquad (2)$$

where

$$[X, X]_{L,n} = \sum_{j=-J}^{J}{}' \sum_{i=-J}^{J}{}' \frac{\left(-X_{n-2L-i}X_{n-L-j} + X_{n-L+j-i}X_{n+L+j}\right)}{L^2}. \qquad (3)$$

If L is even, $\sum'$ denotes a modified summation, in which the first and last terms are to be divided by 2. If L is odd, $\sum'$ denotes an ordinary summation. Generally, L is much smaller than N and $J = L/2$ if K is even and $J = (L-1)/2$ if L is odd. For our computation, we choose $N = 360$, so each sector covers 1° degrees of longitude. To keep a desirable number of main pressure highs and lows, Lorenz suggested keeping ratio $N/L = 30$ and therefore $L = 12$. Parameter $F = 15$ is selected as a compromise between too long doubling time (smaller F) and undesirable shorter waves (larger F). We first choose arbitrary values of the variables X, and, using a fourth order Runge-Kutta method with a time step $\Delta t = 0.05$ or 6 *hours*, we integrate forward for 14400 *steps*, or 10 *years*. We then use the final values, which should be free of transient effect. Figure 1 shows values of model variables with selected parameters. For this setting and by the method of numerical calculation presented in [16], the global largest Lyapunov exponent is $\lambda_{\max} = 0.32$. The definition of a chaotic system according to [3] states, that a bounded dynamical system with a positive Lyapunov

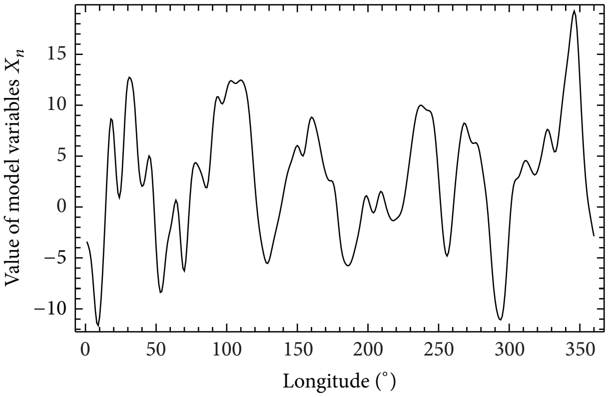


FIGURE 1: An example of longitudinal structure of model variable X.

exponent is chaotic. Because the value of the largest Lyapunov exponent is positive and the system under study is bounded, it is chaotic. Strictly speaking, we also need to exclude the asymptotically periodic behavior, but such a task is impossible to fulfill for the numerical simulation. The choice of parameters F and *time unit* = 5 *days* is made to obtain similar value of the largest Lyapunov exponent as state of the art NWPMs.

3. Ensemble Prediction Method

The ensemble prediction method (EPM) employed is similar to [14] and is used to calculate average initial error growth. We make an initial "run" by choosing error e_{n0} and letting $X'_{n0} = X_{n0} + e_{n0}$ be the "observed" initial value of N variables. We then integrate forward from the true and the observed initial state, for between 25 and 37.5 days ($K = 100$ to $K = 150$ steps). This time length covers initial error growth till the limit of predictability. We obtain N sequences $X_{n0}, \ldots, X_{nK}$ and $X'_{n0}, \ldots, X'_{nK}$, after which we let $e_{nk} = X'_{nk} - X_{nk}$ for all values of k and n. In NWPM, forecast error growth is obtained from an average of values from 90 days and from various number of ensemble members. To simulate that, we make a total of $M_1 = 100$, $M_2 = 250$, and $M_3 = 500$ runs in the above described manner. In each run, new values of X_{n0} are set as the old values of X_{nK}. Finally, we let $e^2(\tau) = 1/N(e_{1k}^2 + \cdots + e_{Nk}^2)$ be the average of the N values, where $\tau = k\Delta t$ is the predictable range and $\log E^2(\tau) = 1/M(\log e^2(\tau)_1 + \cdots + \log e^2(\tau)_M)$ is the average of M values. Logarithmic average is chosen because of its suitability for comparison with growth governed by the largest Lyapunov exponent. For further information, see [17–19].

4. Error Growth Hypotheses

According to Lorenz [14], there is an eventual cessation of the exponential error growth due to processes represented by nonlinear terms in the weather governing equations. Most important are the quadratic terms, which represent the advection of the temperature and velocity fields. Under the assumption that the principal nonlinear terms in the atmospheric equations are quadratic, nonlinear terms in equations governing the field of errors are also quadratic. To describe this, Lorenz [14] defined QH

$$\frac{dE(t)}{dt} = aE(t) - bE(t)^2, \tag{4}$$

where $E(t)$ is a distance at time t between two originally nearby trajectories and a and b are constants. As an alternative, Trevisan et al. [7] introduced LH

$$\frac{dE(t)}{dt} = -cE(t)\ln(gE(t)), \tag{5}$$

where c and g are constants. The explanation follows, if we let $Q(t) = \ln(\overline{E(t)})$ and $\overline{E}$ is the normalized E, then $dQ(t)/dt = a(1 - e^{Q(t)})$ represents the QH. In [7], it is assumed that linear fit $dQ(t)/dt = -aQ(t)$ is superior to the QH.

As modifications of QH (4), we use Simmons's et al. [6] version (QHM),

$$\frac{dE(t)}{dt} = a_m E(t) - b_m E(t)^2 + s_m, \tag{6}$$

that is used for approximation of growth with combination of initial and model error and where a_m, b_m and s_m are constants. We also add the constant term to LH (5) (LHM)

$$\frac{dE(t)}{dt} = -c_m E(t)\ln(g_m E(t)) + l_m, \tag{7}$$

where c_m, g_m, l_m are constants. The reason for choosing (6) and (7) is based on assumption that, at $t = 0$, $dE(t)/dt = 0$ for ((6) and (7)) but $dE(t)/dt \neq 0$ for model data. By adding the constant term, we may solve this difference.

In [7, 20, 21], it is shown on low-dimensional models that if the initial error is sufficiently small and therefore the early stages of error growth are exponential, QH is superior. If the initial error is not small enough, it is better to use LH. Generally, whether an error is small enough to guarantee the exponential growth depends on specific meteorological conditions and/or model under study.

5. Experimental Design

We want to achieve the conditions as similar to NWPM as possible. The size of initial error for NWPM (perfect model framework) is by [9] approximately between 2% and 0.01% of AV of the forecast error for control forecast and between 10% and 3% of AV for ensemble members. Different values of AV fraction are a result of varying resolution and because it is calculated for different variables (Z500, Z1000, and T850). In our case, the AV is $E_{\text{asym}} = 8.4$. This is calculated by four independent methods with the same results. The first method is numerical computation of ensemble prediction approach. Second and third methods are based on formula:

$$E_{\text{asym}} = \overline{(f_1 - X_{\text{avr}})^2} + \overline{(f_2 - X_{\text{avr}})^2} = 2\overline{(f_1 - X_{\text{avr}})^2}, \tag{8}$$

where f_1 is "forecast" from X_{n0}, f_2 from X'_{n0}, and X_{avr} is average value of X_n. The bars above the (8) members mean the average value. The explanation for (8) can be found in [6, 10]. The fourth method is based on assumption [2] that variability of X_n is 71% of E_{asym}.

Recalculation of initial errors to L05II model leads to the sizes of X_n between 0.001 and 0.84. For initial error sizes e_{n0}, we therefore choose randomly from five normal distributions $\mathrm{ND}(\mu;\sigma)$. $\mathrm{ND}_1 = (0; 0.1)$, $\mathrm{ND}_2 = (0; 0.2)$, $\mathrm{ND}_3 = (0; 0.4)$, $\mathrm{ND}_4 = (0; 0.6)$, $\mathrm{ND}_5 = (0; 1)$, where μ is mean and σ is standard deviation. These choices of e_{n0} are made, because [20, 21] shows that change over QH and LH takes place between $e_{n0} = 0.1$ and $e_{n0} = 1$ for L96. NWPM routinely defines initial conditions of ensemble members by the fastest singular vectors. We do not adopt it, because, by [10, 22], it affects early stages of forecast error growth and, in our case, we want to have model data as close as possible to the tested hypotheses. From these initial conditions, the average initial error growth E is calculated from ensemble prediction method by the fourth order Runge-Kutta integration schema with a time step $\Delta t = 0.05$ or 6 *hours* for $M_1 = M_2 = 100$, $M_3 = M_4 = 250$, and $M_5 = 500$. Because we want to study agreement of ((4)–(7)) with model data, we make differences of model data $y_k = (E(\tau + \Delta t) - E(\tau))/\Delta t$ at points $x_k = (E(\tau) + E(\tau + \Delta t))/2$, $k = 1, \dots, K$ and $K = 56$ ($\tau_1 = 14$ *days*), $K = 76$ ($\tau_2 = 19$ *days*), $K =$ *limit of predictability* (τ_3), and we calculate parameters a, b, c, g, a_m, b_m, s_m, c_m, g_m and l_m. The choice of the first two values of K is made, because we want to keep ratio $\tau_{95\%}$/*forecast length* the same for NWPM and L05II.

The solutions of ((4)–(6)) are

$$E(t) = \frac{a}{((a/e_0) - b)\exp(-at) + b}, \tag{9}$$

$$E(t) = \frac{(ge_0)^{\exp(-ct)}}{g}, \tag{10}$$

$$\begin{aligned} E(t) = {} & \frac{1}{2b_m}\left(a_m - \sqrt{-a_m^2 + 4b_m s_m}\right) \\ & \times \tan\left(\frac{1}{2}\sqrt{-a_m^2 + 4b_m s_m}\,t\right. \\ & \left.\left. - \arctan\left(\frac{-a_m + 2b_m e_0}{\sqrt{-a_m^2 + 4b_m s_m}}\right)\right)\right). \end{aligned} \tag{11}$$

LHM (7) cannot be solved analytically and therefore it is solved numerically by the fourth order Runge-Kutta integration schema with a time step $\Delta t_m = 0.25$. We have five types of normal distribution for reaching sizes of initial error $\mathrm{ND}_{1,\dots,5}$, five settings for EPM $M_{1,\dots,5}$, three FL $\tau_{1,\dots,3}$, and five ways of getting data of initial error growth: EPM, ((9)–(11)) and numerical solution of (NS (7)). To answer the key questions, we compute time limits for all combinations. We take M_{1-5}, τ_3, EPM as the most reliable dataset in our experiment for all e_0 and we calculate differences with other combinations at the same time limits.

6. Results and Discussion

Tables 1 and 2 show with darker grey rows the resulting values ($\overline{M}$, τ_3, EPM) for all time limits and for all e_0 represented by $\mathrm{ND}_{1,\dots,5}$. $\overline{M}$ is average value of $M_{1,\dots,5}$ and we use it, because the difference between $M_{1,\dots,5}$ is of the order of 0.1 and M_3 and M_4 do not show closer values to M_5 than M_1 and M_2. Lines in Tables 1 and 2 marked ($\overline{M}$, EPM, τ_3) − ($\overline{M}, \alpha, \beta$) where α successively represents (9), (10), (11), and NS (7), and $\beta = \tau_{1,\dots,3}$ show the difference between most reliable data ($\overline{M}$, τ_3, EPM) and data from combinations of α, β. Columns marked by ε_d, $\varepsilon_{25\%}$, $\varepsilon_{50\%}$, $\varepsilon_{71\%}$, and $\varepsilon_{95\%}$ display standard deviation of $\overline{M}$. Last third of Table 2 shows average values $\overline{\mathrm{ND}}$ of $\mathrm{ND}_{1,\dots,5}$ with standard deviations σ_d, $\sigma_{25\%}$, $\sigma_{50\%}$, $\sigma_{71\%}$, and $\sigma_{95\%}$. From Tables 1 and 2, we can see that QH with solution (9) has an almost constant difference (EPM, τ_3) − (α, β) equal on average $\overline{\mathrm{ND}}$ to value between 2.3 and 2.5 days with σ between 0.1 and 0.3 days. The value of difference (EPM, τ_3) − ((9), τ_1) is higher only for $\tau_{95\%}$. LH with solution (10) does not give good fit to model values and it is the worst choice if we compare it with other approximations. QHM with solution (11) has an almost constant difference (EPM, τ_3) − (α, β) equal on average $\overline{\mathrm{ND}}$ to value between 1.5 and 1.9 days with σ between 0 and 0.2 days. Different from these averages are values of difference (EPM, τ_3) − ((11), τ_2) at $\tau_{95\%}$ and values of difference (EPM, τ_3) − ((11), τ_1) at $\tau_{71\%}$. Values of difference (EPM, τ_3) − ((11), τ_1) at $\tau_{95\%}$ are not displayed, because $E_{\mathrm{asym},\tau 1}$ is smaller than model value at $\tau_{95\%}$ (Figures 2, 3, and 4). The solution NS (7) of LHM does not exist for $\mathrm{ND}_{1,2}$ and $\tau_{2,3}$ (Figure 2). For others ND and τ, there is also an almost constant difference (EPM, τ_3) − (α, β) equal on average $\overline{\mathrm{ND}}$ to value between 1.3 and 1.5 days with σ between 0 and 0.2 days. Different from these averages are values of difference (EPM, τ_3) − (NS (7), τ_2) at $\tau_{95\%}$ and values of difference (EPM, τ_3) − (NS (7), τ_1) at $\tau_{71\%}$ and $\tau_{95\%}$. Let us first take a look on constant differences and their source. Figures 2 to 4 display time variations of the average prediction error E for $\mathrm{ND}_{1,3,5}$. In these figures, we can see that, in contrast to approximations, the model values show negative growth rate for the first day, but turning into increase thereafter. At around two days, the model values reach the same value as it had initially. NWPMs also show this type of behavior [23] and approximation cannot capture that.

To summarize our findings, even though solutions (9) and NS (7) give better approximations to model data than (7) for all $\tau_{1,\dots,3}$, they have major disadvantages. Solution (9) underestimates the model data for τ_2, $\tau_{95\%}$, and τ_1, $\tau_{95\%}$ and therefore we cannot calculate the time when this approximation reaches 95% of E_{asym}. Solution NS (7) does not exist for $\mathrm{ND}_{1,2}$ and $\tau_{2,3}$. If we subtract two days (time when the model values reach the same value that they had initially) from ($\overline{M}$, EPM, τ_3) − ($\overline{M}, \alpha, \beta$), (7) would become superior, because we would get similar result as (9) and NS (7) but without the above mentioned disadvantages. One may argue that, because of subtraction of 2 days, we should recalculate the approximations. We did that and the results are close to the ones with subtraction. It is also good to mention that τ_1 is always higher than $\tau_{25\%}$ and lower than $\tau_{95\%}$ and τ_2 is always

TABLE 1: Average values over $M_{1,\ldots,5}$ of time limits (in days) for model values (EPM), for normal distributions $\mathrm{ND}_{1,\ldots,3}$, and for prediction time length τ_3 (bold rows). Difference between this model values and average values over $M_{1,\ldots,5}$ received from (9)–(11) and NS (7) with parameters a, b, c, g, a_m, b_m, s_m, c_m, g_m, and l_m calculated from approximations of (4)–(7) for τ_1, τ_2, τ_3, and $M_{1,\ldots,5}$ (columns τ_d, $\tau_{25\%}$, $\tau_{50\%}$, $\tau_{71\%}$, $\tau_{95\%}$). Standard deviation of $\overline{M}$ (columns ε_d, $\varepsilon_{25\%}$, $\varepsilon_{50\%}$, $\varepsilon_{71\%}$, $\varepsilon_{95\%}$).

ND_1

(Days)	τ_d	ε_d	$\tau_{25\%}$	$\varepsilon_{25\%}$	$\tau_{50\%}$	$\varepsilon_{50\%}$	$\tau_{71\%}$	$\varepsilon_{71\%}$	$\tau_{95\%}$	$\varepsilon_{95\%}$
(EPM, τ_3)	**4.8**	**0.1**	**12.9**	**0.1**	**16.3**	**0.1**	**19.2**	**0.2**	**25.8**	**0.2**
(EPM, τ_3) − ((9), τ_3)	2.6	0	2.4	0.3	2.4	0.2	2.4	0.3	1.9	0.3
(EPM, τ_3) − ((10), τ_3)	3.8	0	6.4	0.3	6.2	0.2	5.6	0.3	5.1	0.9
(EPM, τ_3) − ((11), τ_3)	2.2	0.2	2	0.4	1.9	0.3	1.9	0.3	1.4	0.6
(EPM, τ_3) − (NS (7), τ_3)	—	—	—	—	—	—	—	—	—	—
(EPM, τ_3) − ((9), τ_2)	2.6	0	2.4	0.2	2.4	0.3	2.5	0.3	2.6	1.2
(EPM, τ_3) − ((10), τ_2)	3.3	0	4.5	0.2	4.3	0.2	4.5	0.3	8.1	0.2
(EPM, τ_3) − ((11), τ_2)	2.1	0.1	1.8	0.1	1.8	0.1	1.9	0.3	2.5	0.6
(EPM, τ_3) − (NS (7), τ_2)	—	—	—	—	—	—	—	—	—	—
(EPM, τ_3) − ((9), τ_1)	2.5	0.1	2.5	0.1	2.6	0.2	2.8	0.6	5.1	2.9
(EPM, τ_3) − ((10), τ_1)	—	—	—	—	—	—	—	—	—	—
(EPM, τ_3) − ((11), τ_1)	1.7	0.2	1.5	0.2	1.3	0.4	0.1	0.5	—	—
(EPM, τ_3) − (NS (7), τ_1)	1.3	0	1.1	0	1.3	0.1	2	0.3	6.6	0.4

ND_2

(Days)	τ_d	ε_d	$\tau_{25\%}$	$\varepsilon_{25\%}$	$\tau_{50\%}$	$\varepsilon_{50\%}$	$\tau_{71\%}$	$\varepsilon_{71\%}$	$\tau_{95\%}$	$\varepsilon_{95\%}$
(EPM, τ_3)	**4.9**	**0.1**	**10.6**	**0.1**	**14**	**0.2**	**16.9**	**0.2**	**24.3**	**0.5**
(EPM, τ_3) − ((9), τ_3)	2.6	0	2.5	0.3	2.6	0.4	2.7	0.4	3	0.4
(EPM, τ_3) − ((10), τ_3)	3.6	0	4.6	0.1	5.1	1.1	4.5	0.1	4.4	1
(EPM, τ_3) − ((11), τ_3)	1.9	0.4	1.6	0.5	1.6	0.4	1.7	0.4	2.1	0.4
(EPM, τ_3) − (NS (7), τ_3)	—	—	—	—	—	—	—	—	—	—
(EPM, τ_3) − ((9), τ_2)	2.6	0	2.5	0.2	2.6	0.3	2.6	0.3	2.2	1.4
(EPM, τ_3) − ((10), τ_2)	3.5	0.1	4.3	0.2	4	0.3	3.8	0.3	7.2	0.4
(EPM, τ_3) − ((11), τ_2)	1.9	0.3	1.6	0.3	1.6	0.3	1.6	0.3	−0.4	2.5
(EPM, τ_3) − (NS (7), τ_2)	—	—	—	—	—	—	—	—	—	—
(EPM, τ_3) − ((9), τ_1)	2.5	0.1	2.4	0.3	2.5	0.3	2.7	0.3	4.1	3
(EPM, τ_3) − ((10), τ_1)	2.9	0.1	2.8	0.2	2.5	0.3	2.6	0.3	7	0.3
(EPM, τ_3) − ((11), τ_1)	1.8	0.1	1.5	0.1	1.6	0.3	1.3	0.5	—	—
(EPM, τ_3) − (NS (7), τ_1)	1.5	0.2	1.5	0.1	1.5	0.2	1.9	0.4	6.5	0.7

ND_3

(Days)	τ_d	ε_d	$\tau_{25\%}$	$\varepsilon_{25\%}$	$\tau_{50\%}$	$\varepsilon_{50\%}$	$\tau_{71\%}$	$\varepsilon_{71\%}$	$\tau_{95\%}$	$\varepsilon_{95\%}$
(EPM, τ_3)	**4.9**	**0.1**	**8.2**	**0.2**	**11.6**	**0.2**	**14.4**	**0.2**	**21.6**	**0.4**
(EPM, τ_3) − ((9), τ_3)	2.6	0.1	2.3	0.1	2.4	0.1	2.1	0.1	2.7	0.2
(EPM, τ_3) − ((10), τ_3)	3.5	0.1	4	0.1	3.9	0	3.2	0.2	2.6	0.8
(EPM, τ_3) − ((11), τ_3)	2.1	0.1	1.6	0.1	1.6	0.2	1.6	0.2	2.1	0.3
(EPM, τ_3) − (NS (7), τ_3)	1	0.2	1.1	0.2	1.3	0.2	1	0.3	1	0.6
(EPM, τ_3) − ((9), τ_2)	2.6	0.1	2.4	0.1	2.3	0.1	2.2	0.2	2.4	0.9
(EPM, τ_3) − ((10), τ_2)	3.4	0	3.6	0.1	3.3	0.2	2.8	0.2	5	0.2
(EPM, τ_3) − ((11), τ_2)	2	0.1	1.5	0.2	1.6	0.2	1.6	0.2	1.2	1.4
(EPM, τ_3) − (NS (7), τ_2)	1.5	0.2	1.5	0.2	1.6	0.2	1.4	0.2	2.6	0.6
(EPM, τ_3) − ((9), τ_1)	2.7	0	2.3	0.1	2.3	0.1	2.2	0.2	3.4	0.6
(EPM, τ_3) − ((10), τ_1)	2.9	0	2.8	0.1	2.5	0.1	2.6	0.1	7	0.2
(EPM, τ_3) − ((11), τ_1)	1.8	0.1	1.4	0.2	1.6	0.2	1.4	0.3	—	—
(EPM, τ_3) − (NS (7), τ_1)	1.6	0.1	1.5	0.2	1.6	0.2	1.6	0.2	4.1	0.7

Table 2: Average values over $M_{1,\dots,5}$ of time limits (in days) for model values (EPM), for normal distributions $\mathrm{ND}_{4,5}$, and for prediction time length τ_3 (bold rows), difference between this model values and average values over $M_{1,\dots,5}$ received from (9)–(11) and NS (7) with parameters a, b, c, g, a_m, b_m, s_m, c_m, g_m, and l_m calculated from approximations of (4)–(7) for τ_1, τ_2, τ_3 and $M_{1,\dots,5}$ (columns τ_d, $\tau_{25\%}$, $\tau_{50\%}$, $\tau_{71\%}$, $\tau_{95\%}$). Standard deviation of $M_{1,\dots,5}$ (columns ε_d, $\varepsilon_{25\%}$, $\varepsilon_{50\%}$, $\varepsilon_{71\%}$, $\varepsilon_{95\%}$) and average value $\overline{\mathrm{ND}}$ and standard deviation (columns σ_d, $\sigma_{25\%}$, $\sigma_{50\%}$, $\sigma_{71\%}$, $\sigma_{95\%}$) of $\mathrm{ND}_{1,\dots,5}$.

						ND_4				
(Days)	τ_d	ε_d	$\tau_{25\%}$	$\varepsilon_{25\%}$	$\tau_{50\%}$	$\varepsilon_{50\%}$	$\tau_{71\%}$	$\varepsilon_{71\%}$	$\tau_{95\%}$	$\varepsilon_{95\%}$
(EPM, τ_3)	**4.9**	**0.1**	**6.8**	**0.1**	**10.2**	**0.2**	**12.9**	**0.1**	**19.7**	**0.3**
(EPM, τ_3) − ((9), τ_3)	2.4	0.1	2.3	0	2.3	0.1	2.3	0.1	2.4	0.3
(EPM, τ_3) − ((10), τ_3)	3.3	0.1	3.5	0.1	3.3	0.1	2.6	0.1	1.7	0.3
(EPM, τ_3) − ((11), τ_3)	1.8	0.1	1.5	0.1	1.4	0.1	1.5	0.2	1.8	0.3
(EPM, τ_3) − (NS (7), τ_3)	1.3	0.1	1.3	0.2	1.5	0.2	1.5	0.2	1.2	0.4
(EPM, τ_3) − ((9), τ_2)	2.5	0.1	2.4	0.1	2.3	0.1	2.3	0.1	2.1	0.6
(EPM, τ_3) − ((10), τ_2)	3.1	0	3.2	0.1	2.8	0.1	2.3	0.1	3.4	0.1
(EPM, τ_3) − ((11), τ_2)	1.6	0	1.3	0.1	1.4	0.2	1.5	0.2	0.4	0.8
(EPM, τ_3) − (NS (7), τ_2)	1.3	0.1	1.4	0.2	1.6	0.1	1.5	0.2	1.6	0.3
(EPM, τ_3) − ((9), τ_1)	2.5	0.1	2.2	0.1	2.2	0.1	2.2	0.2	3.6	0.6
(EPM, τ_3) − ((10), τ_1)	2.7	0.1	2.6	0.2	2.2	0.2	2.1	0.2	5.6	0.1
(EPM, τ_3) − ((11), τ_1)	1.6	0.1	1.5	0.2	1.6	0.2	1.6	0.2	—	—
(EPM, τ_3) − (NS (7), τ_1)	1.6	0.1	1.5	0.1	1.6	0.1	1.6	0.1	3.4	0.5
						ND_5				
(Days)	τ_d	ε_d	$\tau_{25\%}$	$\varepsilon_{25\%}$	$\tau_{50\%}$	$\varepsilon_{50\%}$	$\tau_{71\%}$	$\varepsilon_{71\%}$	$\tau_{95\%}$	$\varepsilon_{95\%}$
(EPM, τ_3)	**5**	**0.1**	**5.2**	**0.2**	**8.2**	**0.2**	**10.9**	**0.3**	**18.2**	**0.3**
(EPM, τ_3) − ((9), τ_3)	2.5	0	2.4	0	2.1	0.2	2.5	0.9	2.8	0.4
(EPM, τ_3) − ((10), τ_3)	3	0	2.9	0	2.5	0.1	1.6	0.2	1.4	0.3
(EPM, τ_3) − ((11), τ_3)	1.7	0.1	1.6	0.1	1.2	0.1	1.3	0.2	1.3	0.4
(EPM, τ_3) − (NS (7), τ_3)	1.6	0.1	1.6	0.1	1.6	0.1	1.6	0.2	2	0.2
(EPM, τ_3) − ((9), τ_2)	2.4	0.1	2.3	0.1	2	0.2	1.9	0.2	3.6	0.2
(EPM, τ_3) − ((10), τ_2)	2.6	0.1	2.6	0.1	1.9	0.1	1.4	0.2	4.6	0.2
(EPM, τ_3) − ((11), τ_2)	1.5	0.1	1.4	0.1	1.3	0.1	1.5	0.2	—	0.3
(EPM, τ_3) − (NS (7), τ_2)	1.6	0.1	1.6	0.1	1.6	0.1	1.6	0.2	1.9	0.2
(EPM, τ_3) − ((9), τ_1)	2.4	0.1	2.5	0.1	2.1	0.1	2	0.2	2.6	0.6
(EPM, τ_3) − ((10), τ_1)	2.8	0.1	2.8	0.1	2.2	0.2	1.5	0.2	2.6	0.2
(EPM, τ_3) − ((11), τ_1)	1.6	0	1.5	0	1.3	0.1	1.4	0.2	1.4	4.3
(EPM, τ_3) − (NS (7), τ_1)	1.6	0.1	1.6	0.1	1.5	0.1	1.6	0.2	1.9	1.1
						$\overline{\mathrm{ND}}$				
(Days)	τ_d	σ_d	$\tau_{25\%}$	$\sigma_{25\%}$	$\tau_{50\%}$	$\sigma_{50\%}$	$\tau_{71\%}$	$\sigma_{71\%}$	$\tau_{95\%}$	$\sigma_{95\%}$
(EPM, τ_3) − ((9), τ_3)	2.5	0.1	2.4	0.1	2.3	0.1	2.4	0.2	2.5	0.3
(EPM, τ_3) − ((10), τ_3)	3.4	0.2	4.3	1	4.2	1.2	3.5	1.3	3	1.4
(EPM, τ_3) − ((11), τ_3)	1.9	0.2	1.6	0.1	1.5	0.2	1.6	0.2	1.9	0.3
(EPM, τ_3) − (NS (7), τ_3)	1.3	0.2	1.3	0.2	1.5	0.1	1.3	0.2	1.4	0.4
(EPM, τ_3) − ((9), τ_2)	2.5	0.1	2.4	0	2.3	0.1	2.3	0.2	2.4	0.2
(EPM, τ_3) − ((10), τ_2)	3.2	0.2	3.7	0.6	3.3	0.6	3	0.9	5.2	1.9
(EPM, τ_3) − ((11), τ_2)	1.8	0.2	1.5	0.1	1.5	0.2	1.6	0.1	1	0.8
(EPM, τ_3) − (NS (7), τ_2)	1.5	0.1	1.5	0.1	1.5	0.0	1.5	0.1	2	0.4
(EPM, τ_3) − ((9), τ_1)	2.5	0.1	2.3	0.1	2.3	0.2	2.3	0.3	3.9	0.5
(EPM, τ_3) − ((10), τ_1)	2.8	0.1	2.7	0.1	2.3	0.3	2.4	0.7	6.5	1.5
(EPM, τ_3) − ((11), τ_1)	1.6	0.1	1.5	0	1.5	0.1	1.2	0.4	—	—
(EPM, τ_3) − (NS (7), τ_1)	1.5	0.1	1.4	0.1	1.5	0.1	1.7	0.2	4.5	1.6

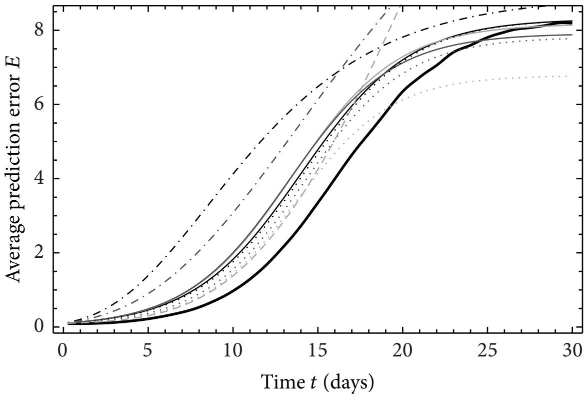


Figure 2: Time variations of the average prediction error E for ND_1, M_5. The thick line represents data from EPS, the thin lines from (9), the dotdashed lines from (10), the dotted lines from (11), and dashed lines from NS (7). The light grey lines represent the data extrapolate from time length τ_1, the grey lines represent the data extrapolated from time length τ_2 and black lines represent the data in time length τ_3.

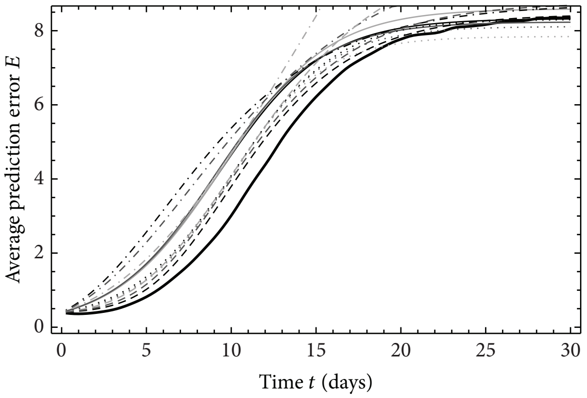


Figure 3: As Figure 2 for ND_3, M_5.

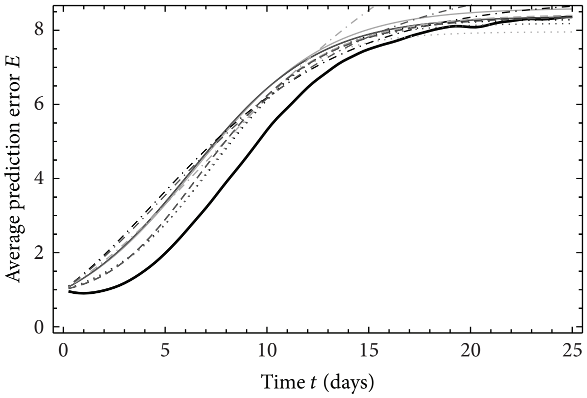


Figure 4: As Figure 2 for ND_5, M_5.

higher than $\tau_{71\%}$. In the case of values at τ_2, $\tau_{95\%}$, τ_1, $\tau_{71\%}$ and τ_1, $\tau_{95\%}$, the highest difference from almost constant values is for τ_1, $\tau_{95\%}$ and the best results are given by QH with solution (9).

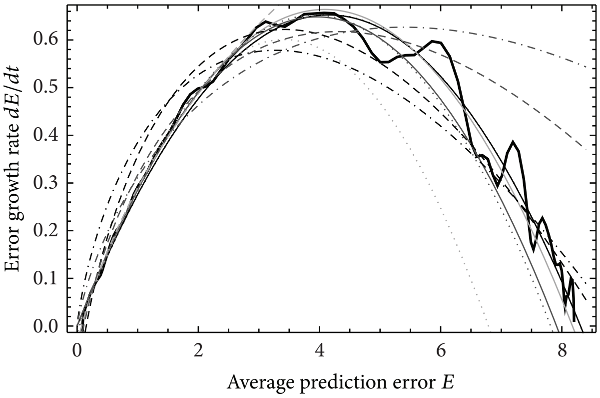


Figure 5: The error growth rate dE/dt versus E for ND_1, M_5. The thick line represents data from EPS, the thin lines from (9), the dotdashed lines from (10), the dotted lines from (11), and dashed lines from NS (7). The light grey lines represent the data extrapolated from time length τ_1, the grey lines represent the data extrapolated from time length τ_2, and black lines represent the data in time length τ_3.

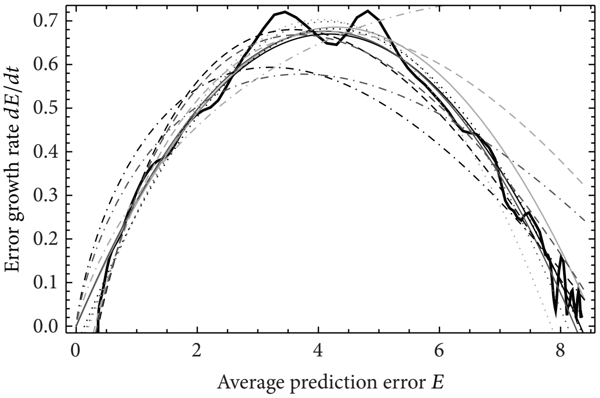


Figure 6: As Figure 5 for ND_3, M_5.

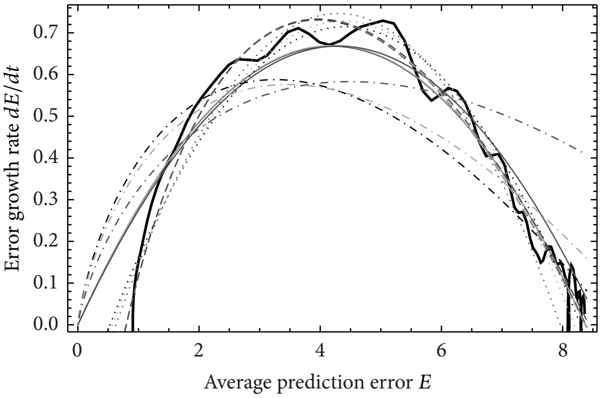


Figure 7: As Figure 5 for ND_5, M_5.

Table 3 focuses on the average values over $M_{1,\ldots,5}$ of $E_{\text{asym},\tau 1,\ldots,3}$ for $ND_{1,\ldots,5}$ and average $\overline{ND}$. This value is for example used to find out if the variability of the model is equal to the variability of the atmosphere [10]. The differences $\varepsilon_{1,\ldots,3}$ from model values E_{asym} (Table 3, Figures 5, 6, and 7) indicate

Table 3: Average values over $M_{1,\dots,5}$ of $E_{\mathrm{asym},\tau 1,\dots,3}$, calculated from approximations (4)–(7) for all initial conditions $\mathrm{ND}_{1,\dots,5}$ and for all prediction time lengths $\tau_{1,\dots,3}$ (bold columns), average value $\overline{\mathrm{ND}}$ over $\mathrm{ND}_{1,\dots,5}$ for all prediction time lengths $\tau_{1,\dots,3}$ and difference $\varepsilon_{1,\dots,3}$ of $E_{\mathrm{asym},\tau 1,\dots,3} - E_{\mathrm{asym}}$.

	$E_{\mathrm{asym},\tau 3}$	ε_3	$E_{\mathrm{asym},\tau 2}$	ε_2	$E_{\mathrm{asym},\tau 1}$	ε_1
(ND_1, (4))	**8.3 ± 0.1**	−0.1 ± 0.1	**8.4 ± 0.3**	0.0 ± 0.3	**8.1 ± 0.5**	−0.3 ± 0.5
(ND_1, (5))	**9.1 ± 0.4**	0.7 ± 0.4	**17.0 ± 1.4**	8.6 ± 1.4	—	—
(ND_1, (6))	**8.3 ± 0.1**	−0.1 ± 0.1	**8.4 ± 0.9**	0.0 ± 0.9	**8.0 ± 2.1**	−0.4 ± 2.1
(ND_1, (7))	**8.8 ± 0.2**	0.4 ± 0.2	**12.7 ± 2.4**	4.3 ± 2.4	**26.8 ± 7.7**	18.4 ± 7.7
(ND_2, (4))	**8.3 ± 0.0**	−0.1 ± 0.0	**8.3 ± 0.2**	−0.1 ± 0.2	**8.8 ± 0.7**	0.4 ± 0.7
(ND_2, (5))	**8.9 ± 0**	0.5 ± 0.0	**10.9 ± 1.0**	2.5 ± 1.0	**63.4 ± 21.0**	55.0 ± 21.0
(ND_2, (6))	**8.3 ± 0.3**	−0.1 ± 0.3	**8.2 ± 0.2**	−0.2 ± 0.2	**7.8 ± 0.7**	−0.6 ± 0.7
(ND_2, (7))	**8.6 ± 0.3**	0.2 ± 0.3	**10.2 ± 1.0**	1.8 ± 1.0	**15.9 ± 2.9**	7.5 ± 2.9
(ND_3, (4))	**8.4 ± 0.0**	0.0 ± 0.0	**8.3 ± 0.1**	−0.1 ± 0.1	**8.3 ± 0.2**	−0.1 ± 0.2
(ND_3, (5))	**8.8 ± 0.2**	0.4 ± 0.2	**10.2 ± 0.2**	1.8 ± 0.2	**18.2 ± 1.7**	9.8 ± 1.7
(ND_3, (6))	**8.3 ± 0.0**	−0.1 ± 0.0	**8.2 ± 0.2**	−0.2 ± 0.2	**7.7 ± 0.2**	−0.7 ± 0.2
(ND_3, (7))	**8.5 ± 0.1**	0.1 ± 0.1	**8.9 ± 0.2**	0.5 ± 0.2	**9.8 ± 0.3**	1.4 ± 0.3
(ND_4, (4))	**8.4 ± 0.1**	0.0 ± 0.1	**8.3 ± 0.1**	−0.1 ± 0.1	**8.7 ± 0.3**	0.3 ± 0.3
(ND_4, (5))	**8.8 ± 0.1**	0.4 ± 0.1	**9.6 ± 0.2**	1.2 ± 0.2	**15.3 ± 1.6**	6.9 ± 1.6
(ND_4, (6))	**8.3 ± 0.1**	−0.1 ± 0.1	**8.2 ± 0.1**	−0.2 ± 0.1	**8.1 ± 0.2**	−0.3 ± 0.2
(ND_4, (7))	**8.4 ± 0.1**	0.0 ± 0.1	**8.5 ± 0.1**	0.1 ± 0.1	**9.3 ± 0.4**	0.9 ± 0.4
(ND_5, (4))	**8.3 ± 0.0**	−0.1 ± 0.0	**8.3 ± 0.1**	−0.1 ± 0.1	**8.5 ± 0.1**	0.1 ± 0.1
(ND_5, (5))	**8.7 ± 0.1**	0.3 ± 0.1	**9.3 ± 0.1**	0.9 ± 0.1	**11.7 ± 0.5**	3.3 ± 0.5
(ND_5, (6))	**8.3 ± 0.0**	−0.1 ± 0.0	**8.2 ± 0.1**	−0.2 ± 0.1	**7.9 ± 0.1**	−0.5 ± 0.1
(ND_5, (7))	**8.4 ± 0.0**	0.0 ± 0.0	**8.3 ± 0.1**	−0.1 ± 0.1	**8.4 ± 0.1**	0.0 ± 0.1
($\overline{\mathrm{ND}}$, (4))	**8.3 ± 0.0**	−0.1 ± 0.0	**8.3 ± 0.1**	−0.1 ± 0.0	**8.5 ± 0.2**	0.1 ± 0.2
($\overline{\mathrm{ND}}$, (5))	**8.9 ± 0.1**	0.5 ± 0.1	**11.4 ± 2.2**	3.0 ± 2.2	**27.2 ± 18.1**	18.8 ± 18.1
($\overline{\mathrm{ND}}$, (6))	**8.3 ± 0.0**	−0.1 ± 0.0	**8.2 ± 0.1**	−0.2 ± 0.1	**7.9 ± 0.1**	−0.5 ± 0.1
($\overline{\mathrm{ND}}$, (7))	**8.5 ± 0.1**	0.1 ± 0.1	**9.7 ± 1.4**	1.3 ± 1.4	**14.0 ± 5.9**	5.6 ± 5.9

really poor approximation by (5) and (7) and usable approximations by (4) and (6), but with the already mentioned disadvantage of underestimations by (6). For (4), ε_3 lies between −0.1 and 0 (relatively against E_{asym} it means between −1.2% and 0%), ε_2 between −0.1 and 0 (between −1.2% and 0%), and ε_1 between −0.4 and 0.4. (−4.8% and 4.8%).

7. Conclusion

This paper studies errors of estimations of time limits and asymptotic value of initial errors growth in chaotic atmospheric model L05II introduced by Lorenz [12] with the parameters as close to NWPM as possible. Five types of initial conditions are represented by five normal distributions. Five settings of EPM showed the differences of order 0.1 and therefore the average value was chosen as model data. Quadratic hypothesis shows the best agreement with model's asymptotic value E_{asym} and good agreement with model time limits. Approximation can be even improved by subtraction of constant value and after that the quadratic hypothesis is closest to model data from all hypotheses. Purpose and size of this constant are explained. Logarithmic hypothesis has the lowest agreement with the model data for time limits and asymptotic value. Modified quadratic hypothesis is good in approximating the model asymptotic value but it is not the best. For time limits, it is the best choice for approximation as long as we do not use the subtraction of the constant. Disadvantage is that, for some cases, this hypothesis underestimates model data and therefore some time limits are not available. Modified logarithmic hypothesis does not give good agreement with model's asymptotic value but gives similar agreement with model's time limit as modified quadratic hypothesis. Disadvantage is that, for the first two initial conditions, it is not solvable and therefore is usable only for larger initial errors. Quadratic hypothesis after subtraction of the constant value overestimates the model data for 0.5 days on average. Higher value is shown only for the shortest prediction time length τ_1 and time limit $\tau_{95\%}$. The size is 1.9 days on average. Relative difference between model's asymptotic value and asymptotic value calculated from quadratic hypothesis is between 0 and 1.2% for prediction time τ_3 and τ_2 and between −4.8% and 4.8% for prediction time τ_1. So, only for the lastly mentioned prediction time, we should calculate with this difference.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors were supported by Project no. SVV-2013-267308.

References

[1] E. N. Lorenz, “The predictability of a flow which possesses many scales of motion,” *Tellus*, vol. 21, no. 3, pp. 289–307, 1969.

[2] H. Savijärvi, “Error growth in a large numerical forecast system,” *Monthly Weather Review*, vol. 123, no. 1, pp. 212–221, 1995.

[3] E. N. Lorenz, “Atmospheric predictability as revealed by naturally occurring analogs,” *Journal of the Atmospheric Sciences*, vol. 26, no. 4, pp. 636–646, 1969.

[4] E. N. Lorenz, “Atmospheric predictability experiments with a large numerical model.,” *Tellus*, vol. 34, no. 6, pp. 505–513, 1982.

[5] A. Dalcher and E. Kalney, “Error growth and predictability in operational ECMWF analyses,” *Tellus*, vol. 39A, pp. 474–491, 1987.

[6] A. J. Simmons, R. Mureau, and T. Petroliagis, “Error growth and estimates of predictability from the ECMWF forecasting system,” *Quarterly Journal of the Royal Meteorological Society*, vol. 121, no. 527, pp. 1739–1771, 1995.

[7] A. Trevisan, P. Malguzzi, and M. Fantini, “On Lorenz’s law for the growth of large and small errors in the atmosphere,” *Journal of the Atmospheric Sciences*, vol. 49, no. 8, pp. 713–719, 1992.

[8] L. S. R. Froude, L. Bengtsson, and K. I. Hodges, “Atmospheric predictability revisited,” *Tellus, Series A: Dynamic Meteorology and Oceanography*, vol. 65, no. 1, Article ID 19022, 2013.

[9] R. Buizza, “Horizontal resolution impact on short- and long-range forecast error,” *Quarterly Journal of the Royal Meteorological Society*, vol. 136, no. 649, pp. 1020–1035, 2010.

[10] L. K. Bengtsson, L. Magnusson, and E. Källén, “Independent estimations of the asymptotic variability in an ensemble forecast system,” *Monthly Weather Review*, vol. 136, no. 11, pp. 4105–4112, 2008.

[11] J. Ma, Y. Zhu, R. Wobus, and P. Wang, “An effective configuration of ensemble size and horizontal resolution for the NCEP GEFS,” *Advances in Atmospheric Sciences*, vol. 29, no. 4, pp. 782–794, 2012.

[12] E. N. Lorenz, “Designing chaotic models,” *Journal of the Atmospheric Sciences*, vol. 62, no. 5, pp. 1574–1587, 2005.

[13] T. Palmer and R. Hagedorn, *Predictability of Weather and Climate*, Cambridge University Press, Cambridge, UK, 2006.

[14] E. N. Lorenz, “Predictability: a problem partly solved,” in *Predictability of Weather and Climate*, T. Palmer and R. Hagegorn, Eds., pp. 1–18, Cambridge University Press, 1996.

[15] E. N. Lorenz, “Deterministic nonperiodic flow,” *Journal of the Atmospheric Sciences*, vol. 20, pp. 130–141, 1963.

[16] J. C. Sprott, *Chaos and Time-Series Analysis*, Oxford University Press, New York, NY, USA, 2003.

[17] A. Trevisan, “Impact of transient error growth on global average predictability measures,” *Journal of the Atmospheric Sciences*, vol. 50, no. 7, pp. 1016–1028, 1993.

[18] R. Ding and J. Li, “Comparisons of two ensemble mean methods in measuring the average error growth and the predictability,” *Acta Meteorologica Sinica*, vol. 25, no. 4, pp. 395–404, 2011.

[19] R. Benzi and G. F. Carnevale, “A possible measure of local predictability,” *Journal of the Atmospheric Sciences*, vol. 46, no. 23, pp. 3595–3598, 1989.

[20] H. Bednar, A. Raidl, and J. Miksovsky, “Initial errors growth in chaotic low-dimensional weather prediction model,” *Advances in Intelligent Systems and Computing*, vol. 210, pp. 333–342, 2013.

[21] H. Bednář, A. Raidl, and J. Mikšovský, “Initial error growth and predictability of chaotic low-dimensional atmospheric model,” *International Journal of Automation and Computing*, vol. 11, no. 3, pp. 256–264, 2014.

[22] A. Trevisan, F. Pancotti, and F. Molteni, “Ensemble prediction in a model with flow regimes,” *Quarterly Journal of the Royal Meteorological Society*, vol. 127, no. 572, pp. 343–358, 2001.

[23] L. Bengtsson and K. I. Hodges, “A note on atmospheric predictability,” *Tellus*, vol. 58, no. 1, pp. 154–157, 2006.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 569841, 6 pages
http://dx.doi.org/10.1155/2015/569841

Research Article

Advanced Approach of Multiagent Based Buoy Communication

Gediminas Gricius,[1] Darius Drungilas,[2] Arunas Andziulis,[2] Dale Dzemydiene,[3] Miroslav Voznak,[4] Mindaugas Kurmis,[1] and Sergej Jakovlev[2]

[1] *Institute of Mathematics and Informatics, Vilnius University, Akademijos Street 4, 08663 Vilnius, Lithuania*
[2] *Department of Informatics Engineering, Faculty of Marine Engineering, Klaipeda University, Bijunu Street 17-206, 91225 Klaipeda, Lithuania*
[3] *Institute of Digital Technologies, Faculty of Social Technologies, Mykolas Romeris University, Ateities Street 20, 08303 Vilnius, Lithuania*
[4] *Department of Telecommunications, Faculty of Electrical Engineering and Computer Science, VSB-Technical University of Ostrava, 17. Listopadu 15, 708 00 Ostrava, Czech Republic*

Correspondence should be addressed to Miroslav Voznak; miroslav.voznak@vsb.cz

Received 27 August 2014; Accepted 10 October 2014

Academic Editor: Ivan Zelinka

Usually, a hydrometeorological information system is faced with great data flows, but the data levels are often excessive, depending on the observed region of the water. The paper presents advanced buoy communication technologies based on multiagent interaction and data exchange between several monitoring system nodes. The proposed management of buoy communication is based on a clustering algorithm, which enables the performance of the hydrometeorological information system to be enhanced. The experiment is based on the design and analysis of the inexpensive but reliable Baltic Sea autonomous monitoring network (buoys), which would be able to continuously monitor and collect temperature, waviness, and other required data. The proposed approach of multiagent based buoy communication enables all the data from the costal-based station to be monitored with limited transition speed by setting different tasks for the agent-based buoy system according to the clustering information.

1. Introduction

There are a variety of tools to monitor and evaluate Baltic Sea hydrometeorological data, but most received information has low spatial coverage and low level of detail in time [1]. Sea wave height, water temperature, and underwater noise data, used for many practical applications, are usually obtained from three sources: buoy measurements, model calculations, and ship observations. Compared to other data acquisition methods, buoy measurements are the most reliable and readily data source available continuously for years [2]. Basically, the network of buoys is involved in mapping the temperature, wave height, and underwater noise at a buoy location using the data retrieved from other buoy locations [3]. However, many hydrometeorological data measurements using sea buoys can be lost due to malfunctions, maintenance, connection problems, or dubious data recorded by the buoy. In order to ensure greater reliability of data collection, it is necessary to develop a distributed information system, predicting complex situations and supporting decision-making processes. Information provided from such system is important for decision-makers and is needed to ensure the provision of information for decision-making institutions [4–6]. An important feature of the buoy network is the ability to monitor, collect, and evaluate wide spatial coverage and real-time hydrometeorological data of the Baltic Sea [5]. A hydrometeorological information system is faced with great data flows, but the data levels are often excessive, depending on the observed region of the water. Therefore, current traditional methods are no longer sufficient to ensure the rapid collection of data and valuable information extraction.

The purpose of this study is to show the possibilities of developing a hydrometeorological data collection system (HMDCS) involving advanced technologies such as multiagent based interaction and data collection between several monitoring system nodes (i.e., buoys) based on self-organizing maps (SOM). The experiment is based on the design of the inexpensive but reliable Baltic Sea autonomous monitoring network (buoys), which would be able to continuously monitor and collect temperature, waviness, and other required data. Moreover, it has the ability to monitor all the data from the coast-based station with limited transition speed by setting different tasks for the agent-based buoy system according to the SOM.

2. Sea Hydrometeorological Data Monitoring

Nowadays, there are numerous and varied designs for autonomous systems used for meteorological and oceanographic monitoring with different integration degrees. The buoy network system used in the Canary Islands is one of them [7]. It has a control center that manages the transmission communications and provides data in a useful form to diverse socioeconomically important sectors which make exhaustive use of the littoral, and data from the buoys are used to manage the coastal environment. The buoys monitor water temperature, salinity, dissolved oxygen, hydrocarbons, and other characteristics, which they can measure when equipped with other sensors such as a fluorometer and a turbidimeter, and each buoy is also able to communicate via GSM modem. Following a programmed sampling rate (every hour), the ECU sends to the central receiver unit a SMS message, which includes a sensor data set, GPS position, and battery level. However, deeper analysis of the data has shown that such a sampling rate is not sufficient, which means that the data transmit protocol must be reevaluated.

In order to provide greater hydrometeorological data monitoring reliability and faster data retrieval, a variety of sensory systems networks [8–10] have been proposed, such as communication technologies that enable communication between sensor nodes [10], systems for communication between maritime platforms like vessels, commercial ships, or buoys [9], and real-time monitoring of the underwater environment where an acoustic mesh network is located between the underwater sensor networks and the central monitoring system [8]. The proposed models can solve various problems but require more flexible solutions for complex data transfer problems. This problem can be solved by developing an active autonomous sensor multiagent based system, which is able to combine data processing methods according to the situation.

3. Hydrometeorological Data Sensory System

3.1. Temperature Data Collection. During the investigation stage of the HMDCS development, several types of temperature sensors were compared. The comparison possibilities are made by analyzing their parameters according to the technical specifications presented in datasheets.

After a comparative analysis of the temperature sensors, we selected the DS18B20 digital sensor. This digital temperature sensor can measure temperatures within the range from −55°C to +125°C at 12-bit precision, with accuracy −0.50°C [11]. However, after additional calculations, it is possible to reduce the temperature measurement error down to 0.10°C. The most attractive feature is the fact that these sensors have already been calibrated at the factory and their accuracy error is ±0.5°C in the range from −10°C to +85°C and ±2°C error over the operating range (55°C to +125°C). Sensor supply voltage is in the range of +3 to +5.5 V. In standby mode, current consumption is close to zero (less than 1 μA), while temperature conversion power use is about 1 mA. The measurement process lasts no more than 0.75 sec. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. In addition, the DS18B20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply. Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-Wire bus. Thus, it is simple to use one microprocessor to control many DS18B20s distributed over an area of few square meters (in our case they are used to measure temperature in different depth of the sea). This part has already become the corner stone of many data logging and temperature control projects.

3.2. Waviness Measurements. At present, sea and ocean waviness measurements use a variety of methods, depending on the geographic region, measuring accuracy, and general tasks [12]. The main and most commonly used are as follows:

(i) ultrasound-based sensors:
   (1) pros: suitable for measuring waves with a height of over 5 meters,
   (2) cons: significant measurement errors,

(ii) rheostat-type structures:
   (1) pros: allow you to get fairly accurate data,
   (2) cons: because of their design features they have a short lifetime,

(iii) satellite image analysis:
   (1) cons: due to the inherent large errors, this can be used only for ocean waviness measurements,

(iv) GPS system:
   (1) cons: not suitable for measuring waves with a height of 0.5–2.0 meters,

(v) accelerometer and gyroscope design:
   (1) pros: small measurement errors, easy implementation.

For our experiment the couple of accelerometer and gyroscope was used. Based on the experience of other scientists [1], accelerometer data were processed by removing the component of gravity, according to the following:

$$\begin{bmatrix} X_E \\ Y_E \\ Z_E \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} \begin{bmatrix} X_S \\ Y_S \\ Z_S \end{bmatrix}. \quad (1)$$

Here, X_S, Y_S, Z_S represent the accelerations measured in the sensor frame, X_E, Y_E, Z_E are the accelerations rotated into the earth coordinate frame, and the direction cosines for the above transformation are in terms of the Euler attitude angles.

The coefficients a, b, and c are calculated using the following formulas:

$$\begin{aligned} a_1 &= \cos\theta\cos\psi, \\ b_1 &= \sin\varphi\sin\theta\cos\psi - \cos\varphi\sin\psi, \\ c_1 &= \sin\varphi\sin\theta\cos\psi + \cos\varphi\sin\psi, \\ a_2 &= \cos\theta\sin\psi, \\ b_2 &= \sin\varphi\sin\theta\cos\psi + \cos\varphi\sin\psi, \\ c_2 &= \sin\varphi\sin\theta\cos\psi - \cos\varphi\sin\psi, \\ a_3 &= -\sin\theta, \\ b_3 &= \sin\varphi\cos\theta, \\ c_3 &= \cos\varphi\cos\theta. \end{aligned} \tag{2}$$

Here, θ, ψ and φ are data from the gyroscope. After the accelerations have been rotated into the earth frame, the earth-referenced accelerations of the buoy are given by

$$\begin{aligned} A_X &= -gX_E, \\ A_Y &= -gY_E, \\ A_Z &= g\left(1 - Z_E\right), \end{aligned} \tag{3}$$

where A_X, A_Y, and A_Z are accelerations with eliminated gravity force along the earth-oriented X-, Y-, and Z-axes.

3.3. Data Transmissions. After comparing the most popular data transmission protocols such as Bluetooth, UWB, ZigBee, Wi-Fi, and others [5], it was decided that ZigBee is the most suitable transmission protocol for such a task (low-cost, low power, mesh network support). So, this mesh-type network protocol was used for developing the HMDCS buoy network. ZigBee is an open standard for short-range wireless networks based on the Physical Layer and the Media Access Control from IEEE 802.15.4, focusing on minimizing the overall power consumption and at the same time maximizing network reliability [13].

The ZigBee protocol offers three kinds of devices to form a PAN (personal area network):

(1) end-devices, which periodically collect data and transmit it,

(2) routers: they collect data from end-devices and forward it to the destination (like another router or to the final coordinator),

(3) coordinator: one of the routers in a PAN is usually configured as a coordinator. The main function of the coordinator is the parameterization and management of the PAN, and the collection of network data.

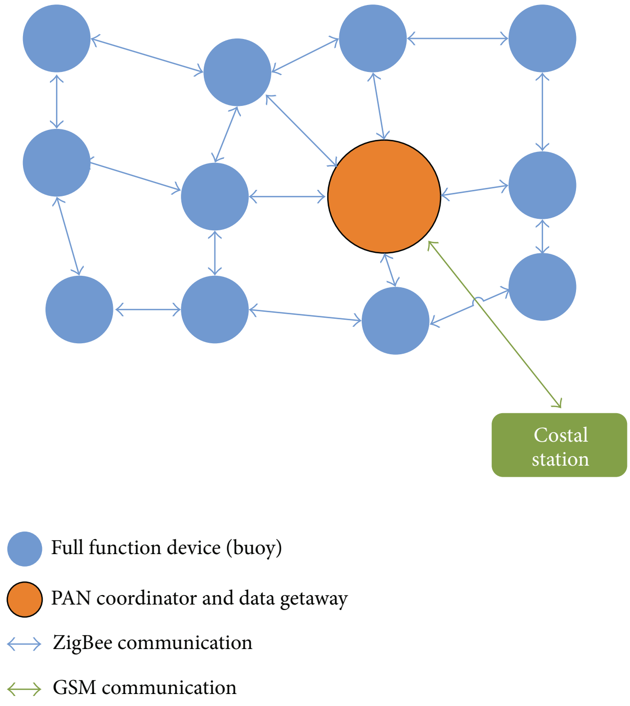


Figure 1: Mesh network.

In our case, we used so-called "full function devices" which collect data and work as a router and coordinator, which manages the PAN network and sends collected data via GSM to a coastal station (Figure 1). The following ZigBee network configuration was used for transmitting data to the coastal station.

3.4. Buoy Power Management. Power requirements for the electronic buoy system are 5 V DC at 76 mA in active cycle (active sensors and microcontroller are calculating data; transmitter is sending information) and about 27 mA at passive cycle (microcontroller is in sleep mode, and only the receiver is powered up for wake up using external interrupt). Active cycle lengths is about 5 seconds in every 10 minutes, so the duty cycle of the buoy is 0,0083. The buoy power supply consists of a battery bank of 18 AA type Ni-MH battery cells, arranged in 3 parallel groups of 6 cells connected in series. The capacity of each battery used in our buoy system is 2200 mAh so the capacity of each group of 3 batteries combined in parallel groups is 6600 mAh at 7.2 V. Expected lifetime of such configuration system at 25°C temperature would be about 10 days. Batteries placed in the bottom of the buoy also serve as ballast.

4. Agent Action Distribution Using SOM

The proposed multiagent sensory system is based on the goal of task distribution for agents according to action similarities. This can be implemented applying self-organising map neural networks (SOM). A SOM defines a two-dimensional nonlinear manifold as a regular array of discrete points. In

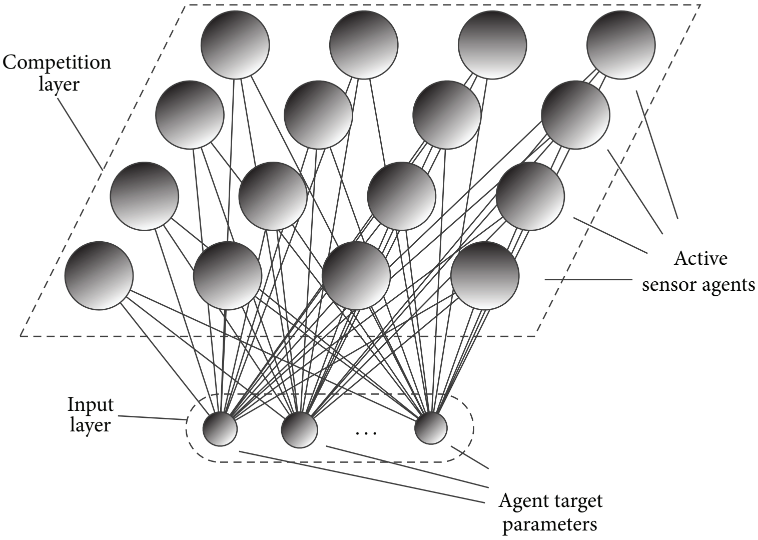


Figure 2: SOM for autonomous agents-based sensory system.

this way, the application of unsupervised learning allows a multidimensional vector represented in two-dimensional output space. The SOM output layer neurons retain a topological structure according to internal data structure. A typical SOM neural network architecture is shown in Figure 2. The input nodes represent the parameter vector, which according to the similarity is projected in the two-dimensional output space—the competitive layer. The input layer represents the parameters of the agents' target selection, and the competitive layer represents the autonomous agents-based sensory system.

In order to get the topological structure of the SOM, a training process should be applied. Each unit in the competition layer array is associated with a parametric reference vector weight of dimension n. Each input vector is compared with the reference vector weight w_j of each unit. The best match with the smallest Euclidean distance is defined as the response, and the input is mapped onto this location. Initially, all reference vector weights are assigned to small random values and are updated as [14]

$$\Delta w_j = \alpha_n(t)\, h_j(g,t)\left(x_i - w_j(t)\right), \tag{4}$$

where $\alpha(t)$ is the learning rate at time t and $h_j(g,t)$ is the neighborhood function from winner unit neuron g to neuron j at time t. In general, the neighborhood function decreases monotonically as a function of the distance from neuron g to neuron n. This decreasing property is a necessary condition for convergence [14].

SOM competition layer nodes correspond to individual agents as active sensory nodes, which are able to process data at a different level (filtering, sampling, transfer, and other). The capacity of the wireless network, the data capture excess in the central database, and so on depend on these characteristics. Assuming that each agent as an active buoy sensor node performs different actions, the central unit can distribute tasks for the agents in accordance with their capabilities and the required information. In this case, we use three parameters as the inputs for the SOM, which determine the actions performed by the agents—the significance of the measurement data, hydrometeorological characteristics of interest, and the number of the sampling rate. These parameters as appropriate capability are predefined for every active sensor agent in the SOM's competition layer. Under these settings, the actions are distributed for the agents according to the common goal and the capability of each agent. For example, if we need raw data, the task will be forwarded to agents that have a high data transfer bandwidth but do not have filtering capabilities.

5. Multiagent System Model for Hydrometeorological Sensory System

For proper buoy operation, a multiagent type system was designed. The agent software was developed using a multi-agent framework and works internally in the buoy. Figure 3 shows one buoy agent example. The buoy agent has one main goal: measure data and different tasks given by posting a newMeasurmentGoal message from coordinator (SOM network). The buoy agent can read new data using capability Measure (Figure 4). Once the sensor has read the data, the messages onReadWTemper (for water temperature), oReadOTemper (for weather temperature), and onReadWaveHg (for wave height) occur.

The buoy agent stores data in the local DB and if necessary, it is able to post it to other agents via the ZigBee network using the plan SendData.

6. Results and Discussion

For sea wave height, five different measurement methods were analyzed: using an ultrasonic sensor, a rheostat-type sensor, accelerometer and gyroscope sensors, satellite photos, and GPS data [12]. For data transmission from buoys to the main station different transmission methods and protocols

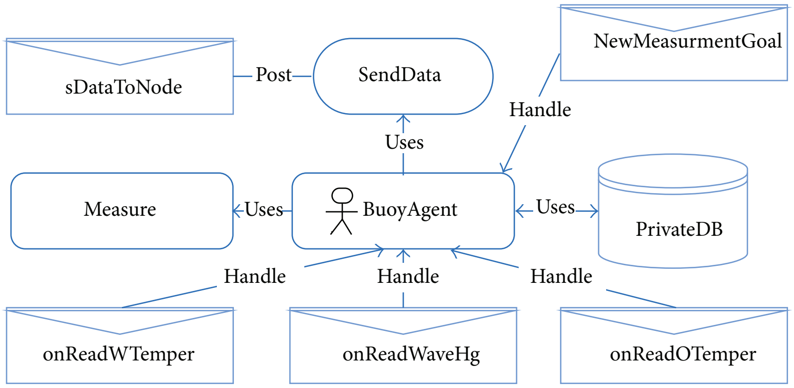


Figure 3: Buoy agent schematic.

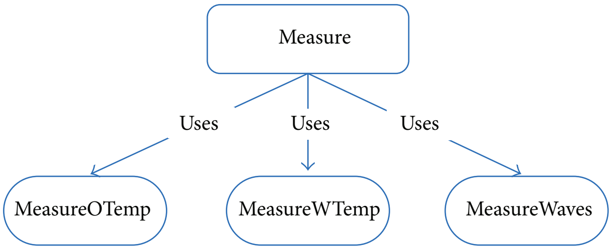


Figure 4: Measure data capability.

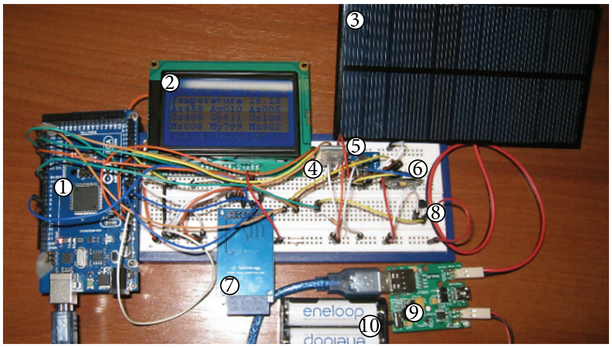


Figure 5: Buoy electronic system prototype.

were analyzed, but most focused on mesh-type wireless networks and agent-based communication methods [11, 15].

For testing purposes, an experimental buoy sensory system was developed. The core component of the prototype is the Arduino Mega platform with an ATmega2560 microcontroller which operates at 16 MHz clock frequency (Figure 5-(1)). The experimental buoy system is powered by solar power supply, which also recharges the Ni-MH batteries, which allow the buoy sensory system to operate at night (Figure 5-(3), (9), and (10)). Buoy status is shown on an LCD display (Figure 5-(2)). XBee Pro modules (Figure 5-(4)) implement communication via the ZigBee protocol and have 10 mW transmission power and according to the specifications, an expected distance of about 1–1.5 km outdoors. Temperature measurements (underwater and weather) are implemented using a DS18B20 sensors array connected to a 1-wire network (Figure 5-(5)). Data logging is to MMC (Figure 5-(7)). The wave height is measured using an MPU6050 (Figure 5-(6)) accelerometer/gyroscope and calculated by the provided method.

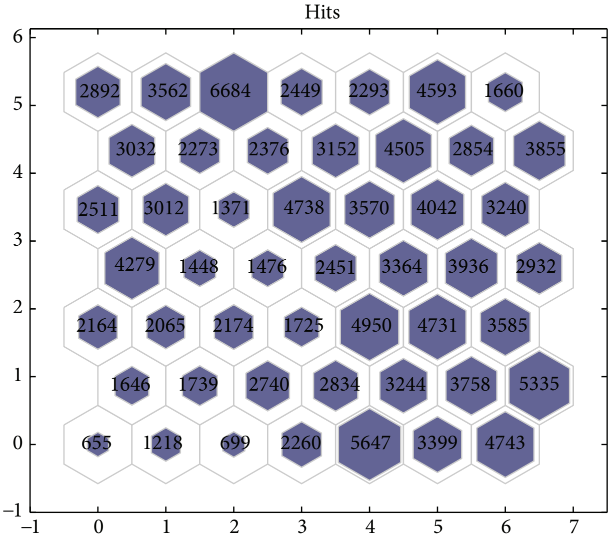


Figure 6: The distributed sensory node priorities using the SOM neural network.

The constructed prototype was placed in a hermetic housing and tested offshore in the Baltic Sea. The construction design and electronics solutions look very promising: one buoy's electronics cost only about 100 EUR, and the experimentally tested point to point network with 10 mW Xbee modules in open sea has a transmission distance of at least 900 m (it is enough to build a mesh-type buoy network).

According to the Baltic Sea Monitoring Data Base [16] a hydrometeorological mesh-type data collection network was established, which enables the performance evaluation of each sensor node. This evaluation allows the coastal central station to distribute the agent performance according to the amount of required data. Figure 6 shows the distribution of sensory node priorities using the SOM neural network.

Each sensor node priority defines the importance of the measurements and the amount of data transmitted; that is, a sensor node with higher priority requires the agent to transmit larger amounts of data, which should allow a more accurate assessment of the sea region of interest.

7. Conclusions

This paper presents the possibilities of developing a hydrometeorological data collection system (HMDCS) involving advanced technologies such as multiagent based interaction and data collection between several monitoring system nodes (i.e., buoys) based on self-organizing maps (SOM). The proposed solutions of HMDCS look very promising because of the inexpensive but reliable Baltic Sea autonomous monitoring network (buoys), which is able to continuously monitor and collect temperature, waviness, and other required data. The multiagent type system was designed to monitor data from a coastal-based station with limited transition speed by setting different tasks for the agent-based buoy system according to the SOM.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the Latvia-Lithuania cross-border cooperation programme within the project “JRTC Extension in Area of Development of Distributed Real-Time Signal Processing and Control Systems,” project code LLIV-215, and the project “Lithuanian Maritime Sector’s Technologies and Environment Research Development,” project code VP1-3.1-ŠMM-08-K-01-019, and was partially supported by SGS Grant no. SP2014/72, VSB-Technical University of Ostrava, Czech Republic, by the European Regional Development Fund in the IT4 Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070), and by the development of human resources in research and development of the latest soft computing methods and their application in practice project (CZ.1.07/2.3.00/20.0072) funded by the Operational Programme Education for Competitiveness, cofinanced by ESF and the state budget of the Czech Republic.

References

[1] L. Bender, N. L. Guinasso, J. N. Walpert, and S. D. Howden, “A comparison of methods for determining significant wave heights,” *Journal of Atmospheric and Oceanic Technology*, vol. 27, no. 6, pp. 1012–1028, 2010.

[2] V. Panchang, L. Zhao, and Z. Demirbilek, “Estimation of extreme wave heights using GEOSAT measurements,” *Ocean Engineering*, vol. 26, no. 3, pp. 205–225, 1998.

[3] S. N. Londhe, “Development of wave buoy network using soft computing techniques,” in *OCEANS 2008—MTS/IEEE Kobe Techno-Ocean*, pp. 1–8, 2008.

[4] A. A. Bielskis, A. Andziulis, O. Ramasauskas, E. Guseinoviene, D. Dzemydiene, and G. Gricius, “Multi-agent based E-social care support system for inhabitancies of a smart eco-social apartment,” *Electronics and Electrical Engineering*, no. 107, pp. 11–14, 2011.

[5] D. Bykovas, D. Drungilas, A. Andziulis, and J. Venskus, “Jūrų tyrimų ir monitoringo sensorinės automatizuotos informacinės sistemos, skirtos išankstiniam ekologinių problem identifikavimui, projektavimo koncepcija. Jūros ir krantų tyrimai,” in *7-oji Nacionalinė Jūros Mokslų ir Technologijų Konferencija*, pp. 39–42, Klaipėdos Universiteto Baltijos Pajūrio Aplinkos Tyrimų ir Planavimo Institutas, Klaipėda, Lithuania, 2013.

[6] D. Dzemydienė, “Sprendimų paramos sistemos galimybės vertinti vandens taršos procesus. Jūros ir krantų tyrimai,” in *7-oji Nacionalinė Jūros Mokslų ir Technologijų Konferencij*, pp. 69–72, Klaipėdos Universiteto Baltijos Pajūrio Aplinkos Tyrimų ir Planavimo Institutas, Klaipėda, Lithuania, April 2013.

[7] C. Barrera, M. J. Rueda, J. C. Elgue, and O. Llinas, “Red ACOMAR: coastal moored buoy network for real-time surveillance, control and observation in Canary islands,” in *OCEANS 2006*, pp. 1–5, 2006.

[8] J. H. Laarhuis, “MaritimeManet: mobile ad-hoc networking at sea,” in *Proceedings of the International Waterside Security Conference (WSS ’10)*, pp. 1–6, November 2010.

[9] T. Li, “Multi-sink opportunistic routing protocol for underwater mesh network,” in *Proceedings of the International Conference on Communications, Circuits and Systems (ICCCAS ’08)*, pp. 405–409, May 2008.

[10] M. A. Mirza, M. Z. Shakir, and M. Slim-Alouini, “A GPS-free passive acoustic localization scheme for underwater wireless sensor networks,” in *Proceedings of the 8th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS ’11)*, pp. 879–884, Valencia, Spain, October 2011.

[11] Maxim Integrated: DS18B20, Datasheat, 2008.

[12] C. Collins, *In Situ Wave Measurements: Sensor Comparison and Data Analysis*, vol. 3 of *Open Access Theses*, University of Miami, 2012.

[13] A. Sieber, M. Cocco, J. Markert, M. F. Wagner, R. Bedini, and P. Dario, “Zigbee based buoy network platform for environmental monitoring and preservation: temperature profiling for better understanding of mucilage massive blooming,” in *Proceedings of the 6th Workshop on Intelligent Solutions in Embedded Systems (WISES ’08)*, pp. 1–14, July 2008.

[14] D. Drungilas, A. A. Bielskis, and V. Denisov, “An intelligent control system based on non-invasive man machine interaction,” in *Innovations in Computing Sciences and Software Engineering*, pp. 63–68, 2010.

[15] “Texas Instruments: LM35 Precision Centigrade Temperature Sensors,” Datasheat, 2013.

[16] International Council for the Exploration of the Sea, “Baltic Sea monitoring data,” http://ocean.ices.dk/Helcom/Helcom.aspx.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 389273, 10 pages
http://dx.doi.org/10.1155/2015/389273

Research Article

Nonlinear versus Ordinary Adaptive Control of Continuous Stirred-Tank Reactor

Jiri Vojtesek and Petr Dostal

Faculty of Applied Informatics, Tomas Bata University in Zlin, Nam. T.G. Masaryka 5555, 760 01 Zlin, Czech Republic

Correspondence should be addressed to Jiri Vojtesek; vojtesek@fai.utb.cz

Received 12 August 2014; Accepted 7 October 2014

Academic Editor: Ivan Zelinka

Unfortunately, the major group of the systems in industry has nonlinear behavior and control of such processes with conventional control approaches with fixed parameters causes problems and suboptimal or unstable control results. An adaptive control is one way to how we can cope with nonlinearity of the system. This contribution compares classic adaptive control and its modification with Wiener system. This configuration divides nonlinear controller into the dynamic linear part and the static nonlinear part. The dynamic linear part is constructed with the use of polynomial synthesis together with the pole-placement method and the spectral factorization. The static nonlinear part uses static analysis of the controlled plant for introducing the mathematical nonlinear description of the relation between the controlled output and the change of the control input. Proposed controller is tested by the simulations on the mathematical model of the continuous stirred-tank reactor with cooling in the jacket as a typical nonlinear system.

1. Introduction

The control of the chemical processes in the industry is always challenging because of the nonlinearity of the major group of systems. The continuous stirred-tank reactor (CSTR) is one of the most common used types of chemical reactors because of easy controllability [1].

The adaptive control [2] is a control technique with good theoretical background and also practical implementations. It uses idea of the living organisms that adopts their behavior to the actual environmental conditions. There are also various adaptation techniques and variations described, for example, in [3].

The control method used here is based on the combination of the adaptive control and nonlinear control. Theory of nonlinear control (NC) can be found, for example, in [4, 5]. The nonlinear adaptive controller is divided via Wiener's model [6] into two parts: the dynamic linear part (DLP) and the static nonlinear part (SNP). The DLP uses polynomial synthesis [7] with pole-placement method and spectral factorization and all these methods satisfy basic control requirements such as disturbance attenuation, stability, and reference signal tracking. The second, nonlinear, part uses measurements of the steady-state behavior of the system for mathematical description of the dependence between the controlled output variable and the control input variable.

The controlled system, CSTR, with originally nonlinear behavior could be mathematically described for the control purposes by the external linear model (ELM) [8], parameters of which could vary because of the nonlinearity of the system. This problem could be overcome with the use of recursive identification which recomputes parameters of the ELM according to the actual state and the behavior of the system. There were used delta- (δ-) model [8] in this work as a special type of discrete-time models parameters of which approaches to the continuous ones for the small sampling period as it is proofed, for example, in [9].

The results are also compared with classical adaptive control which uses only ELM as a linear representation of the originally nonlinear controller [10, 11] to show the improvement of this nonlinear adaptive control strategy.

The proposed control strategies were verified by simulations on the mathematical model of CSTR with cooling in the jacket [12]. This mathematical model was studied also in [10]

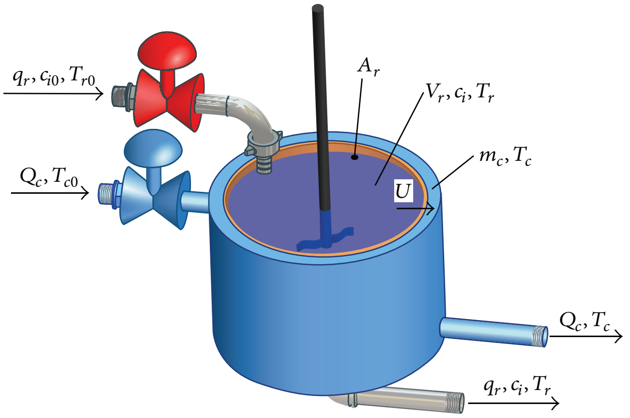


FIGURE 1: Continuous stirred-tank reactor with cooling in the jacket.

and classic adaptive controller was applied in [11]. All simulations were done in the mathematical software Matlab, version 7.0.1.

2. Controlled Plant

The system under the consideration is a continuous stirred-tank reactor (CSTR) with the so-called *Van der Vusse reaction* $A \rightarrow B \rightarrow C, 2A \rightarrow D$ inside and cooling jacket—see the scheme of the CSTR in Figure 1.

If we introduce common simplifications like the perfect mixture of the reactant, all densities, transfer coefficients, heat capacities, and the volume of the reactant are constant throughout the reaction, and the mathematical model developed with the use of material and heat balances inside has form of the set of ordinary differential equations (ODEs) [12]

$$\begin{aligned}
\frac{dc_A}{dt} &= \frac{q_r}{V_r}\left(c_{A0} - c_A\right) - k_1 c_A - k_3 c_A^2, \\
\frac{dc_B}{dt} &= -\frac{q_r}{V_r} c_B + k_1 c_A - k_2 c_B, \\
\frac{dT_r}{dt} &= \frac{q_r}{V_r}\left(T_{r0} - T_r\right) - \frac{h_r}{\rho_r c_{pr}} + \frac{A_r U}{V_r \rho_r c_{pr}}\left(T_c - T_r\right), \\
\frac{dT_c}{dt} &= \frac{1}{m_c c_{pc}}\left(Q_c + A_r U\left(T_r - T_c\right)\right),
\end{aligned} \tag{1}$$

where t in (1) is the time, c are concentrations, T represents temperatures, c_p is used for specific heat capacities, q_r means the volumetric flow rate of the reactant, Q_c is the heat removal of the cooling liquid, V_r is volume of the reactant, ρ stands for densities, A_r is the heat exchange surface, and U is the heat transfer coefficient. Indexes $(\cdot)_A$ and $(\cdot)_B$ belong to compounds A and B, respectively, $(\cdot)_r$ denotes the reactant mixture, $(\cdot)_c$ denotes cooling liquid, and $(\cdot)_0$ are feed (inlet) values.

TABLE 1: Fixed parameters of the CSTR.

Name of the parameter	Symbol and value of the parameter
Volume of the reactant	$V_r = 0.01\,\text{m}^{-3}$
Density of the reactant	$\rho_r = 934.2\,\text{kg}\cdot\text{m}^{-3}$
Heat capacity of the reactant	$c_{pr} = 3.01\,\text{kJ}\cdot\text{kg}^{-1}\cdot\text{K}^{-1}$
Weight of the coolant	$m_c = 5\,\text{kg}$
Heat capacity of the coolant	$c_{pc} = 2.0\,\text{kJ}\cdot\text{kg}^{-1}\cdot\text{K}^{-1}$
Surface of the cooling jacket	$A_r = 0.215\,\text{m}^2$
Heat transfer coefficient	$U = 67.2\,\text{kJ}\cdot\text{min}^{-1}\cdot\text{m}^{-2}\cdot\text{K}^{-1}$
Preexponential factor for reaction 1	$k_{01} = 2.145 \cdot 10^{10}\,\text{min}^{-1}$
Preexponential factor for reaction 2	$k_{02} = 2.145 \cdot 10^{10}\,\text{min}^{-1}$
Preexponential factor for reaction 3	$k_{03} = 1.5072 \cdot 10^{8}\,\text{min}^{-1}\cdot\text{kmol}^{-1}$
Activation energy of reaction 1 to R	$E_1/R = 9758.3\,\text{K}$
Activation energy of reaction 2 to R	$E_2/R = 9758.3\,\text{K}$
Activation energy of reaction 3 to R	$E_3/R = 8560\,\text{K}$
Enthalpy of reaction 1	$h_1 = -4200\,\text{kJ}\cdot\text{kmol}^{-1}$
Enthalpy of reaction 2	$h_2 = 11000\,\text{kJ}\cdot\text{kmol}^{-1}$
Enthalpy of reaction 3	$h_3 = 41850\,\text{kJ}\cdot\text{kmol}^{-1}$
Input concentration of compound A	$c_{A0} = 5.1\,\text{kmol}\cdot\text{m}^{-3}$
Input temperature of the reactant	$T_{r0} = 387.05\,\text{K}$

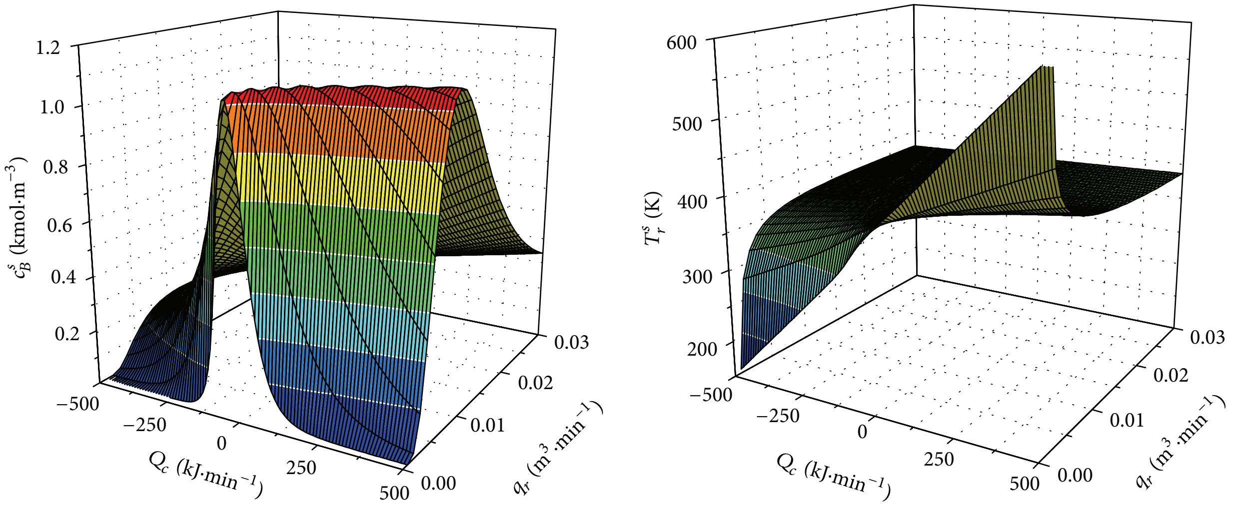
The variable h_r and k_{1-3} in (1) denote the reaction heat and reaction rates which are computed from

$$\begin{gathered}
h_r = h_1 \cdot k_1 \cdot c_A + h_2 \cdot k_2 \cdot c_B + h_3 \cdot k_3 \cdot c_A^2, \\
k_j\left(T_r\right) = k_{0j} \cdot \exp\left(\frac{-E_j}{RT_r}\right), \quad \text{for } j = 1, 2, 3,
\end{gathered} \tag{2}$$

where h_i stands for reaction enthalpies. Reaction rates k_{1-3} in the second equation are nonlinear functions of the reactants temperature computed via *Arrhenius law* with k_{0j} as rate constants, E_j are activation energies, and R means gas constant.

Equations (1) together with (2) construct the *mathematical model of the plant* used later for simulation studies. Due to simplifications introduced above we can say that this type of reactor is *a nonlinear lumped-parameters system*. We have four state variables c_A, c_B, T_r, and T_c and four input variables: the volumetric flow rate of the reactant, q_r, the heat removal of the coolant, Q_c, the input concentration, c_{A0}, and input temperature of the reactant, T_{r0}. The fixed values of the reactor are shown in Table 1 [12].

It is good to know behavior of the system before the design of the controller. This behavior is usually obtained from the steady-state and dynamic analyses of the system which will be described in the next subchapters.



(a)

(b)

FIGURE 2: Steady-state characteristics of the product's concentration c_B^s (a) and reactant's temperature T_r^s (b).

2.1. Steady-State Analysis. This analysis observes the behavior of the system in the steady-state, that is, in time $t \rightarrow \infty$. Mathematically speaking, derivatives with respect to time in the set of ODEs (1) are equal to zero; that is,

$$\frac{d(\cdot)}{dt} = 0 \tag{3}$$

which means that the set of ODEs (1) is transformed to the set of nonlinear algebraic equations that can be solved, for example, with the simple iterative method. This method is easily programmable in common mathematical software.

Results of steady-state analyses for different volumetric flow rate of the reactant $q_r = \langle 0; 0.03\rangle$ m$^3 \cdot$ min^{-1} and heat removal of cooling $Q_c = \langle -500; 500\rangle$ kJ $\cdot$ min^{-1} are shown in Figures 2(a) and 2(b).

Both graphs show highly nonlinear steady-state behavior of this system.

2.2. Dynamic Analysis. The second, dynamic, analysis shows the response of the system to the step change of the input quantity. Although there could be theoretically four input quantities, the volumetric flow rate of the reactant, q_r, and heat removal of the cooling, Q_c, were chosen as an input variables mainly from the practical point of view. Figures 3 and 4 show dynamic responses for various step changes of the input quantities in the working point $q_r^s = 2.365 \cdot 10^{-3}$ m$^3 \cdot$ min^{-1} and $Q_c^s = -18.56$ kJ $\cdot$ min^{-1}. Inputs u_1 and u_2 represent step changes of the q_r and Q_c, respectively, and outputs y_1 and y_2 show difference of the output products concentration, c_B, and reactants temperature, T_r, from their initial, that is, steady-state, value:

$$\begin{aligned}
u_1(t) &= Q_c(t) - Q_c^s \quad \left[\mathrm{kJ} \cdot \mathrm{min}^{-1}\right], \\
u_2(t) &= q_r(t) - q_r^s \quad \left[\mathrm{m}^3 \cdot \mathrm{min}^{-1}\right], \\
y_1(t) &= c_B(t) - c_B^s \quad \left[\mathrm{kmol} \cdot \mathrm{m}^{-3}\right], \\
y_2(t) &= T_r(t) - T_r^s \quad [\mathrm{K}],
\end{aligned} \tag{4}$$

where initial values of c_B and T_r are $c_B^s = 1.0903$ kmol$\cdot$m^{-3} and $T_r^s = 387.34$ K.

3. Nonlinear Adaptive Control Strategy

The control strategy here is based on the factorization of controller into the static nonlinear part (SNP) and the dynamic linear part (DLP); see Figure 5. This control scheme configuration is called a *Wiener system*.

As written in the previous part, there are theoretically four input and four output variables. In this case, the change of the output concentration, c_B, from its steady-state value, c_B^s, was controlled with the change of the volumetric flow rate of the reactant, q_r, from the working point, q_r^s; that is,

$$\begin{aligned}
u(t) &= \Delta q_r = q_r(t) - q_r^s \left[\mathrm{m}^3 \cdot \mathrm{min}^{-1}\right], \\
y(t) &= \Delta c_B = c_B(t) - c_B^s \left[\mathrm{kmol} \cdot \mathrm{m}^{-3}\right].
\end{aligned} \tag{5}$$

The dynamic part DLP in Figure 5 represents linear dynamic relation between the tracking error $e(t)$ and the input to the nonlinear static part $u_0(t) = \Delta c_{Bw}(t)$ which is difference between the concentration of the product, $c_B(t)$, and its desired value. The second static nonlinear part then describes the relation between $u_0(t)$ and corresponding change of the input volumetric flow rate of the reactant $\Delta q_r(t)$.

The schematic representation of the control system can be found in Figure 6.

3.1. Static Nonlinear Part. The nonlinear part uses properties of the system in the steady-state described above.

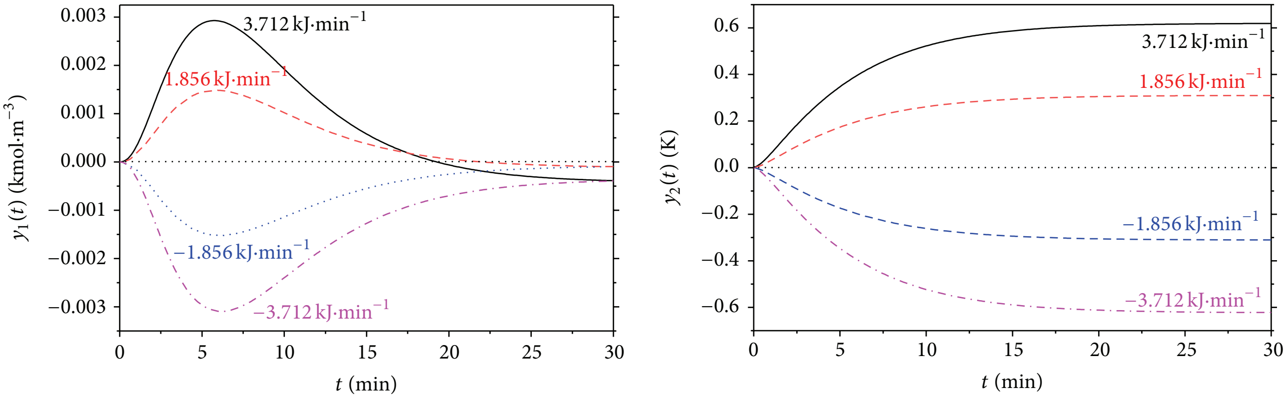


FIGURE 3: Results of dynamic analysis for the step changes of the heat removal of the cooling, ΔQ_c.

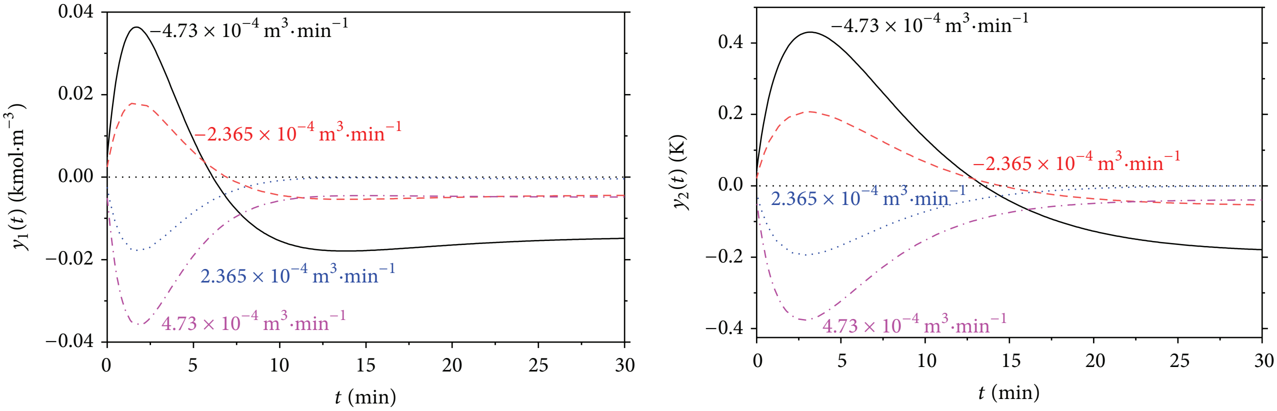


FIGURE 4: Results of dynamic analysis for the step changes of the volumetric flow rate of the reactant, Δq_r.

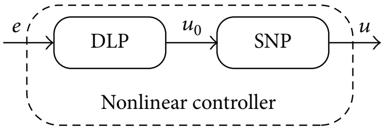


FIGURE 5: The scheme of the nonlinear controller.

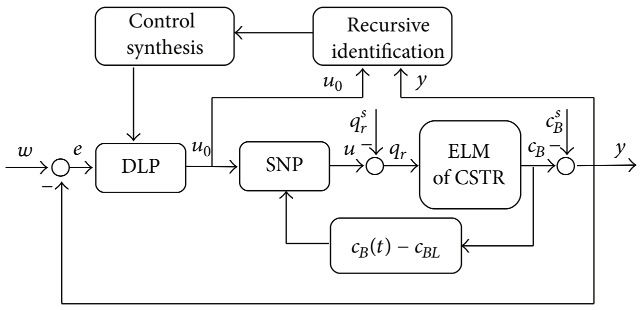


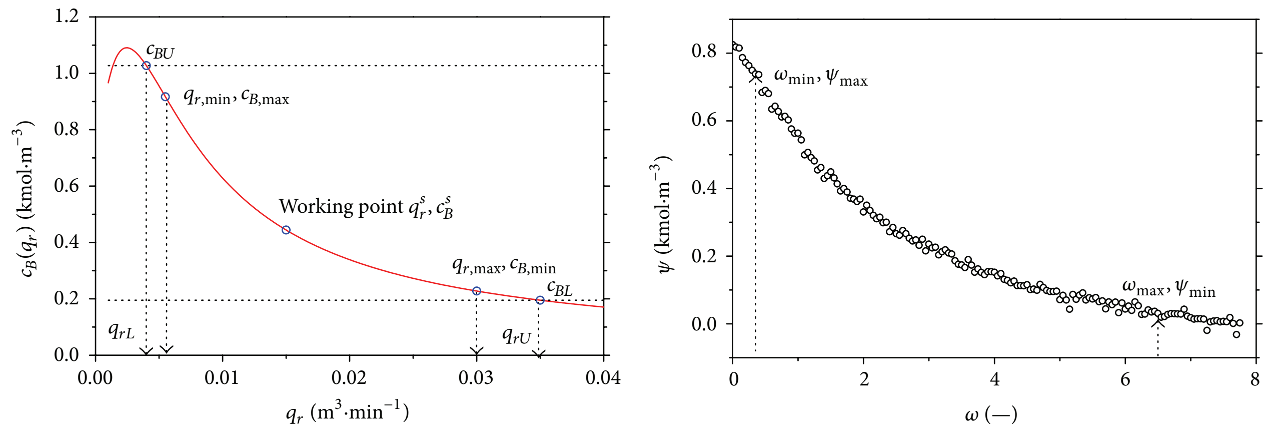
FIGURE 6: Control scheme of the nonlinear adaptive control.

If we do the steady-state characteristic for the volumetric flow rate of the reactant, q_r, from the range $q_r = \langle 0.001; 0.04 \rangle$ m$^3 \cdot$ min^{-1}, results for the steady-state values of the products concentration, c_B^s, are shown in Figure 7(a). The operation of the controller was chosen in the interval where $q_{r,\min} = 0.0055$ m$^3 \cdot$ min^{-1} and $q_{r,\max} = 0.03$ m$^3 \cdot$ min^{-1}. Working point of the system was chosen in the middle of this interval and includes also the nonlinearity of the system. This point is defined by the volumetric flow rate $q_r^s = 0.015$ m$^3 \cdot$ min^{-1} and heat removal of the coolant $Q_c^s = -18.56$ kJ·min^{-1}. The steady-state value of the controlled concentration is in this point $c_B^s = 0.442$ kmol $\cdot$ m^{-3}.

Due to later approximation and better unification of the variables, the new x and y variables ω and ψ are introduced and

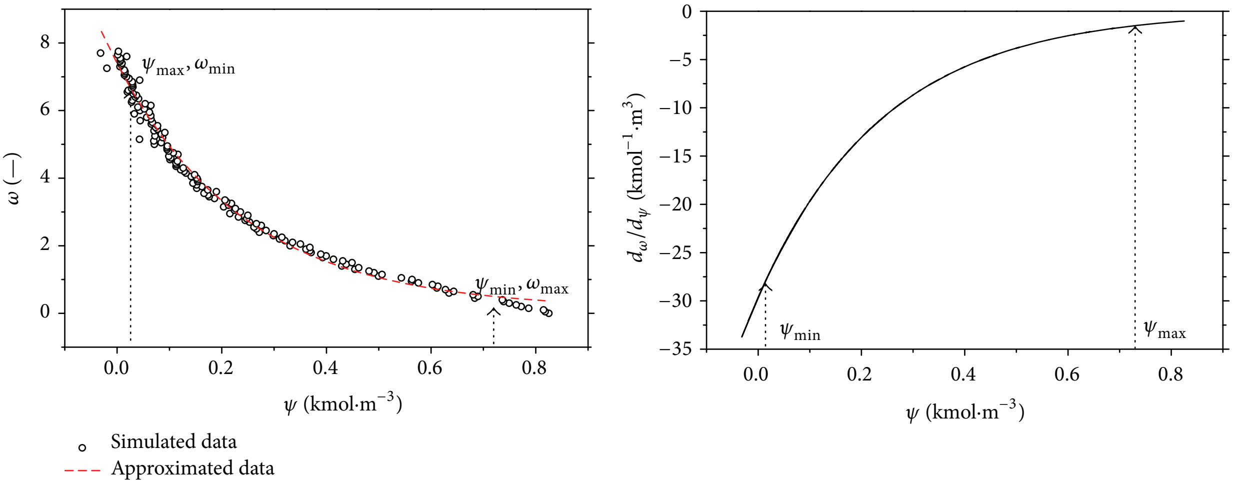
$$\omega = \frac{q_r^s - q_{rL}}{q_{rL}} \; [-]; \qquad \psi = c_B^s - c_{BL} \; \left[\mathrm{kmol} \cdot \mathrm{m}^{-3}\right], \qquad (6)$$

where q_{rL} is lower bound from the interval and c_{BL} is corresponding products concentration from the upper bound q_{rU}; see Figure 7. It is recommended to choose this interval slightly longer than those in $q_{r,\min} \le q_r(t) \le q_{r,\max}$ which



(a) (b)

Figure 7: The steady-state characteristic (a) and noised data in new coordinates $\psi = f(\omega)$ (b).



(a) (b)

Figure 8: The simulated and approximated steady-state characteristic in new coordinates (a) and the course of the derivative of $d\psi/d\omega$ (b).

means in this case that lower and upper bounds of the input variable and equivalent values of the concentrations are

$$\begin{aligned} q_{rL} &= 0.004\,\mathrm{m}^3 \cdot \mathrm{min}^{-1}, \\ q_{rU} &= 0.035\,\mathrm{m}^3 \cdot \mathrm{min}^{-1}, \\ c_{BL} &= 0.1953\,\mathrm{kmol} \cdot \mathrm{m}^{-3}, \\ c_{BU} &= 1.0274\,\mathrm{kmol} \cdot \mathrm{m}^{-3}. \end{aligned} \tag{7}$$

It is common that the measured data on the real system are affected by the measurement errors—see Figure 7(b) for new coordinates. To emulate these errors, the random white-noise error on the output variable is introduced here and the values for new coordinates with noised data are shown in Figure 7(b).

The difference of the input volumetric flow rate of the coolant is from (5) $u(t) = \Delta q_r(t)$ and the nonlinear part can be then computed from

$$u(t) = \Delta q_r(t) = q_{rL} \left(\frac{d\omega}{d\psi} \right)_{\psi(c_B)} u_0(t). \tag{8}$$

The values of q_{rL} and $u_0(t)$ in (8) are known and the derivative $d\omega/d\psi$ is unknown.

The procedure for computing of the value of this derivative for the specific value of products concentration, c_B, is the following. The inverse of coordinates ω and ψ is done first; see Figure 8(a). Then, the data are approximated, for example, by the exponential, polynomial, and so forth, functions.

For example, the exponential function in the general form

$$\omega = f(\psi) = a \cdot e^{-b \cdot \psi} + c \tag{9}$$

was used in this case. The course of this approximation is shown in Figure 8(a) (red dashed line) with the identified values of constants $a = 7.1601$, $b = 4.1806$, and $c = 0.1707$.

As there is the derivative $d\omega/d\psi$ in (8), this derivative is in this case

$$\frac{d\omega}{d\psi} = -29.9335 \cdot e^{-4.1806\cdot\psi}. \tag{10}$$

The course of this function is shown in Figure 8(b).

3.2. External Linear Model of CSTR. The dynamic behavior of the controlled system, in our case CSTR, together with the SNP derived above is observed for the step responses of the input u_0; see Figure 2. Five changes u_0 were done for the working point defined by input values $q_r^s = 0.015\,\mathrm{m}^3 \cdot \mathrm{min}^{-1}$ and $Q_c^s = -18.56\,\mathrm{kJ} \cdot \mathrm{min}^{-1}$ and results are shown in Figure 9.

The gain of the system SNP+CSTR is computed as

$$g_s = \lim_{t\to\infty} \frac{y(t)}{u_0} \tag{11}$$

and the values of this gain, g_s, are shown also in Figure 9.

Although the system has nonlinear behavior, presented output dynamic responses could be described by the first order continuous-time transfer function

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b(s)}{a(s)} = \frac{b_0}{s + a_0} \tag{12}$$

with s as a complex variable and polynomials $a(s)$ and $b(s)$ come from identification. This transfer function could be then in the form of the differential equation

$$\dot{y}(t) + a_0 y(t) = b_0 u(t). \tag{13}$$

3.3. Identification of the ELM. The online identification of the continuous-time ELM (12) is not very simple. On the other hand, δ-identification models belong to the class of discrete models but their parameters are close to the continuous ones for very small sampling period.

The delta-model introduces a new complex variable γ as an alternative to complex variables s in continuous-time and z in discrete-time. The so-called forward δ-model for $\beta = 0$ was used here with the γ operator:

$$\gamma = \frac{z-1}{T_v}, \tag{14}$$

where T_v is a sampling period and z is a discrete-time complex variable.

The continuous model (12) is then rewritten to the form

$$a^{\delta}(\delta)\, y(t') = b^{\delta}(\delta)\, u(t'), \tag{15}$$

where polynomials $a^{\delta}(\delta)$ and $b^{\delta}(\delta)$ are discrete polynomials and their coefficients are different from those of the CT models $a(s)$ and $b(s)$ in (12). Time t' denotes discrete-time.

Equation (13) could be then with the substitution and simplifications rewritten to

$$y_{\delta}(k) = -a_0^{\delta} y_{\delta}(k-1) + b_0^{\delta} u_{\delta}(k-1), \tag{16}$$

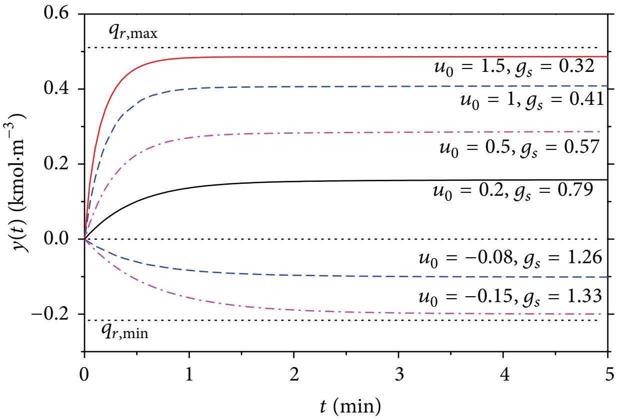


Figure 9: Results of dynamic analyses for the changes of input u_0.

where new, recomputed, values of input and output variables are

$$\begin{aligned} y_{\delta}(k) &= \frac{y(k) - y(k-1)}{T_v}; \\ y_{\delta}(k-1) &= y(k-1); \\ u_{\delta}(k-1) &= u(k-1). \end{aligned} \tag{17}$$

The regression vector, ϕ_{δ}, and vector of parameters, θ_{δ}, used for identification are then

$$\begin{aligned} \varphi_{\delta}(k-1) &= \left[-y_{\delta}(k-1), u_{\delta}(k-1)\right]^T; \\ \theta_{\delta}(k) &= \left[a_0^{\delta}, b_0^{\delta}\right]^T \end{aligned} \tag{18}$$

and the differential equation (16) could be rewritten to the vector form:

$$y_{\delta}(k) = \theta_{\delta}^T(k) \cdot \varphi_{\delta}(k-1) + e(k), \tag{19}$$

where $e(k)$ is a general random immeasurable component. The task of the recursive identification is to find unknown vector of parameters, θ_{δ}, from the measured data vector ϕ_{δ}. The simple recursive least-squares (RLS) method was used in this work. This method together with exponential and directional forgetting modifications produces sufficient results as it was proofed by the previous experiments.

3.4. Dynamic Linear Part. The last part from Figure 6 which has not been discussed is the dynamic linear part (DLP). The feedback controller with one degree-of-freedom (1DOF) is designed with the use of polynomial approach [7].

The scheme of this control configuration is shown in Figure 10, where w represents reference signal (wanted value), e is control error ($e = w - y$), u is control signal, v is immeasurable error, and y is controlled output from the system. The block $G(s)$ is controlled system described by the transfer function (12) and $Q(s)$ is feedback controller, the transfer function of which has general polynomial form:

$$\widetilde{Q}(s) = \frac{q(s)}{s \cdot \widetilde{p}(s)}, \tag{20}$$

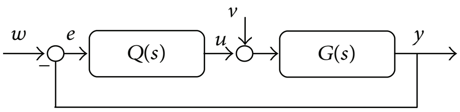


Figure 10: One degree-of-freedom (1DOF) control configuration.

where parameters of polynomials and $q(s)$ are computed from Diophantine equation:

$$a(s) \cdot s \cdot \widetilde{p}(s) + b(s) \cdot q(s) = d(s). \tag{21}$$

On the other hand, polynomials of the ELM $a(s)$ and $b(s)$ in (21) are known from the recursive identification and we expect that parameters of this polynomial are very close to parameters of correspondent δ polynomials $a^{\delta}(\delta)$ and $b^{\delta}(\delta)$ in (16). The polynomial $d(s)$ on the right side of (21) is an optional stable polynomial and the degree of this polynomial is $\deg d(s) = \deg a(s) + \deg \widetilde{p}(s) + 1$. Roots of this polynomial are called *poles of the closed-loop* and their position affects quality of the control.

There are several ways to construct this optional polynomial, for example, the pole-placement method, LQ approach, and so forth. The choice here combines the pole-placement method with spectral factorization of the identified polynomial $a(s)$. The polynomial $d(s)$ has then two parts:

$$d(s) = n(s) \cdot (s + \alpha)^2, \tag{22}$$

where $\alpha > 0$ is an optional coefficient reflecting closed-loop poles and stable polynomial $n(s)$ is obtained from the spectral factorization of the polynomial $a(s)$—$n^{*}(s) \cdot n(s) = a^{*}(s) \cdot a(s)$, which is known from the recursive identification. The transfer function of the controller (20) is for this concrete ELM (12):

$$\widetilde{Q}(s) = \frac{q(s)}{s \cdot \widetilde{p}(s)} = \frac{q_1 s + q_0}{p_0 \cdot s} \tag{23}$$

and parameters q_1, q_0, and p_0 are computed from Diophantine equation (21).

The control synthesis presented above is derived in the continuous-time, but identification and recomputation of the controllers parameters run in discrete-time (δ-models). That is why we call this controller *hybrid adaptive controller*.

4. Classic Adaptive Control

It is good to show how the nonlinear adaptive control could improve classic adaptive control described, for example, in [10] or [11].

Let us consider the control configuration displayed in Figure 11 without the SNP part.

This means that system is controlled only with the use of adaptive controller based on the ELM without the knowledge about static behavior of the system. The design and computation of the controller are the same as what is described in Sections 3.2–3.4. The only difference is that the input variable to the ELM is here $u(t)$ unlike $u_0(t)$ in the nonlinear adaptive control described above.

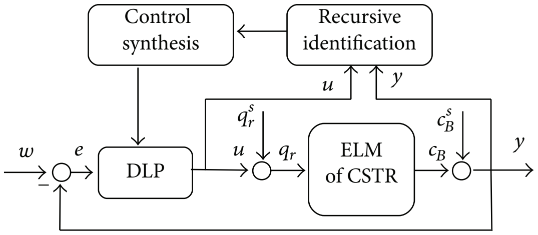


Figure 11: Control scheme of the classic adaptive control.

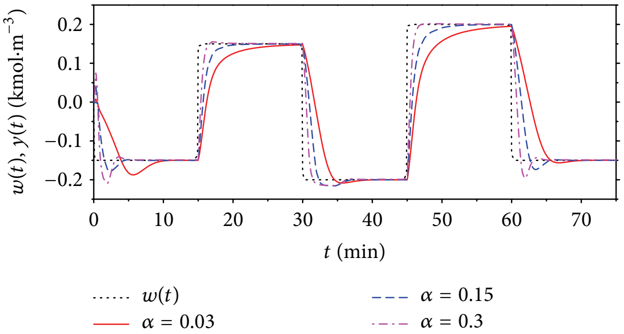


Figure 12: The course of the output variable $y(t)$ and the reference signal $w(t)$ for various values of the tuning parameter α for nonlinear adaptive control.

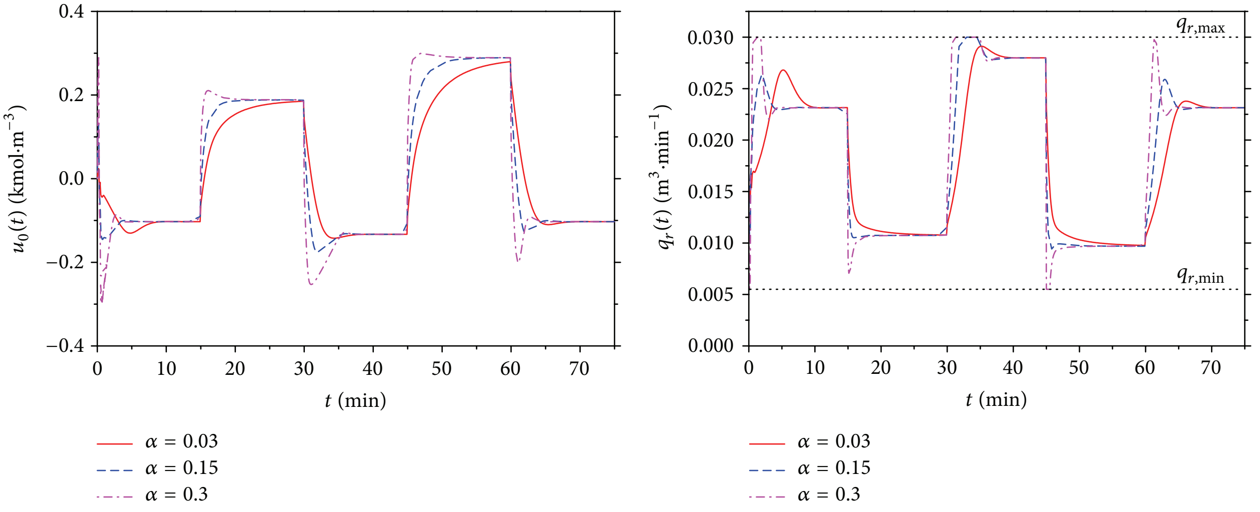
Results of this control are displayed and commented on in the next section.

5. Simulation Experiment

The goal of this last section is to verify proposed classic and nonlinear adaptive controllers by simulations on the mathematical model (1) of the CSTR. The simulations were done for three values of the α from (22) which could be understood as a tuning parameter. The sampling period was $T_v = 0.1$ min, the simulation time was 75 min, and 5 step changes of the reference signal $w(t)$ were done during this time.

Figure 12 shows courses of the output variable, $y(t)$, for various $\alpha = 0.03$, 0.15, and 0.3. It is clear that the increasing value of this parameter results in the quicker output response but overshoots especially for the negative step changes. On the other hand, Figure 13 shows the course of the input variable u_0 (Figure 13(a)) as an output from the DLP which is also input to the SNP. Figure 13(b) is the course of the volumetric flow rate q_r as an output from the SNP and the input to the mathematical model of CSTR; see schematic representation in Figure 6. We can say that decreasing value of the parameter α results in smoother course of both input variables.

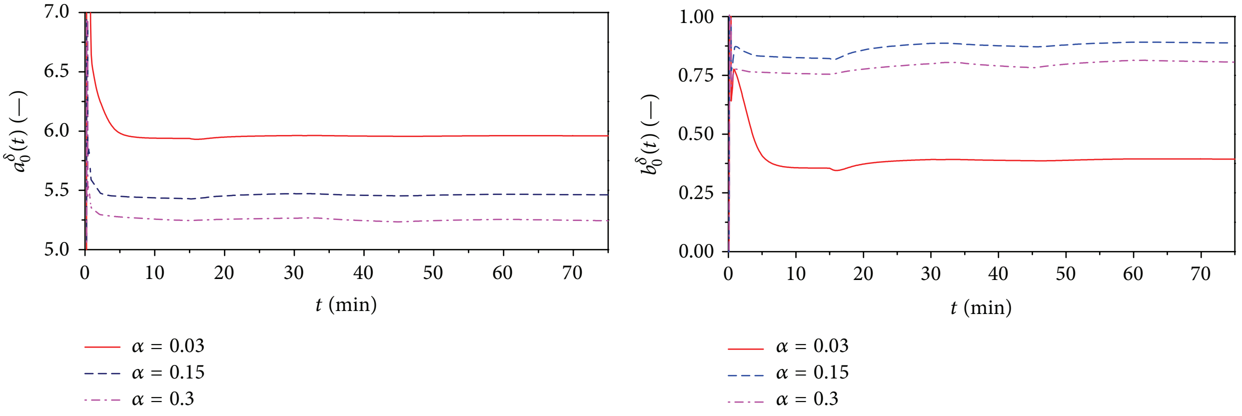
The course of identified parameters during the control is shown in Figure 14. Graphs show usability of proposed recursive least-squares method with exponential forgetting that is used for online identification of the ELM. The only problem could be found at the very beginning of the control because it needs some initial time to stabilize the parameters



(a)

(b)

Figure 13: Outputs from the LDP u_0 or various values of α (a) and the courses of the computed input variable, q_r, to the ELM for various α (b).



(a)

(b)

Figure 14: The course of identified parameters a_0^δ (a) and b_0^δ (b) for nonlinear adaptive controller.

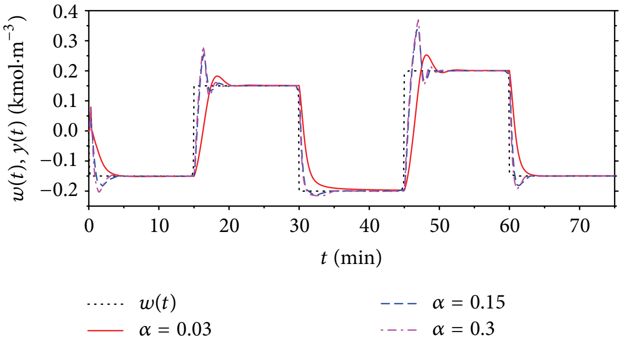


Figure 15: The course of the output variable $y(t)$ and the reference signal $w(t)$ for various values of the tuning parameter α for classic adaptive control.

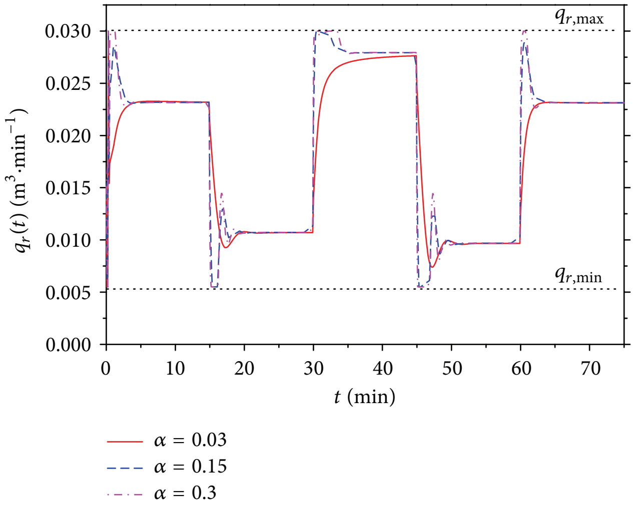


FIGURE 16: The courses of the computed input variable, q_r, to the ELM for various α for classic adaptive control.

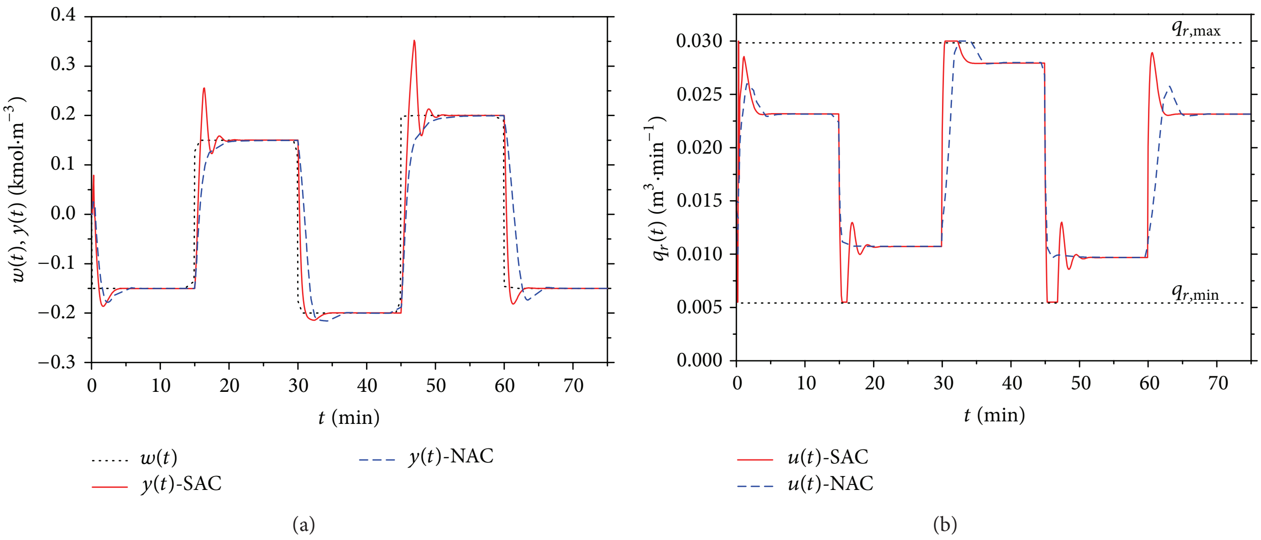


FIGURE 17: Comparison of resulted courses of output variable $y(t)$ (a) and input variable $q_r(t)$ (b) for nonlinear and classic adaptive control for $\alpha = 0.15$.

as the identification starts from general values of the vector of parameters $\theta_\delta = [0.1 \;\; 0.1]^T$.

The task of this contribution was also to show improvement of the nonlinear adaptive approach compared with the classic adaptive control described in Section 4. The simulation studies were done for the same values of the root position $\alpha = 0.03$, 0.15, and 0.3 and results are shown in Figures 15 and 16.

Compared control results for $\alpha = 0.15$ are shown in Figure 17. Results for both comparisons have shown that nonlinear adaptive control produces better control results especially for the positive changes of the reference signal $w(t)$. The improvement is evident also for the course of the input value $u(t)$ which could be also very important from the practical point of view.

6. Conclusion

The paper deals with the adaptive control of the CSTR as a typical member of the nonlinear system with lumped parameters. The mathematical model of such system is described by the set of four nonlinear ordinary differential equations and simulation is in this case related to the numerical solution of this set of ODEs. The static and dynamic analysis have shown high nonlinearity of this system which means that controlling of such process with conventional control methods could lead to suboptimal or even very bad control results. The adaptive control is one way to how we can overcome this problem. The adaptive approach here was based on the choice of the delta external linear model of the originally nonlinear system, parameters of which are identified recursively during the control, and the parameters of the controller are also recomputed according to these identified ones. This method satisfies appropriate reaction of the controller to the change of the state of the system or the random disturbance. The control synthesis employs polynomial theory together with the pole-placement method and spectral factorization. These methods satisfy basic control requirements such as stability, reference signal tracking, and disturbance attenuation. The contribution shows also the improvement of this so-called classic adaptive control by the nonlinear theory which is based on the Wiener system where the controller is divided into the dynamic linear part and the static nonlinear part. The dynamic linear part is the same as in classic adaptive control but the static nonlinear part uses simulated or measured steady-state characteristics of the mathematical model to describe the relation between controlled concentration of the product and the change of the reactants volumetric flow rate as an input variable. Both controllers could be tuned by the choice of the parameter α as a position of the root in the pole-placement method. Presented results have shown that increasing value of this parameter results in quicker output response but with overshoots for both standard and nonlinear adaptive controllers. Comparison of both controllers with the same settings has shown better control results for nonlinear adaptive control especially for the positive step changes of the reference signal. Although the system has nonlinear behavior, proposed control strategies cope with it well and it could be used also for similar types of systems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] J. Ingham, I. J. Dunn, E. Heinzle, and J. E. Prenosil, *Chemical Engineering Dynamics. An Introduction to Modeling and Computer Simulation*, VCH, Weinheim, Germany, 2nd edition, 2000.

[2] K. J. Astrom and B. Wittenmark, *Adaptive Control*, Addison Wesley, Reading, Mass, USA, 1989.

[3] V. Bobal, J. Böhm, J. Fessl, and J. Machácek, *Digital Self-tuning Controllers: Algorithms, Implementation and Applications*, Advanced Textbooks in Control and Signal Processing, 2005.

[4] A. Astolfi, D. Karagiannis, and R. Ortega, *Nonlinear and Adaptive Control with Applications*, Springer, London, UK, 2008.

[5] T. L. Vincent and W. J. Grantham, *Nonlinear and Optimal Control Systems*, John Wiley & Sons, New York, NY, USA, 1997.

[6] M. Nakamura, T. Sugi, and S. Goto, "Nonlinear sep aration model and control for a complex process realized by conventional PID controller hardware," in *Proceedings of the 4th Asian Control Conference*, pp. 274–279, Singapore, 2002.

[7] V. Kučera, "Diophantine equations in control—a survey," *Automatica*, vol. 29, no. 6, pp. 1361–1375, 1993.

[8] H. Middleton and G. C. Goodwin, *Digital Control and Estimation—A Unified Approach*, Prentice Hall, Englewood Cliffs, NJ, USA, 2004.

[9] D. L. Stericker and N. K. Sinha, "Identification of continuous-time systems from samples of input-output data using the δ-operator," *Control Theory and Advanced Technology*, vol. 9, no. 1, pp. 113–125, 1993.

[10] J. Vojtesek, P. Dostal, and R. Haber, "Simulation and control of a continuous stirred tank reactor," in *Proceedings of the 6th Portuguese Conference on Automatic Control (CON-TROLO '04)*, pp. 315–320, Faro, Portugal, 2004.

[11] J. Vojtesek and P. Dostal, "From steady-state and dynamic analysis to adaptive control of the CSTR reactor," in *Proceedings of the 19th European Conference on Modelling and Simulation (ECMS '05)*, pp. 591–598, Riga, Latvia, June 2005.

[12] H. Chen, A. Kremling, and F. Allgwer, "Nonlinear predictive control of a benchmark CSTR," in *Proceedings of the 3rd European Control Conference*, Rome, Italy, 1995.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 205749, 10 pages
http://dx.doi.org/10.1155/2015/205749

Research Article

ECG Prediction Based on Classification via Neural Networks and Linguistic Fuzzy Logic Forecaster

Eva Volna, Martin Kotyrba, and Hashim Habiballa

University of Ostrava, 30 Dubna 22, 70103 Ostrava, Czech Republic

Correspondence should be addressed to Martin Kotyrba; martin.kotyrba@osu.cz

Received 16 July 2014; Accepted 20 November 2014

Academic Editor: Mohammed Chadli

The paper deals with ECG prediction based on neural networks classification of different types of time courses of ECG signals. The main objective is to recognise normal cycles and arrhythmias and perform further diagnosis. We proposed two detection systems that have been created with usage of neural networks. The experimental part makes it possible to load ECG signals, preprocess them, and classify them into given classes. Outputs from the classifiers carry a predictive character. All experimental results from both of the proposed classifiers are mutually compared in the conclusion. We also experimented with the new method of time series transparent prediction based on fuzzy transform with linguistic IF-THEN rules. Preliminary results show interesting results based on the unique capability of this approach bringing natural language interpretation of particular prediction, that is, the properties of time series.

1. Background

Biometrical data is typically represented as an image or a quantification of measured physiological or behavioural characteristics. As this data should refer to very complex human behaviour or describe very precisely physiological characteristic (typically iris scan, fingerprint, palm vein image, hand scan, voice, walk pattern, etc.), this data can easily become very large and hard to process. For this reason, modern ways of data processing and classification are applied for biometrical data. The leading method is the usage of neural networks [1].

For more than four decades, computers have been used in the classification of the electrocardiogram (ECG) resulting in a huge variety of techniques [2] all designed to enhance the classification accuracy to levels comparable to that of a "gold standard" of expert cardiology opinion. Included in these techniques are multivariate statistics, decision trees, fuzzy logic, expert systems, and hybrid approaches [3]. The recent interest in neural networks coupled with their high levels of performance has resulted in many instances of their application in this field [4].

The electrocardiogram is a technique of recording bioelectric currents generated by the heart. Clinicians can evaluate the conditions of a patient's heart from the ECG and perform further diagnosis. ECG records are obtained by sampling the bioelectric currents sensed by several electrodes, known as leads. A typical one-cycle ECG tracing is shown in Figure 3.

1.1. Backpropagation Neural Networks. A neural network is a parallel, distributed information processing structure consisting of processing elements (which can possess a local memory and can carry out localized information processing operations) interconnected together with unidirectional signal channels called connections. Each processing element has a single output connection which branches into as many collateral connections as desired (each carrying the same signal, the processing element output signal). The processing element output signal can be of any mathematical type desired. All of the processing that goes on within each processing element must be completely local: that is, it must depend only upon the current values of the input signals

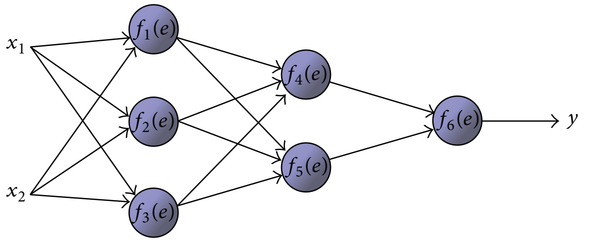


FIGURE 1: A backpropagation network architecture.

arriving at the processing element via impinging connections and upon values stored in the processing element's local memory [5].

The backpropagation neural network architecture is a hierarchical design consisting of fully interconnected layers or rows of processing units (with each unit itself comprised of several individual processing elements). Backpropagation belongs to the class of mapping neural network architectures and therefore the information processing function that it carries out is the approximation of a bounded mapping or function $f : A \subset R^n \rightarrow R^m$, from a compact subset $\mathbf{A}$ of n-dimensional Euclidean space to a bounded subset $f[\mathbf{A}]$ of m-dimensional Euclidean space, by means of training on examples (x_1, z_1), (x_2, z_2),…,(x_k, z_k)…. It will always be assumed that such examples of a mapping f are generated by selecting $\mathbf{x_k}$ vectors randomly from $\mathbf{A}$ in accordance with a fixed probability density function $p(\mathbf{x})$. The operational use to which the network is to be put after training is also assumed to involve random selections of input vectors $\mathbf{x}$ in accordance with $p(\mathbf{x})$. The backpropagation architecture described in this paper is the basic, classical version (Figure 1). The backpropagation learning algorithm is composed of two procedures: (a) forward propagation of signals and (b) backpropagation weight training [5].

Feed-Forward. Assume that each input factor in the input layer is denoted by x_i; the y_j and z_k represent the output in the hidden layer and the output layer, respectively. And the y_j and z_k can be expressed as follows (1):

$$\begin{aligned} y_j &= f\left(X_j\right) = f\left(w_{oj} + \sum_{i=1}^{I} w_{ij} x_i\right), \\ z_k &= f\left(Y_k\right) = f\left(w_{ok} + \sum_{j=1}^{J} w_{ik} y_j\right), \end{aligned} \tag{1}$$

where the w_{oj} and w_{ok} are the bias weights for setting threshold values, f is the activation function used in both hidden and output layers, and X_j and Y_k are the temporarily computing results before applying activation function f. In this study, a sigmoid function is selected as the activation function. Therefore, the actual outputs y_j and z_k in hidden and output layers, respectively, can be also written as

$$\begin{aligned} y_j &= f\left(X_j\right) = \frac{1}{1 + e^{-X_j}}, \\ z_k &= f\left(Y_k\right) = \frac{1}{1 + e^{-Y_k}}. \end{aligned} \tag{2}$$

The activation function f introduces the nonlinear effect to the network and maps the result of computation to a domain (0, 1). This sigmoid function is differentiable. The derivative of the sigmoid function in (2) can be easily derived as $f' = f(1 + -f)$.

Backpropagation Weight Training. The error function is defined as

$$E = \frac{1}{2} \sum_{k=1}^{K} e_k^2 = \sum_{k=1}^{K} \left(t_k - z_k\right)^2, \tag{3}$$

where t_k is a predefined network output (or desired output or target value) and e_k is the error in each output node. The goal is to minimize E so that the weight in each link is accordingly adjusted and the final output can match the desired output. To get the weight adjustment, the gradient descent strategy is employed. In the link between hidden and output layers, computing the partial derivative of E with respect to the weight w_{jk} produces

$$\frac{\partial E}{\partial w_{jk}} = -e_k {f'}^{(Y_k) y_j} = -\delta_k y_j \quad \text{where } \delta_k = \left(t_k - z_k\right) f'\left(Y_k\right). \tag{4}$$

The weight adjustment in the link between hidden and output layers is computed by $\Delta w_{jk} = \alpha \times y_j \times \delta_k$, where α is the learning rate, a positive constant between 0 and 1. The new weight herein can be updated by the following $w_{jk}(n + 1) = w_{jk}(n) + \Delta w_{jk}(n)$, where n is the number of iterations. Similarly, the error gradient in links between input and hidden layers can be obtained by taking the partial derivative with respect to w_{ij} as

$$\frac{\partial E}{\partial w_{ij}} = -\Delta_j x_j = f'\left(X_j\right) \sum_{k=1}^{K} \partial_k w_{jk}. \tag{5}$$

The new weight in the hidden-input links can be now corrected as $\Delta w_{ij} = \alpha \times x_i \times \Delta_j$ and $w_{ij}(n + 1) = w_{ij}(n) + \Delta_j$. Training the BP-networks with many samples is sometimes a time-consuming task. The learning speed can be improved by introducing the momentum term η. Usually, η falls in the range $\langle 0, 1 \rangle$. For the iteration n, the weight change Δw can be expressed. The backpropagation learning algorithm used in artificial neural networks is shown in many text books [3–6].

1.2. Fuzzy Logic. Fuzzy logics form heterogeneous family of formalisms capable of successful modelling of uncertain and vague information processing [7]. The usage of fuzzy logic for analysis and prediction of time series can be perceived as a complement method to neural network based methods. The symbolic background of fuzzy logic brings an advantage of human readable symbolic representation of prediction interpretation. It does not necessarily mean that fuzzy logic based time series analysis is more accurate and more efficient but its power lies in transparent and interpretable results that it gives [8–11].

Time series analysis and prediction are an important task that can be used in many areas of practice. The task of getting the best prediction to given series may bring interesting engineering applications in wide number of areas like economics, geography, or industry. Solution to the problem of obtaining best results in prediction of time series can be based on well-known and simple methods like Winters or Linear method. In this paper, we use a method based on two methods originally developed by members of Institute for Research and Applications of Fuzzy Modeling, which is a part of University of Ostrava. The aim of the paper is not to present the details of the methods already published but to present a tool implementing them. The first method is based on the notion of F-transform (fuzzy transform) devised by the group of Professor Perfilieva et al. [12]. The second approach uses the linguistic rules utilizing fuzzy logic and deduction that is a well-known formalism with very good results in variety of practical applications like industrial ones.

The idea of the fuzzy transform is to transform a given function defined in one space into another, usually simpler space, and then to transform it back. The simpler space consists of a finite vector of numbers. The reverse transform then leads to a function, which approximates the original one. More details can be found in [12].

The fuzzy transform is defined with respect to a fuzzy partition, which consists of basic functions. Let $c_1 < \cdots < c_n$ be fixed nodes within $[a, b]$ such that $c_1 = a$, $c_n = b$, and $n \geq 2$. We say that fuzzy sets $A_1, \ldots, A_n \in F([a, b])$ are basic functions forming a fuzzy partition of $[a, b]$ if they fulfill the following conditions for $i = 1, \ldots, n$:

(1) $A_i(c_i) = 1$;

(2) $A_i(x) = 0$ for $x / \in (c_{i-1}, c_{i+1})$, where for uniformity of notation we put $c_0 = c_1 = a$ and $c_{n+1} = c_n = b$;

(3) A_i is continuous;

(4) A_i strictly increases on $[c_{i-1}, c_i]$ and strictly decreases on $[c_i, c_{i+1}]$;

(5) for all $x \in [a, b]$,

$$\sum_{i=1}^{n} A_i(x) = 1. \tag{6}$$

Let a fuzzy partition of $[a, b]$ be given by basic functions $A_1, \ldots, A_n$, $n \geq 2$ and let $f: [a, b] \rightarrow R$ be a function that is known on a set $\{x_1, \ldots, x_T\}$ of points.

The n-tuple of real numbers $[F_1, \ldots, F_n]$ given by

$$F_i = \frac{\sum_{t=1}^{T} f(x_t) A_i(x_t)}{\sum_{t=1}^{T} A_i(x_t)}, \quad i = 1, \ldots, n, \tag{7}$$

is a fuzzy transform of f with respect to the given fuzzy partition.

The numbers $F_1, \ldots, F_n$ are called the components of the fuzzy transform of f.

Let $F_n[f]$ be the fuzzy transform of f with respect to $A_1, \ldots, A_n \in F([a, b])$.

Then the function $f_{F,n}$ given on $[a, b]$ by

$$f_{F,n}(x) = \sum_{i=1}^{n} F_i A_i(x) \tag{8}$$

is called the inverse fuzzy transform of f.

Fuzzy IF-THEN rules can be understood as a specific conditional sentence of natural language of the form IF X_1 is A_1 AND $\cdots$ AND X_n is A_n THEN Y is B, where $A_1, \ldots, A_n$ and B are evaluative expressions (very small, roughly big, etc.). An example fuzzy IF-THEN rule is as follows.

IF the number of cars sold in the current year is more or less small and the half-year sales increment is medium, THEN the upcoming half-year increment will be medium.

The part of the rule before THEN is called the antecedent and the part after it is consequent. Fuzzy IF-THEN rules are usually gathered in a linguistic description:

$$\begin{aligned} R_1 &:= \text{IF } X_1 \text{ is } A_{11} \text{ AND} \cdots \text{AND } X_n \\ &\quad \text{is } A_{1n} \text{ THEN } Y \text{ is } B_1, \\ &\quad \vdots \\ R_m &:= \text{IF } X_1 \text{ is } A_{m1} \text{ AND} \cdots \text{AND } X_n \\ &\quad \text{is } A_{mn} \text{ THEN } Y \text{ is } B_m. \end{aligned} \tag{9}$$

Time series prediction based on these two main approaches works as follows. Let time series x_t, $t = 1, \ldots, T$ be viewed as a discrete function x on a time axis t. Then $F_n[x] = [X_1, \ldots, X_n]$ is the fuzzy transform of the function x with respect to a given fuzzy partition. The inverse fuzzy transform then serves us as a model of the trend-cycle of a given time series. By subtracting the trend-cycle (inverse fuzzy transform) values from the time series lags, we get pure seasonal components. This is how the fuzzy transform helps us to model and decompose a given time series.

Logical dependencies between components $X_1, \ldots, X_n$ of the fuzzy transform may be described by the fuzzy rules. These rules are generated automatically from the given data and are used for forecasting the next components. Fuzzy transform components as well as their first and second order differences are used as antecedent variables. For forecasting future fuzzy transform components based on the generated fuzzy rules, a special inference method—perception-based logical deduction is used. The seasonal components are forecasted autoregressively. Finally, both forecasted components, trend-cycle and seasonal, are composed together to obtain the forecast of time series lags. These methods are integrated into an implementation, PC application called *linguistic fuzzy logic forecaster (LFLF)*, which enables as to produce linguistic descriptions that describe properties of data treated like a time series.

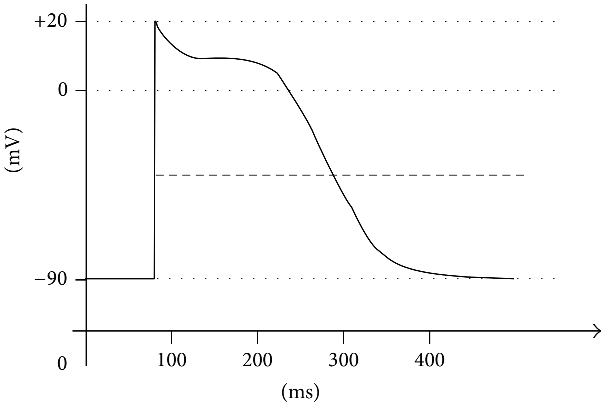


Figure 2: The cardiac action potentials.

2. Basic Principles of ECG Evaluation

ECG scanning has its own rules, which are in accordance with the laws of physics. The heart irritation spreads in all directions. In the case that the depolarisation spreads towards the electrode, which is placed on the body surface, a positive deflection is recorded on an ECG monitor. A negative deflection is recorded at the opposite end of the body. The ECG waveform is written with a chart speed of 25 mm·s^{-1}. An algorithm describing the curve goes in the following steps. First, we evaluate the shape and rhythm of ventricular complexes or atrial, which can be either regular or irregular. Then we evaluate the frequency of ventricular complexes and atrial fibrillations. Contraction of each muscle of the human body (and thus the heart as well) is associated with electrical changes called depolarization, which can be detected by electrodes. The heart contains two basic types of cells: myocardial cells, which are responsible for generating the pressure necessary to pump blood throughout the body, and conduction cells, which are responsible for rapidly spreading electrical signals to the myocardial cells in order to coordinate pumping. A graph of an action potential of a muscle of cardiac cells is shown in Figure 2.

A normal electrocardiogram is illustrated in Figure 3. The figure also includes definitions for various segments and intervals in the ECG. The deflections in this signal are denoted in alphabetic order starting with the letter *P*, which represents atrial depolarization. The ventricular depolarization causes the *QRS* complex, and repolarization is responsible for the T-wave. Atrial repolarization occurs during the *QRS* complex and produces such a low signal amplitude that it cannot be seen apart from the normal ECG.

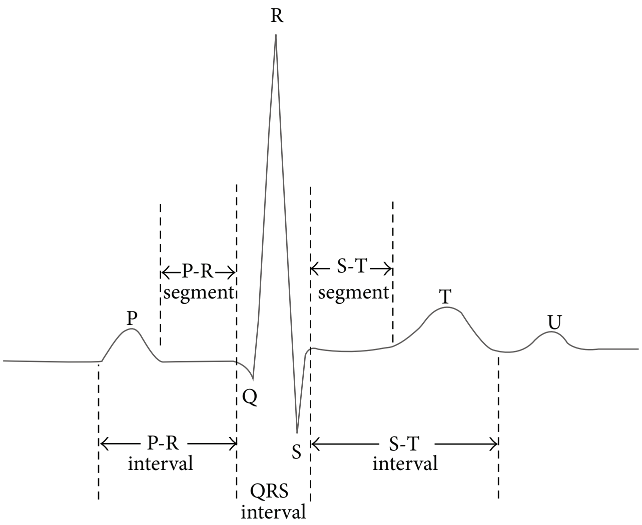


Figure 3: A typical one-cycle ECG tracing (adapted from http://www.ni.com/white-paper/app/largeimage?lang=cs&imageurl=%2Fcms%2Fimages%2Fdevzone%2Ftut%2F2007-07-09_141618.jpg).

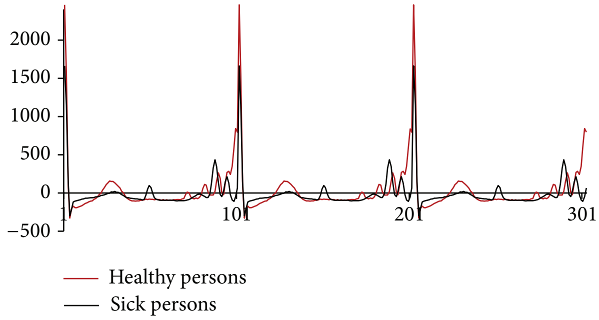


Figure 4: Comparison of mean values of ECG waveforms for healthy/sick persons.

3. Signal Processing Using Neural Networks and Fuzzy Logic

In practice, a relatively reliable diagnostic program stored in ECG monitors has been used, which is a guideline for determining the final diagnosis of heart disorders. This program works according to the principle of IF-THEN rules. The values of the electrical signal are discretized and uploaded into expert systems in the form of thousand rules. The aim of this paper is to use a different approach based on the principle of neural networks. The proposed methodology of solution could be summarized into the following steps:

(1) a conversion of analog signal from the ECG monitor to a computer,

(2) using multilayer networks that are fully connected,

(3) obtaining ECG waveforms in collaboration with the University Hospital in Poruba, specifically at the Department Cardiac Surgery from sick patients and at the Department Traumatology from healthy patients (i.e., "healthy" with regard to heart diseases),

(4) ECG waveforms built training/test sets,

(5) neural network adaptation,

(6) testing phases.

3.1. Technical Equipment. ECG measurements were performed using ADDA Junior with converter ADDA Junior,

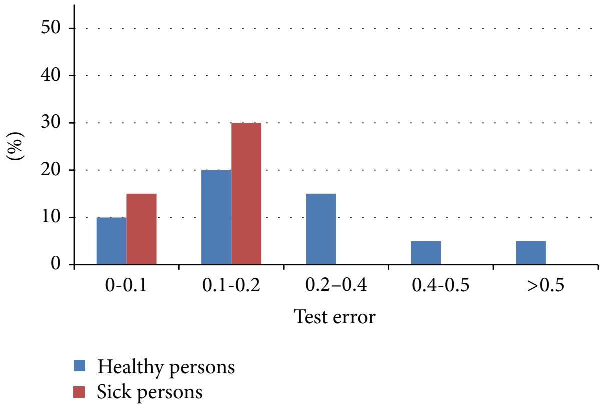


Figure 5: Experimental results, test error for healthy/sick persons.

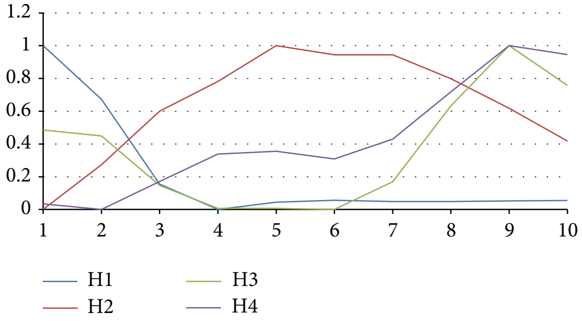


Figure 6: Patterns representing healthy persons.

which was connected to a computer via bidirectional parallel cable (CETRONICS). Technical parameters of the A/D converter (8-bit conversion) were the following:

(i) 3 measuring ranges,

(ii) measuring of a frequency of AC voltage at any channel,

(iii) autoranging for measuring the frequency of 100 Hz, 1 kHz, and 10 kHz,

(iv) input resistance of 300 kΩ,

(v) measurement accuracy 1%.

Technical parameters of the /D/A converter (a programmable voltage source ±10 V) were the following:

(i) maximum current consumption of 15 mA (after optimizing 4A at the output),

(ii) power of the converter ±15 V (stabilized).

4. Experimental Results

4.1. Time Series Classification and Prediction via Neural Networks. The training set consisted of modified ECG waveforms. We used a backpropagation neural network with topology 101-10-1. The output unit represents a diagnose 0/1, a healthy/sick person. A smaller number of inputs would not be appropriate due to the nature of the ECG waveform. We use 34 ECG time series associated with sick persons and 36 ECG time series associated with healthy persons. 25 time series of each group were used as a training set and the rest as a test set. Figure 4 shows a comparison of mean values of ECG waveforms for healthy/sick persons. We used the backpropagation method [5, 6] for the adaptation with the following parameters: the learning rate value is 0.1 and momentum is 0. The conducted experimental studies also showed that training patterns are mixed randomly in each cycle of adaptation. This ensures their greater diversity which acts as a measure of system stability. Uniform system in a crisis usually collapses entirely, while system with such diversity of trained patterns remains functional despite of crisis of its individual parts. The condition of end of the adaptation algorithm specified the limit value of the overall network error, $E < 0.1$.

The test set consisted of 20 samples (11 health and 9 sick persons) that were not included in the training set. The summary results for this type of experiment are shown in a graph in Figure 5. For clarity, the results of testing are given in percentage. The average test error was 0.194. A healthy population was detected with an average error of 0.263 and sick population with an average error of 0.109.

4.1.1. Pattern Recognition Classifier Leading to Prediction. For the purpose of adaptation of the pattern recognition classifier, it is necessary to remark that determination of training patterns is one of the key tasks. Improperly chosen patterns can lead to confusion of neural networks. During our experimental work, we made some study which included ECG pattern recognition. When creating appropriate patterns of the training set, we used characteristic curves shown as mean

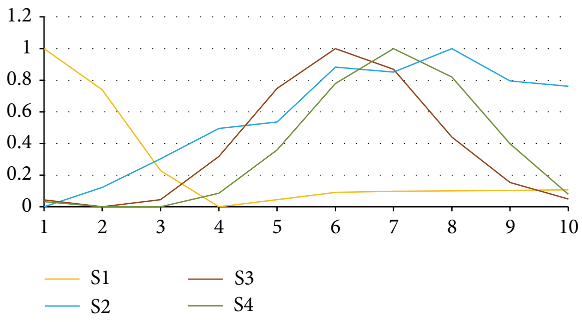


Figure 7: Patterns representing sick persons.

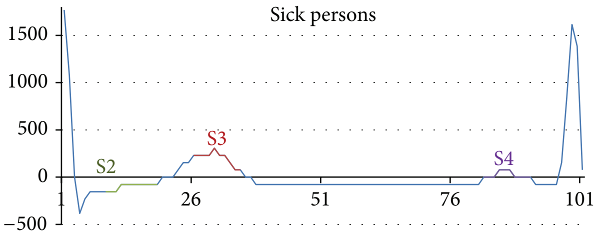


Figure 8: Some recognized patterns that occur in ECG time series.

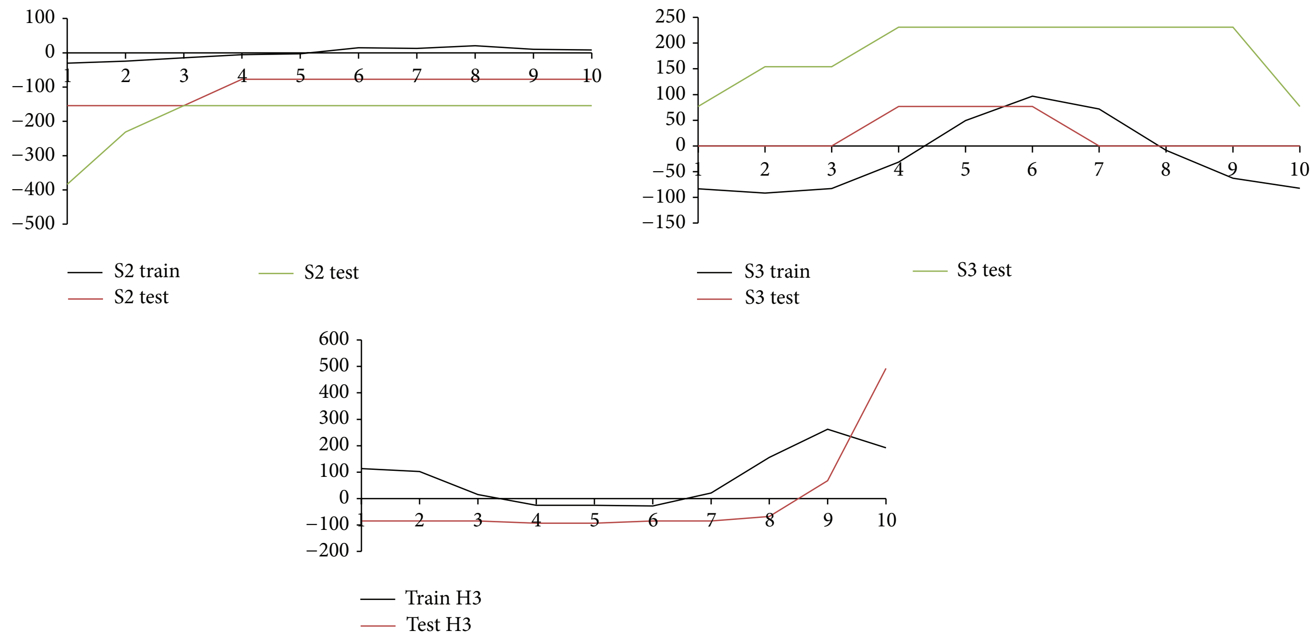


Figure 9: Training patterns, their representation in used test sets.

values from ECG waveforms for healthy and sick persons (Figure 4). We use two different groups of patterns. Patterns H1–H4 (Figure 6) represent healthy persons and patterns S1–S4 (Figure 7) represent sick persons. The whole training set is shown in Table 1.

Pattern recognition classifier is based on backpropagation neural network and is able to recognise wave structures in given time series [13, 14]. Artificial neural networks need training sets for their adaptation. In our experimental work, the training set consisted of 8 patterns representing the basic structure of the various waves in ECG graphs; see Figures 6 and 7. Input data is sequences always including n consecutive numbers, which are transformed into interval $\langle 0, 1\rangle$ by formula (10). Samples are adjusted for the needs of backpropagation networks with sigmoid activation function in this way [5, 6]:

$$x'_j = \frac{x_j - \min(x_i, \ldots, x_{i+n-1})}{\max(x_i, \ldots, x_{i+n-1}) - \min(x_i, \ldots, x_{i+n-1})}, \qquad (j = i, \ldots, i+n-1), \tag{10}$$

where x'_j is normalized output value of the jth neuron ($j = i, \ldots, i+n-1$) and $(x_i, \ldots x_{i+n-1})$ are $n-1$ consecutive output values that specify sequences (patterns) from the training set (e.g., training pars of input and corresponding output vectors). Input vector contains 10 components. Output vector has got 8 components and each output unit represents one of 8 different types of ECG wave samples. A neural network architecture is 10-10-8 (e.g., 10 units in the input layer, 10 units in the hidden layer, and 8 units in the output layer). The net is fully connected. Adaptation of the neural network starts with randomly generated weight values.

We used the backpropagation method for the adaptation with the following parameters: the learning rate value is 0.1 and momentum is 0. We have utilized our experience from earlier times; that is, training patterns were mixed randomly in each cycle of adaptation. The condition of end of the adaptation algorithm specified the limit value of the overall network error, $E < 0.1$.

In order to test the efficiency of the method, we applied the same set of data that we used in the previous experimental part. Outputs from the classifier produce sets of values that are assigned to each recognized training pattern in the given test time series. It is important to appreciate what can be considered as an effective criterion related to consensus of similarity. The proposed threshold resulting from our experimental study was determined at least $p = 70\%$. Figure 9 shows a comparison of patterns, how were learned (S2, S3, H3 train) and how were recognized in test time series (S2, S3, H3 test). The neural network is able to discover some connections, which are almost imperceptible. Illustration of some recognized patterns that occur in ECG time series is shown in Figure 8. Outputs from the classifier carry a predictive character. The neural network determines if the time series belongs to a healthy or sick person on the basis of the recognised ECG patterns which appear in the time series history.

The methodology of testing is shown in Figure 10. This means that if the test pattern S1, S2, S3, or S4 appeared in ECG waveform with probability $p_S \geq p$ ($p = 70\%$), thus it was predicted to be "a sick person." Then we work only with the remaining time series. If the test pattern H1, H2, H3, or H4 appeared in ECG waveform with probability $p_H \geq p$ ($p = 70\%$), thus it was predicted to be "a healthy person." In all other cases, the ECG time series was unspecified. We examined a total of 20 data sets. Each of them

Table 1: The training set.

Patterns	Inputs										Outputs							
H1	1.000	0.672	0.155	0.000	0.045	0.057	0.049	0.049	0.053	0.055	1	0	0	0	0	0	0	0
H2	0.000	0.273	0.600	0.782	1.000	0.945	0.945	0.799	0.618	0.418	0	1	0	0	0	0	0	0
H3	0.485	0.449	0.147	0.007	0.007	0.000	0.169	0.632	1.000	0.757	0	0	1	0	0	0	0	0
H4	0.035	0.000	0.170	0.338	0.356	0.309	0.430	0.719	1.000	0.946	0	0	0	1	0	0	0	0
S1	1.000	0.740	0.228	0.000	0.045	0.091	0.098	0.101	0.104	0.107	0	0	0	0	1	0	0	0
S2	0.000	0.123	0.304	0.495	0.536	0.883	0.851	1.000	0.796	0.761	0	0	0	0	0	1	0	0
S3	0.044	0.000	0.045	0.319	0.748	1.000	0.868	0.440	0.154	0.050	0	0	0	0	0	0	1	0
S4	0.033	0.000	0.000	0.085	0.360	0.779	1.000	0.820	0.399	0.079	0	0	0	0	0	0	0	1

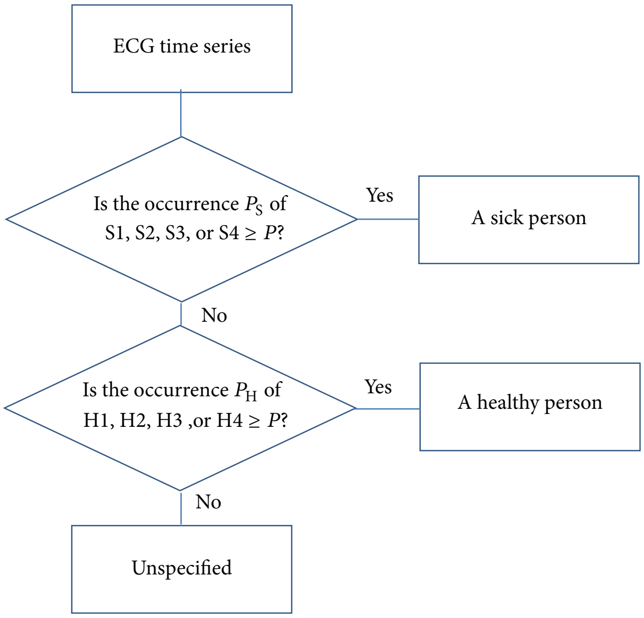


Figure 10: The methodology of testing.

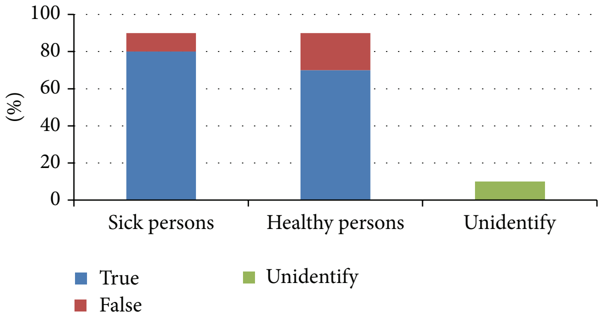
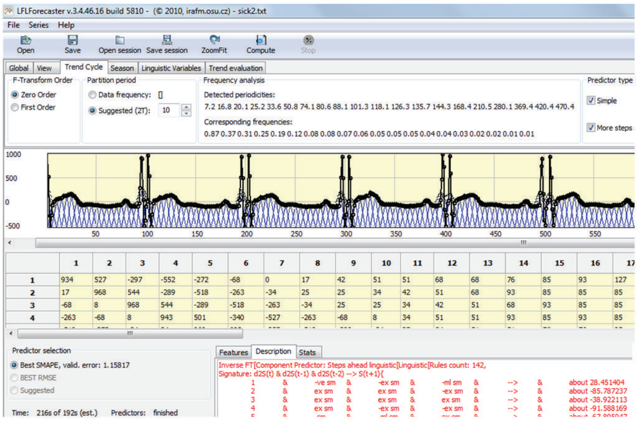


Figure 11: Experimental results, test error.

contains 101 values that assign 92 possible patterns. The whole number of examined patterns is 1840. The graph in Figure 11 demonstrates a summary of results, where "sick persons" represent patterns S1–S4 and "healthy persons" represent patterns H1–H4. The resulting prediction is based on the methodology; see Figure 10.



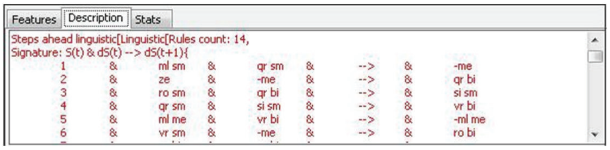
```
F-Transform Order
● Zero Order
○ First Order
Partition period
○ Data frequency: []
● Suggested (2T): 10
Frequency analysis
Detected periodicities:
7.2 16.8 20.1 25.2 33.6 50.8 74.1 80.6 88.1 101.3 118.1 126.3 135.7 144.3 168.4 210.5 280.1 369.4 420.4 470.4
Corresponding frequencies:
0.87 0.37 0.31 0.25 0.19 0.12 0.08 0.08 0.07 0.06 0.05 0.05 0.05 0.04 0.04 0.03 0.02 0.02 0.01 0.01
Predictor type
☑ Simple
☑ More steps
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	934	527	-297	-552	-272	-68	0	17	42	51	51	68	68	76	85	93	127
2	17	968	544	-289	-518	-263	-34	25	25	34	42	51	68	93	85	85	85
3	-68	8	968	544	-289	-518	-263	-34	25	25	34	42	51	68	93	85	85
4	-263	-68	8	943	501	-340	-527	-263	-68	8	34	51	51	93	85	93	85

```
Predictor selection
● Best SMAPE, valid. error: 1.15817
○ BEST RMSE
○ Suggested
Time:   216s of 192s (est.)     Predictors:   finished
```

```
Inverse FT[Component Predictor: Steps ahead linguistic[Linguistic[Rules count: 142,
Signature: d2S(t) & d2S(t-1) & d2S(t-2) --> S(t+1){
        1      &      -ve sm    &     -ex sm    &     -ml sm    &     -->    &     about 28.451404
        2      &      ex sm     &     ex sm     &     -ex sm    &     -->    &     about -85.787237
        3      &      ex sm     &     ex sm     &     ex sm     &     -->    &     about -38.922113
        4      &      ex sm     &     -ex sm    &     -ex sm    &     -->    &     about -91.588169
```

Figure 12: LFLF application.



```
Steps ahead linguistic[Linguistic[Rules count: 14,
Signature: S(t) & dS(t) --> dS(t+1){
        1      &      ml sm     &     qr sm     &     -->    &     -me
        2      &      ze        &     -me       &     -->    &     qr bi
        3      &      ro sm     &     qr bi     &     -->    &     si sm
        4      &      qr sm     &     si sm     &     -->    &     vr bi
        5      &      ml me     &     vr bi     &     -->    &     -ml me
        6      &      vr sm     &     -me       &     -->    &     ro bi
```

Figure 13: Winning predictor linguistic description (trend-cycle model).

4.2. Time Series Classification and Prediction via Linguistic Fuzzy Logic Forecaster. We tried also to utilize above presented method of time series analysis through linguistic fuzzy logic forecaster (LFLF) [15]; see Figure 12.

Basic usage of the application is to analyse given time series and find best predictor with respect to validation part of time series given. We evaluate efficiency of predictors by *SMAPE* (symmetric mean absolute percentage error). It enables us to make analysis of trend-cycle of a time series and also seasonal part. The main advantage lies in prediction based on transparent linguistic descriptions that provide the model of a time series behaviour. Linguistic variables are of the following types:

(i) value: we directly mean the components of the fuzzy transform,

(ii) difference, first order differences of fuzzy transform components that are given as follows: differences between components $\Delta X_i = X_i - X_{i-1}$,

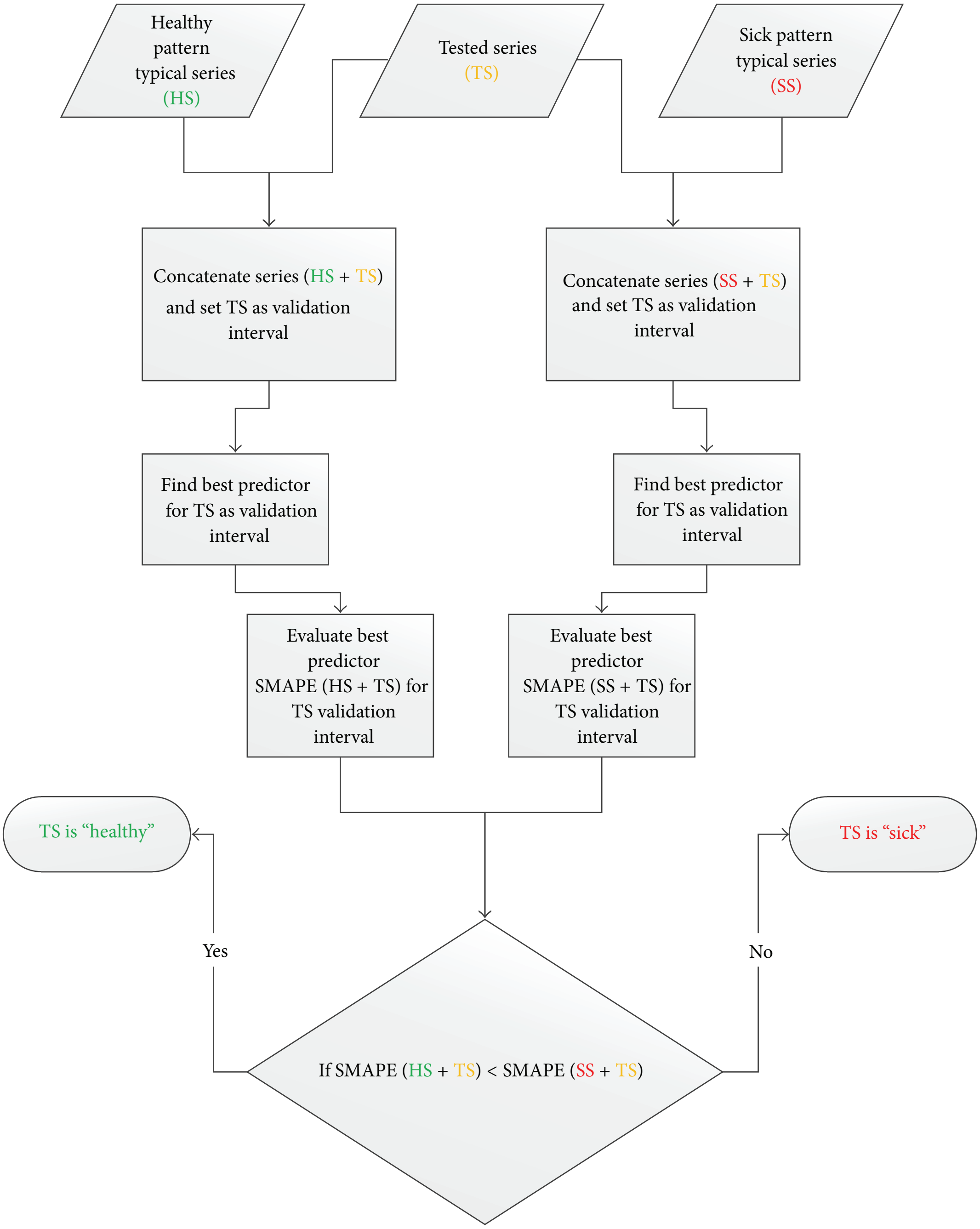


FIGURE 14: Recognition by linguistic fuzzy logic predictors for typical learning series.

(iii) second difference: these are values of second order differences of components of the fuzzy transform as follows: $\Delta^2 X_i = \Delta X_i - \Delta X_{i-1}$.

LFLF application enables us to define minimal and maximal number of these particular variables in a rule of linguistic description as well as the total number of antecedent variables.

A rule consisting of these variables has the following structure and can be described as a signature (fuzzy rules describing the trend-cycle model). Particularly, S denotes the trend-cycle components, dS their differences, and $d2S$ their second order differences. The argument (t), $(t-1)$, and so forth, denotes the time lag of the component.

For example, taking signature $S(t)\&dS(t) \rightarrow dS(t+1)$ denotes the fact that X_i and ΔX_i are the antecedent variables and ΔX_{i+1} is the consequent variable of the winning model and hence, we deal with rules of the form

$$\text{IF } X_i \text{ is } A_i \text{ AND } \Delta X_i \text{ is } A_{\Delta i} \text{ THEN } \Delta X_{i+1} \text{ is } A_{\text{Delta}i}. \tag{11}$$

Table 2: Example of algorithm evaluation on 10 “healthy” and 10 “sick” patients.

Patient	SMAPE (HS + TS)	SMAPE (SS + TS)	Result	Actual	Match
H11	3.38232	3.30989	Sick	Healthy	**NO**
H12	1.33555	3.40168	Healthy	Healthy	YES
H13	2.73377	4.58243	Healthy	Healthy	YES
H14	3.44581	2.37995	Sick	Healthy	**NO**
H15	2.04677	2.30998	Healthy	Healthy	YES
H16	3.73377	3.98572	Healthy	Healthy	YES
H17	1.03658	2.20151	Healthy	Healthy	YES
H18	3.22689	2.38111	Sick	Healthy	**NO**
H19	2.43544	3.42159	Healthy	Healthy	YES
H20	2.63355	3.79940	Healthy	Healthy	YES
S11	1.72265	1.31922	Sick	Sick	YES
S12	2.04804	0.38562	Sick	Sick	YES
S13	2.99464	0.51305	Sick	Sick	YES
S14	2.52248	0.67941	Sick	Sick	YES
S15	2.65972	1.75233	Sick	Sick	YES
S16	2.63674	2.39694	Sick	Sick	YES
S17	2.57638	1.65941	Sick	Sick	YES
S18	4.24496	2.85006	Sick	Sick	YES
S19	2.96533	1.01009	Sick	Sick	YES
S20	3.23630	1.10960	Sick	Sick	YES

Every single fuzzy rule can be taken as a sentence of natural language, for example, first rule from Figure 13.

IF X_i is ml sm AND ΔX_i is qr sm, THEN ΔX_{i+1} is −me may be read as follows.

If the number of cars sold in the current year is more or less small and the half-year sales increment is quite roughly small then the upcoming half-year increment will be negative medium.

4.2.1. Recognition of “Healthy” and “Sick” Patterns by LFLF. Our method to use linguistic fuzzy logic forecasting is based on simple idea that best predictor learning from both “healthy” and “sick” pattern samples, respectively, can be used for validation with tested pattern taken as validation part of the series. Then we can evaluate SMAPE for both these cases: compound series SMAPE (“healthy” + tested) and SMAPE (“sick” + tested).

If SMAPE (“healthy” + tested) < SMAPE (“sick” + tested) then the tested pattern is supposed to be “healthy”; otherwise, the tested pattern is supposed to be “sick.”

The idea is schematically shown in Figure 14.

For testing purposes, we created two necessary typical learning time series: “healthy” (HS) and “sick” (SS) according to the algorithm above. They both consist of 1010 samples made from 10 typical series of “healthy” and “sick” patients with 101 measured ECG values. Then we have created 10 concatenated series according to the scheme in Figure 14 with 10 randomly selected patients with “healthy” ECG measurement; that is, 20 files were produced (10x HS + TS and 10x SS + TS). The same concatenated series were also made from 10 “sick” patients measurements. This made us additional 20 files with concatenated series (10x HS + TS and 10x SS + TS). For 20 patients (Table 2) tested ECG we have 2 concatenated series giving SMAPE (HS + TS) and SMAPE (SS + TS).

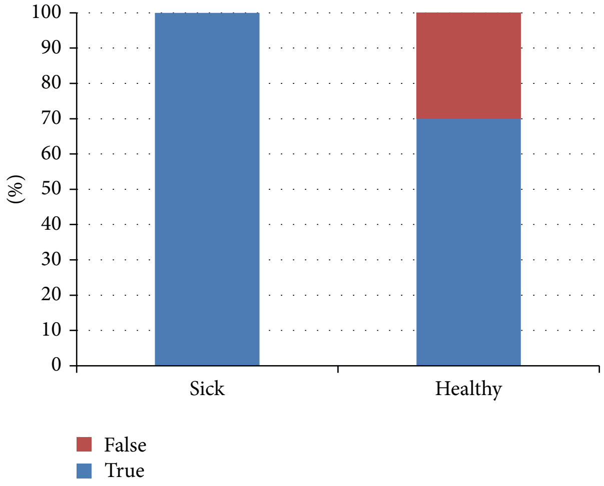


Figure 15: Experimental results, LFLF.

Our method based on LFLF proved very good results for right identification of sick patient records. Nevertheless, it produces large amount of false positive identification of sick pattern for healthy patients (Figure 15). This result is consistent with our approach using neural networks. Of course, our preliminary research has a limited extent and should be perceived only as narrative result, which shows interesting properties especially in complementation of neural network results.

5. Conclusion

In this paper, a short introduction into the field of ECG waves recognition using backpropagation neural network has been given. Main objective was to recognise the normal cycles and arrhythmias and perform further diagnosis. We proposed two detection systems that have been created with usage of neural networks. One of them is adapted according to the training set. Here, each pattern represents the whole one ECG cycle. Then, an output unit represents a diagnose 0/1, a healthy/sick person. The second one approach uses neural network, in which training set contains two different groups of patterns for healthy/sick persons. According to the results of experimental studies, it can be stated that ECG waves patterns were successfully extracted in given time series and recognised using suggested method, as can be seen from figures in Experimental Result section. It might result in better mapping of the time series behaviour for better prediction.

Both approaches were able to predict with high probability if the ECG time series represents sick or healthy persons. It is interesting that a sick diagnose was recognised with higher accuracy in both experimental works.

The third approach based on LFLF is currently only in the stage of preliminary experiments, but it conforms to the former results based on neural networks. This approach is novel and could be good supplement to other soft-computing methods for this task.

Disclaimer

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The research described here has been financially supported by University of Ostrava Grants SGS16/PřF/2014 and SGS14/PřF/2014.

References

[1] P. Tangkraingkij, C. Lursinsap, S. Sanguansintukul, and T. Desudchit, "Selecting relevant EEG signal locations for personal identification problem using ICA and neural network," in *Proceedings of the 8th IEEE/ACIS International Conference on Computer and Information Science (ICIS '09)*, pp. 616–621, June 2009.

[2] C. D. Nugent, J. A. C. Webb, N. D. Black, and G. T. H. Wright, "Electrocardiogram 2: classification," *Automedica*, vol. 17, pp. 281–306, 1999.

[3] S. Russell and P. Norvig, *Artificial Intelligence—A Modern Approach*, Prentice Hall, 2nd edition, 2003.

[4] C. D. Nugent, J. A. Lopez, A. E. Smith, and N. D. Black, "Reverse engineering of neural network classifiers in medical applications," in *Proceedings of the 7th EFOMP Congress*, vol. 17, p. 184, Physica Medica, 2001.

[5] L. Fausett, *Fundamentals of Neural Network*, Prentice Hall, 1994.

[6] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[7] V. Novák, *Fuzzy Sets and Their Applications*, Adam Hilger, Bristol, UK, 1989.

[8] S. Aouaouda, M. Chadli, P. Shi, and H. R. Karimi, "Discrete-time H_-/H_∞ sensor fault detection observer design for nonlinear systems with parameter uncertainty," *International Journal of Robust and Nonlinear Control*, 2013.

[9] M. Chadli, A. Abdo, and S. X. Ding, "H_-/H_∞ fault detection filter design for discrete-time Takagi-Sugeno fuzzy system," *Automatica*, vol. 49, no. 7, pp. 1996–2005, 2013.

[10] M. Chadli and H. R. Karimi, "Robust observer design for unknown inputs takagi-sugeno models," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 1, pp. 158–164, 2013.

[11] S. Aouaouda, M. Chadli, and H. Karimi, "Robust static output-feedback controller design against sensor failure for vehicle dynamics," *IET Control Theory & Applications*, vol. 8, no. 9, pp. 728–737, 2014.

[12] I. Perfilieva, V. Novák, V. Pavliska, A. Dvořák, and M. Štěpnička, "Analysis and prediction of time series using fuzzy transform," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '08)*, pp. 3876–3880, Hong Kong, June 2008.

[13] E. Volna, M. Kotyrba, and R. Jarusek, "Multi-classifier based on Elliott wave's recognition," *Computers & Mathematics with Applications*, vol. 66, no. 2, pp. 213–225, 2013.

[14] E. Volná, M. Kotyrba, and R. Jarušek, "Prediction by means of Elliott waves recognition," in *Nostradamus: Modern Methods of Prediction, Modeling and Analysis of Nonlinear Systems*, vol. 192 of *Advances in Intelligent Systems and Computing*, pp. 241–250, Springer, Berlin, Germany, 2013.

[15] H. Habiballa, V. Pavliska, and A. Dvořák, "Software system for time series prediction based on F-transform and linguistic rules," in *Proceedings of the 8th International Conference on Applied Mathematics Aplimat*, pp. 381–386, Bratislava, Slovakia, 2009.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 152485, 9 pages
http://dx.doi.org/10.1155/2014/152485

Research Article

Generalized Synchronization with Uncertain Parameters of Nonlinear Dynamic System via Adaptive Control

Cheng-Hsiung Yang and Cheng-Lin Wu

Graduate Institute of Automation and Control, National Taiwan University of Science and Technology, No. 43, Section 4, Keelung Road, Taipei 106, Taiwan

Correspondence should be addressed to Cheng-Hsiung Yang; chyang123123@mail.ntust.edu.tw

Received 19 June 2014; Revised 18 August 2014; Accepted 18 August 2014; Published 11 September 2014

Academic Editor: M. Chadli

An adaptive control scheme is developed to study the generalized adaptive chaos synchronization with uncertain chaotic parameters behavior between two identical chaotic dynamic systems. This generalized adaptive chaos synchronization controller is designed based on Lyapunov stability theory and an analytic expression of the adaptive controller with its update laws of uncertain chaotic parameters is shown. The generalized adaptive synchronization with uncertain parameters between two identical new Lorenz-Stenflo systems is taken as three examples to show the effectiveness of the proposed method. The numerical simulations are shown to verify the results.

1. Introduction

The chaos synchronization phenomenon has the following feature: the trajectories of the master and the slave chaotic system are identical in spite of starting from different initial conditions or different nonlinear dynamic system. However, slight differentiations of initial conditions, for chaotic dynamical systems, will lead to completely different trajectories [1–14]. The issue may be treated as the control law design for observer of slave chaotic system using the master chaotic system so as to ensure that the controlled receiver synchronizes with the master chaotic system. Hence, the slave chaotic system completely traces the dynamics of the master chaotic system in the course of time [15–19]. The key technique of chaos synchronization for secret communication has been widely investigated. Until now, a wide variety of approaches have been proposed for control and synchronization of chaotic systems, such as adaptive control [20, 21], backstepping control [22–25], sliding mode control [26–28], and fuzzy control [29–31], just to name a few. The forenamed strategies and many other existing skills of synchronization mainly concern the chaos synchronization of two identical chaotic systems with known parameters or identical unknown parameters [32–38].

Among many kinds of chaos synchronizations, the generalized synchronization is widely studied. This means that there exists a given functional relationship between the states of the master system and that of the slave system $y = f(x)$. In this paper, a new generalized synchronization with uncertain parameters,

$$\begin{aligned} \dot{x} &= f(t, x, A(t)), \\ y &= x + F(t), \end{aligned} \tag{1}$$

is studied, where x, y are the state vectors of the master and slave system, respectively, and the $A(t)$ is uncertain chaotic parameters in f. The $F(t)$ may be given a regular/chaotic dynamical system.

The rest of the paper is organized as follows. In Section 2, by the Lyapunov asymptotic stability theorem, the generalized synchronization with uncertain chaotic parameters by adaptive control scheme is given. In Section 3, various adaptive controllers and update laws are designed for the generalized synchronization with uncertain parameters of the identical Lorenz-Stenflo systems. The numerical simulation of three examples is also given in Section 3. Finally, some concluding remarks are given in Section 4.

2. Generalized Adaptive Synchronization with Uncertain Parameters Scheme

Consider the master system

$$\dot{x} = f\left(t, x, A\left(t\right)\right) \tag{2}$$

and the slave system

$$\dot{y} = f\left(t, y, \widehat{A}\left(t\right)\right) + u, \tag{3}$$

where $x = [x_1, x_2, \ldots, x_n]^T \in R^n$, $y = [y_1, y_2, \ldots, y_n]^T \in R^n$ denote the master states vector and slave states vector, respectively, the f is nonlinear vector functions, the $A(t)$ is uncertain chaotic parameters in f, the $\widehat{A}(t)$ is estimates of uncertain chaotic parameters in f, and the $u = [u_1, u_2, \ldots, u_n]^T \in R^n$ is adaptive control vector.

Our goal is to design a controller $u(t)$ and an adaptive law $\dot{\widetilde{A}}$ so that the state vector of the slave system equation (3) asymptotically approaches the state vector of the master system equation (2) plus a given vector regular/chaotic function $F(t) = [F_1(t), F_2(t), \ldots, F_n(t)]^T$, and finally the generalized adaptive synchronization with uncertain parameters will be accomplished in the sense that the limit of the states error vector $e(t) = [e_1, e_2, \ldots, e_n]^T$ and parameters error vector $\widetilde{A}(t) = [\widetilde{A}_1, \widetilde{A}_2, \ldots, \widetilde{A}_m]^T$ approaches zero:

$$\lim_{t\to\infty} e\left(t\right) = 0, \tag{4a}$$

$$\lim_{t\to\infty} \widetilde{A}\left(t\right) = 0, \tag{4b}$$

where $e_i = x_i - y_i + F_i(t)$, $(i = 1, 2, \ldots, n)$ and $\widetilde{A}_j = A_j(t) - \widehat{A}_j(t)$, $(j = 1, 2, \ldots, m)$.

From (4a), we have

$$\dot{e}_i = \dot{x}_i - \dot{y}_i + \dot{F}_i\left(t\right), \quad i = 1, 2, \ldots, n. \tag{5}$$

Introduce (2) and (3) in (5) as

$$\dot{e} = f\left(t, x, A\left(t\right)\right) - f\left(t, y, \widehat{A}\left(t\right)\right) + \dot{F}\left(t\right) - u\left(t\right). \tag{6}$$

A Lyapunov function candidate $V(e, \widetilde{A})$ is chosen as a positive definite function as

$$V\left(e, \widetilde{A}\right) = \frac{1}{2}e^T e + \frac{1}{2}\widetilde{A}^T\widetilde{A}. \tag{7}$$

Its derivative along the solution of (7) is

$$\begin{aligned}\dot{V}\left(e, \widetilde{A}\right) = e^T\left[f\left(t, x, A\left(t\right)\right) - f\left(t, y, \widehat{A}\left(t\right)\right) + \dot{F}\left(t\right) - u\left(t\right)\right] \\ + \widetilde{A}^T\dot{\widetilde{A}},\end{aligned} \tag{8}$$

where $u(t)$ and $\dot{\widetilde{A}}$ are chosen so that $\dot{V} = e^T C_1 e + \widetilde{A}^T C_2 \widetilde{A}$, C_1 and C_2 are negative constants, and $\dot{V}$ is a negative definite function of $e_1, e_2, \ldots, e_n$ and $\widetilde{A}_1, \widetilde{A}_2, \ldots, \widetilde{A}_m$. When

$$\lim_{t\to\infty} e = 0, \qquad \lim_{t\to\infty} \widetilde{A} = 0, \tag{9}$$

the generalized adaptive synchronization with uncertain parameters is obtained.

3. Results of Numerical Simulation

In this section, a mathematical proof is provided for the three cases' results of numerical, adaptive synchronization, generalized adaptive synchronization, and generalized adaptive synchronization with uncertain parameters.

3.1. Case I Adaptive Synchronization. The master system is new Lorenz-Stenflo system [39]:

$$\begin{aligned}\dot{x}_1 &= a\left(x_2 - x_1\right), \\ \dot{x}_2 &= cx_1 - x_1x_3 - x_2, \\ \dot{x}_3 &= bx_4 - dx_3 + x_1x_2, \\ \dot{x}_4 &= -x_1 - ax_4,\end{aligned} \tag{10}$$

where $a = 3.7$, $b = 1.5$, $c = 26$, and $d = 0.7$. The initial conditions are $x_1(0) = 30$, $x_2(0) = 30$, $x_3(0) = 30$, and $x_4(0) = 30$. The phase portrait is shown in Figure 1.

The slave system is

$$\begin{aligned}\dot{y}_1 &= \widehat{a}y_2 - \widehat{a}y_1 + u_1, \\ \dot{y}_2 &= \widehat{c}y_1 - y_1y_3 - y_2 + u_2, \\ \dot{y}_3 &= y_1y_2 - \widehat{d}y_3 + \widehat{b}y_4 + u_3, \\ \dot{y}_4 &= -y_1 - \widehat{a}y_4 + u_4,\end{aligned} \tag{11}$$

where $\widehat{a}$, $\widehat{b}$, $\widehat{c}$, $\widehat{d}$, and $\widehat{r}$ are estimates of uncertain parameters a, b, c, d, and r, respectively. The initial conditions of salve system are $y_1(0) = -50$, $y_2(0) = -50$, $y_3(0) = -50$, and $y_4(0) = -50$.

Our objective is to design the controllers such that the trajectories, $x(t)$ and $y(t)$, of the master system and slave system satisfy

$$\lim_{t\to\infty} \left\|x\left(t\right) - y\left(t\right)\right\| = 0. \tag{12a}$$

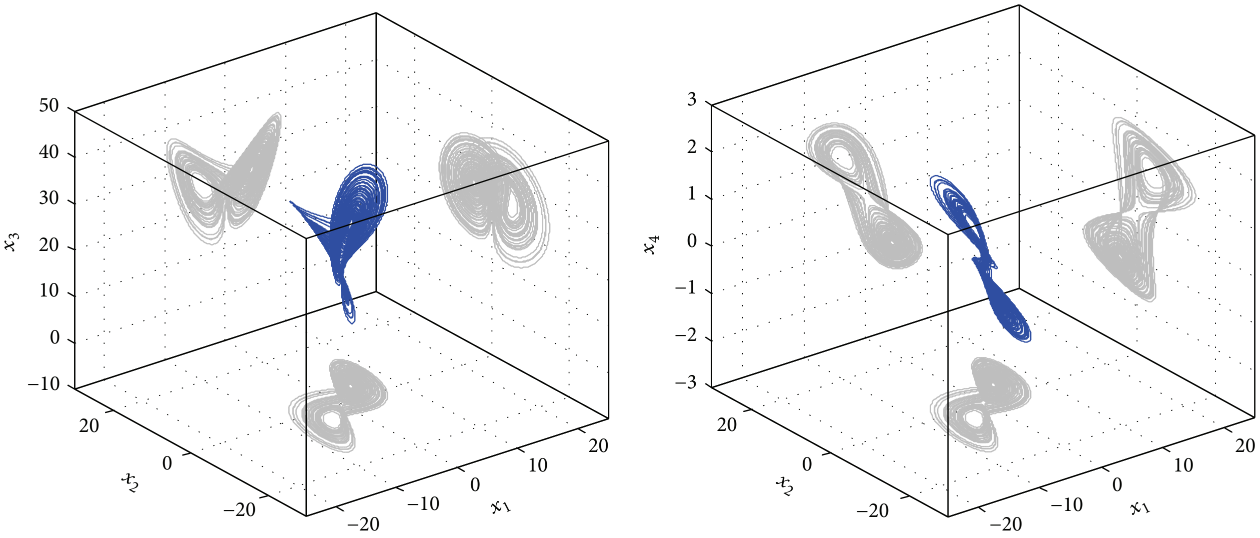
Our objective is to design the controllers parameters estimation update laws $\dot{\widetilde{A}} = [\dot{\widetilde{a}}, \dot{\widetilde{b}}, \dot{\widetilde{c}}, \dot{\widetilde{d}}]^T$ such that the trajectories, $A(t)$ and $\widehat{A}(t)$, of the uncertain chaotic parameters and estimates of uncertain chaotic parameters satisfy

$$\lim_{t\to\infty} \left\|A\left(t\right) - \widehat{A}\left(t\right)\right\| = 0, \tag{12b}$$

where $\|\cdot\|$ denotes the Euclidean norm.

Define an error vector function

$$e_i = x_i - y_i, \quad \left(i = 1, 2, 3, 4\right). \tag{13}$$



(a)

(b)

FIGURE 1: Three-dimension phase portrait of the four-dimensional Lorenz-Stenflo system and its projection.

From the error functions, we get the error dynamics

$$\begin{aligned}
\dot{e}_1 &= \hat{a}e_2 + \tilde{a}x_2 - \hat{a}e_1 - \tilde{a}x_1 - u_1,\\
\dot{e}_2 &= \hat{c}e_1 + \tilde{c}x_1 + e_1e_3 - x_1e_3 - x_3e_1 - e_2 - u_2,\\
\dot{e}_3 &= -e_1e_2 + x_1e_2 + x_2e_1 - \hat{d}e_3 - \tilde{d}x_3 + \hat{b}e_4 + \tilde{b}x_4 - u_3,\\
\dot{e}_4 &= -e_1 - \hat{a}e_4 - \tilde{a}x_4 - u_4,
\end{aligned} \tag{14}$$

where $e_1 = x_1 - y_1$, $e_2 = x_2 - y_2$, $e_3 = x_3 - y_3$, $e_4 = x_4 - y_4$, $\tilde{a} = a - \hat{a}$, $\tilde{b} = b - \hat{b}$, $\tilde{c} = c - \hat{c}$, and $\tilde{d} = d - \hat{d}$.

Choose a Lyapunov function candidate in the form of a positive definite function

$$\begin{aligned}
&V\left(e_1, e_2, e_3, e_4, \tilde{a}, \tilde{b}, \tilde{c}, \tilde{d}\right)\\
&\quad = \frac{1}{2}\left(e_1^2 + e_2^2 + e_3^2 + e_4^2 + \tilde{a}^2 + \tilde{b}^2 + \tilde{c}^2 + \tilde{d}^2\right),
\end{aligned} \tag{15}$$

and its time derivative is

$$\begin{aligned}
&\dot{V}\left(e_1, e_2, e_3, e_4, \tilde{a}, \tilde{b}, \tilde{c}, \tilde{d}\right)\\
&\quad = e_1\dot{e}_1 + e_2\dot{e}_2 + e_3\dot{e}_3 + e_4\dot{e}_4 + \tilde{a}\dot{\tilde{a}} + \tilde{b}\dot{\tilde{b}} + \tilde{c}\dot{\tilde{c}} + \tilde{d}\dot{\tilde{d}}\\
&\quad = e_1\left(\hat{a}e_2 + \tilde{a}x_2 - \hat{a}e_1 - \tilde{a}x_1 - u_1\right)\\
&\qquad + e_2\left(\hat{c}e_1 + \tilde{c}x_1 + e_1e_3 - x_1e_3 - x_3e_1 - e_2 - u_2\right)\\
&\qquad + e_3\left(-e_1e_2 + x_1e_2 + x_2e_1 - \hat{d}e_3 - \tilde{d}x_3\right.\\
&\qquad\qquad \left. + \hat{b}e_4 + \tilde{b}x_4 - u_3\right)\\
&\qquad + e_4\left(-e_1 - \hat{a}e_4 - \tilde{a}x_4 - u_4\right) + \tilde{a}\left(-\dot{\hat{a}}\right) + \tilde{b}\left(-\dot{\hat{b}}\right)\\
&\qquad + \tilde{c}\left(-\dot{\hat{c}}\right) + \tilde{d}\left(-\dot{\hat{d}}\right) + \tilde{r}\left(-\dot{\hat{r}}\right).
\end{aligned} \tag{16}$$

Choose the parameters estimation update laws as follows:

$$\begin{aligned}
\dot{\tilde{a}} &= -\dot{\hat{a}} = -e_1\left(x_2 - x_1\right) + e_4x_4 - \tilde{a},\\
\dot{\tilde{b}} &= -\dot{\hat{b}} = -e_3x_4 - \tilde{b},\\
\dot{\tilde{c}} &= -\dot{\hat{c}} = -e_2x_1 - \tilde{c},\\
\dot{\tilde{d}} &= -\dot{\hat{d}} = e_3x_3 - \tilde{d}.
\end{aligned} \tag{17}$$

The initial values of estimates for uncertain parameters are $\hat{a}(0) = 0$, $\hat{b}(0) = 0$, $\hat{c}(0) = 0$, and $\hat{d}(0) = 0$. Through (16) and (17), the appropriate controllers can be designed as

$$\begin{aligned}
u_1 &= \hat{a}e_2 - \hat{a}e_1 + e_1,\\
u_2 &= \hat{c}e_1 + e_1e_3 - x_1e_3 - x_3e_1 - e_2 + e_2,\\
u_3 &= -e_1e_2 + x_1e_2 + x_2e_1 - \hat{d}e_3 + \hat{b}e_4 + e_3,\\
u_4 &= -e_1 - \hat{a}e_4 + e_4.
\end{aligned} \tag{18}$$

Substituting (18) and (17) into (16), we obtain

$$\begin{aligned}
&\dot{V}\left(e_1, e_2, e_3, e_4, \tilde{a}, \tilde{b}, \tilde{c}, \tilde{d}\right)\\
&\quad = -{e_1}^2 - {e_2}^2 - {e_3}^2 - {e_4}^2 - \tilde{a}^2 - \tilde{b}^2 - \tilde{c}^2 - \tilde{d}^2 < 0.
\end{aligned} \tag{19}$$

Since the Lyapunov function $V(e_1, e_2, e_3, e_4, \tilde{a}, \tilde{b}, \tilde{c}, \tilde{d})$ is positive definite and its derivative $\dot{V}(e_1, e_2, e_3, e_4, \tilde{a}, \tilde{b}, \tilde{c}, \tilde{d})$ is

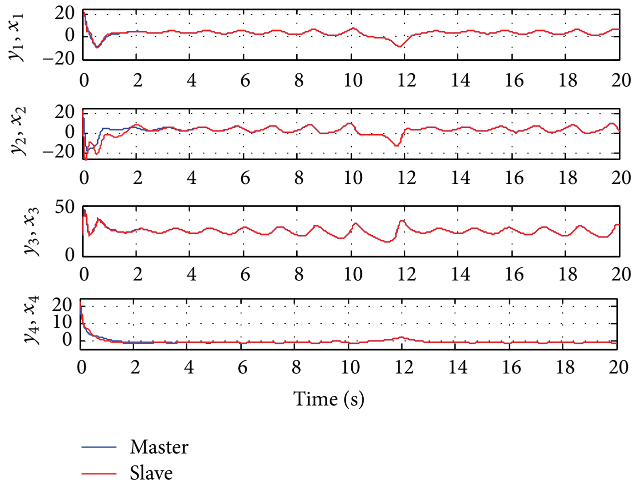


FIGURE 2: Time histories of the master system and slave system for Case I.

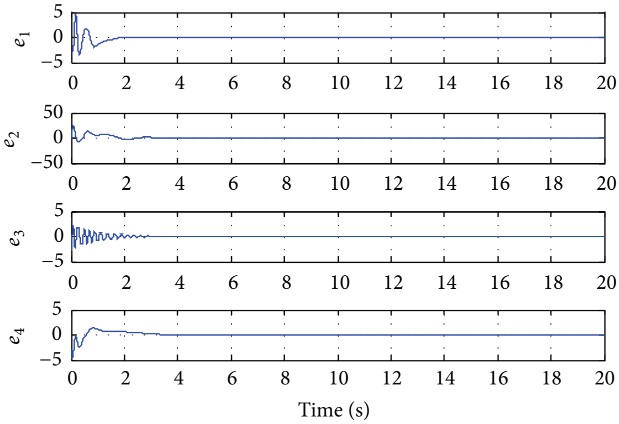


FIGURE 3: Time histories of error states for Case I.

negative definite in the neighborhood of the zero solutions for (12a) and (12b), according to the Lyapunov stability theory, the zero solutions of error states dynamic and parameters error vector are asymptotically stable; namely, the slave system equation (11) can asymptotically converge to its master system equation (10) with the adaptive control law equation (18) and the estimation parameter update law equation (17). The adaptive synchronization concept proof had to be completed. The numerical simulation results are shown in Figures 2, 3, and 4.

3.2. Case II Generalized Adaptive Synchronization. The given functional system for generalized synchronization is also a new Lorenz-Stenflo system but with different initial conditions: $w_1(0) = 25$, $w_2(0) = 25$, $w_3(0) = 25$, and $w_4(0) = 25$:

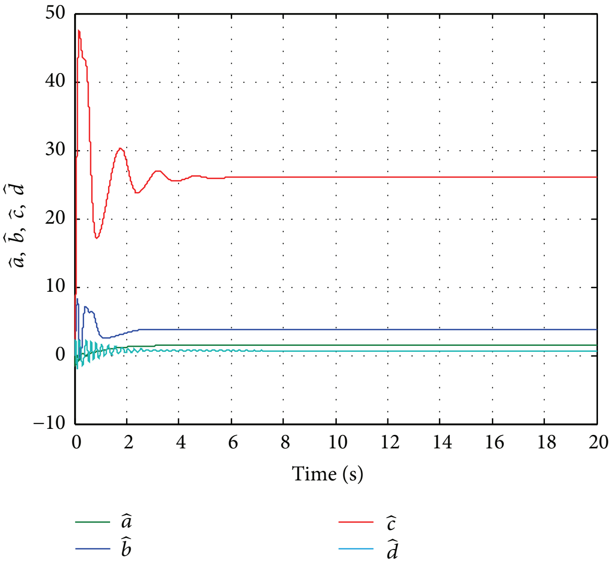


FIGURE 4: Time histories of estimated parameters for Case I.

$$\begin{aligned}
\dot{w}_1 &= a\left(w_2 - w_1\right), \\
\dot{w}_2 &= cw_1 - w_1 w_3 - w_2, \\
\dot{w}_3 &= bw_4 - dw_3 + w_1 w_2, \\
\dot{w}_4 &= -aw_4 - w_1 .
\end{aligned} \tag{20}$$

When the time approaches infinite, the error functions approach zero. The generalized adaptive synchronization can be accomplished as

$$\lim_{t \to \infty} e = 0, \tag{21}$$

where the error functions here can be defined as

$$e_i = x_i + w_i - y_i, \quad (i = 1, 2, 3, 4) . \tag{22}$$

From the error functions equation (22), we get the error dynamics

$$\begin{aligned}
\dot{e}_1 &= \hat{a}\left(e_2 - e_1\right) + \tilde{a}\left(x_2 - x_1 + w_2 - w_1\right) - u_1, \\
\dot{e}_2 &= \hat{c}e_1 + \tilde{c}\left(x_1 + w_1\right) - e_2 + x_1\left(w_3 - e_3\right) + w_1\left(x_3 - e_3\right) \\
&\quad - e_1\left(x_3 + w_3 - e_3\right) - u_2, \\
\dot{e}_3 &= -\hat{d}e_3 - \tilde{d}\left(x_3 + w_3\right) + \hat{b}e_4 + \tilde{b}\left(x_4 + w_4\right) \\
&\quad - x_1\left(w_2 - e_2\right) - w_1\left(x_2 - e_2\right) + \left(x_2 + w_2 - e_2\right) - u_3, \\
\dot{e}_4 &= -e_1 - \hat{a}e_4 - \tilde{a}\left(x_4 + w_4\right) - u_4,
\end{aligned} \tag{23}$$

where $\tilde{a} = a - \hat{a}$, $\tilde{b} = b - \hat{b}$, $\tilde{c} = c - \hat{c}$, and $\tilde{d} = d - \hat{d}$.

Choose a Lyapunov function in the form of a positive definite function

$$V\left(e_1, e_2, e_3, e_4, \tilde{a}, \tilde{b}, \tilde{c}, \tilde{d}\right) = \frac{1}{2}\left(e_1^2 + e_2^2 + e_3^2 + e_4^2 + \tilde{a}^2 + \tilde{b}^2 + \tilde{c}^2 + \tilde{d}^2\right), \tag{24}$$

and its time derivative is

$$\begin{aligned}
&\dot{V}\left(e_1, e_2, e_3, e_4, \tilde{a}, \tilde{b}, \tilde{c}, \tilde{d}\right) \\
&\quad = e_1\dot{e}_1 + e_2\dot{e}_2 + e_3\dot{e}_3 + e_4\dot{e}_4 + \tilde{a}\dot{\tilde{a}} + \tilde{b}\dot{\tilde{b}} + \tilde{c}\dot{\tilde{c}} + \tilde{d}\dot{\tilde{d}} \\
&\quad = e_1\left(\hat{a}\left(e_2 - e_1\right) + \tilde{a}\left(x_2 - x_1 + w_2 - w_1\right) - u_1\right) \\
&\qquad + e_2\left(\hat{c}e_1 + \tilde{c}\left(x_1 + w_1\right) - e_2 + x_1\left(w_3 - e_3\right)\right. \\
&\qquad\qquad \left. + w_1\left(x_3 - e_3\right) - e_1\left(x_3 + w_3 - e_3\right) - u_2\right) \\
&\qquad + e_3\left(-\hat{d}e_3 - \tilde{d}\left(x_3 + w_3\right) + \hat{b}e_4 + \tilde{b}\left(x_4 + w_4\right)\right. \\
&\qquad\qquad - x_1\left(w_2 - e_2\right) - w_1\left(x_2 - e_2\right) \\
&\qquad\qquad \left. + e_1\left(x_2 + w_2 - e_2\right) - u_3\right) \\
&\qquad + e_4\left(-e_1 - \hat{a}e_4 - \tilde{a}\left(x_4 + w_4\right) - u_4\right) + \tilde{a}\left(-\dot{\hat{a}}\right) + \tilde{b}\left(-\dot{\hat{b}}\right) \\
&\qquad + \tilde{c}\left(-\dot{\hat{c}}\right) + \tilde{d}\left(-\dot{\hat{d}}\right) + \tilde{r}\left(-\dot{\hat{r}}\right).
\end{aligned} \tag{25}$$

Choose the parameters estimation update laws as follows:

$$\begin{aligned}
\dot{\tilde{a}} &= -\dot{\hat{a}} = -e_1\left(x_2 - x_1 + w_2 - w_1\right) + e_4\left(x_4 + w_4\right) - \tilde{a}, \\
\dot{\tilde{b}} &= -\dot{\hat{b}} = -e_3\left(x_4 + w_4\right) - \tilde{b}, \\
\dot{\tilde{c}} &= -\dot{\hat{c}} = -e_2\left(x_1 + w_1\right) - \tilde{c}, \\
\dot{\tilde{d}} &= -\dot{\hat{d}} = e_3\left(x_3 + w_3\right) - \tilde{d}.
\end{aligned} \tag{26}$$

The initial values of estimates for uncertain parameters are $\hat{a}(0) = 0$, $\hat{b}(0) = 0$, $\hat{c}(0) = 0$, and $\hat{d}(0) = 0$. Through (25) and (26), the appropriate controllers can be designed as

$$\begin{aligned}
u_1 &= \hat{a}e_2 - \hat{a}e_1 + e_1, \\
u_2 &= \hat{c}e_1 - e_2 + x_1\left(w_3 - e_3\right) + w_1\left(x_3 - e_3\right) \\
&\quad - e_1\left(x_3 + w_3 - e_3\right) + e_2, \\
u_3 &= -\hat{d}e_3 + \hat{b}e_4 - x_1\left(w_2 - e_2\right) - w_1\left(x_2 - w_2\right) \\
&\quad + e_1\left(x_2 + w_2 - e_2\right) + e_3, \\
u_4 &= -e_1 - \hat{a}e_4 + e_4.
\end{aligned} \tag{27}$$

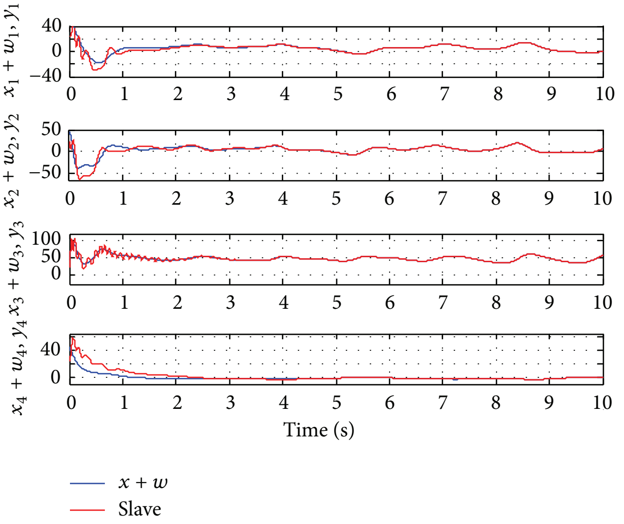


FIGURE 5: Time histories of the generalized signal (master x system plus given function w) and slave system for Case II.

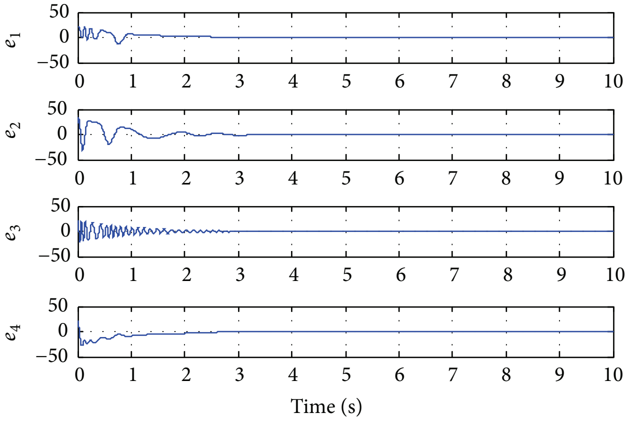


FIGURE 6: Time histories of error states for Case II.

Substituting (26) and (27) into (25), we obtain

$$\dot{V}\left(e_1, e_2, e_3, e_4, \tilde{a}, \tilde{b}, \tilde{c}, \tilde{d}\right) = -{e_1}^2 - {e_2}^2 - {e_3}^2 - {e_4}^2 - \tilde{a}^2 - \tilde{b}^2 - \tilde{c}^2 - \tilde{d}^2 < 0. \tag{28}$$

Since the Lyapunov function $V(e_1, e_2, e_3, e_4, \tilde{a}, \tilde{b}, \tilde{c}, \tilde{d})$ is positive definite and its derivative $\dot{V}(e_1, e_2, e_3, e_4, \tilde{a}, \tilde{b}, \tilde{c}, \tilde{d})$ is negative definite in the neighborhood of the zero solutions for (12a) and (12b), according to the Lyapunov stability theory, the zero solutions of error states dynamic and parameters error vector are asymptotically stable; namely, the slave system equation (11) can asymptotically converge to its master system equation (10) with the adaptive control law equation (27) and the estimation parameter update law equation (26). The generalized adaptive synchronization concept proof had to be completed. The numerical simulation results are shown in Figures 5, 6, and 7.

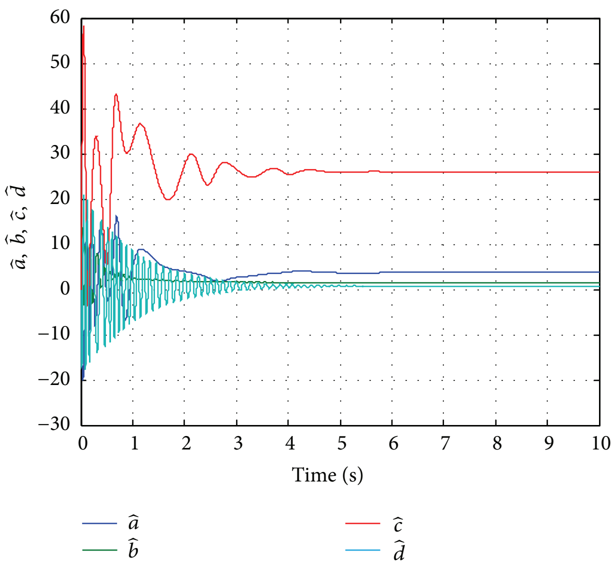


Figure 7: Time histories of estimated parameters for Case II.

3.3. Case III Generalized Adaptive Synchronization with Uncertain Parameters. Consider that the master system is the new Lorenz-Stenflo system with uncertain chaotic parameters

$$
\begin{aligned}
\dot{x}_1 &= A_1(t)(x_2 - x_1), \\
\dot{x}_2 &= A_3(t)x_1 - x_1x_3 - x_2, \\
\dot{x}_3 &= A_2(t)x_3 - A_4(t)x_3 + x_1x_2, \\
\dot{x}_4 &= -A_1(t)x_4 - x_1,
\end{aligned}
\tag{29}
$$

where $A_1(t)$, $A_2(t)$, $A_3(t)$, and $A_4(t)$ are uncertain chaotic parameters. The uncertain parameters are given as

$$
\begin{aligned}
A_1(t) &= a(1 + f_1z_1), \\
A_2(t) &= b(1 + f_2z_2), \\
A_3(t) &= c(1 + f_3z_3), \\
A_4(t) &= d(1 + f_4z_4),
\end{aligned}
\tag{30}
$$

where f_1, f_2, f_3, and f_4 are arbitrary positive constants. Positive constants are $f_1 = f_2 = f_3 = f_4 = 0.005$. The chaotic signals z_1, z_2, z_3, and z_4 are given as the states of system as follows:

$$
\begin{aligned}
\dot{z}_1 &= a(z_2 - z_1), \\
\dot{z}_2 &= cz_1 - z_1z_3 - z_2, \\
\dot{z}_3 &= bz_4 + -dz_3 + z_1z_2, \\
\dot{z}_4 &= -az_4 - z_1.
\end{aligned}
\tag{31}
$$

The initial constants of the chaotic signals are $z_1(0) = 0.7$, $z_2(0) = 0.7$, $z_3(0) = 0.7$, and $z_4(0) = 0.7$. The new Lorenz-Stenflo system with uncertain chaotic parameters of master system will exhibit a more complex dynamic behavior since the parameters of the system change over time.

The generalized synchronization error functions can be defined as

$$
e_i = x_i + w_i - y_i, \quad (i = 1, 2, 3, 4). \tag{32}
$$

From the error functions equation (32), the error dynamics becomes

$$
\begin{aligned}
\dot{e}_1 &= \widehat{A}_1(e_2 - e_1 - w_2 + w_1) + \widetilde{A}_1(x_2 - x_1) \\
&\quad + a(w_2 - w_1) - u_1, \\
\dot{e}_2 &= \widehat{A}_3(e_1 - w_1) + \widetilde{A}_3x_1 + cw_1 + x_1(w_3 - e_3) \\
&\quad + w_1(x_3 - e_3) - e_1(x_3 + w_3 - e_3) - e_2 - u_2, \\
\dot{e}_3 &= \widehat{A}_2(e_4 - w_4) + \widetilde{A}_2x_4 + bw_4 - \widehat{A}_4(e_3 - w_3) - \widetilde{A}_4x_3 \\
&\quad - dw_3 - x_1(w_2 - e_2) - w_1(x_2 - e_2) \\
&\quad + e_1(x_2 + w_2 - e_2) - u_3, \\
\dot{e}_4 &= -e_1 - \widehat{A}_1(e_4 - w_4) - \widetilde{A}_1x_4 - aw_4 - u_4,
\end{aligned}
\tag{33}
$$

where $\widetilde{A}_i(t) = A_i(t) - \widehat{A}_i(t)$ $(i = 1, 2, 3, 4)$.

Choose a Lyapunov function in the form of a positive definite function:

$$
\begin{aligned}
&V\left(e_1, e_2, e_3, e_4, \widetilde{A}_1, \widetilde{A}_2, \widetilde{A}_3, \widetilde{A}_4\right) \\
&\quad = \frac{1}{2}\left(e_1^2 + e_2^2 + e_3^2 + e_4^2 + \widetilde{A}_1^2(t) + \widetilde{A}_2^2(t)\right. \\
&\qquad \left. + \widetilde{A}_3^2(t) + \widetilde{A}_4^2(t)\right),
\end{aligned}
\tag{34}
$$

and its time derivative is

$$
\begin{aligned}
&\dot{V}\left(e_1, e_2, e_3, e_4, \widetilde{A}_1, \widetilde{A}_2, \widetilde{A}_3, \widetilde{A}_4\right) \\
&\quad = e_1\dot{e}_1 + e_2\dot{e}_2 + e_3\dot{e}_3 + e_4\dot{e}_4 + \widetilde{A}_1\dot{\widetilde{A}}_1 + \widetilde{A}_2\dot{\widetilde{A}}_2 \\
&\qquad + \widetilde{A}_3\dot{\widetilde{A}}_3 + \widetilde{A}_4\dot{\widetilde{A}}_4 \\
&\quad = e_1\left(\widehat{A}_1(e_2 - e_1 - w_2 + w_1) + \widetilde{A}_1(x_2 - x_1)\right. \\
&\qquad\qquad \left. + a(w_2 - w_1) - u_1\right) \\
&\qquad + e_2\left(\widehat{A}_3(e_1 - w_1) + \widetilde{A}_3x_1 + cw_1 + x_1(w_3 - e_3)\right. \\
&\qquad\qquad \left. + w_1(x_3 - e_3) - e_1(x_3 + w_3 - e_3) - e_2 - u_2\right) \\
&\qquad + e_3\left(\widehat{A}_2(e_4 - w_4) + \widetilde{A}_2x_4 + bw_4 - \widehat{A}_4(e_3 - w_3)\right. \\
&\qquad\qquad - \widetilde{A}_4x_3 - dw_3 - x_1(w_2 - e_2) - w_1(x_2 - e_2) \\
&\qquad\qquad \left. + e_1(x_2 + w_2 - e_2) - u_3\right)
\end{aligned}
$$

$$\begin{aligned}&+ e_4\left(-e_1 - \widehat{A}_1\left(e_4 - w_4\right) - \widetilde{A}_1 x_4 - a w_4 - u_4\right)\\&+ \widetilde{A}_1\left(-\dot{\widehat{A}}_1\right) + \widetilde{A}_2\left(-\dot{\widehat{A}}_2\right) + \widetilde{A}_3\left(-\dot{\widehat{A}}_3\right) + \widetilde{A}_4\left(-\dot{\widehat{A}}_4\right).\end{aligned} \tag{35}$$

Choose the parameters estimation update laws for those uncertain parameters as follows:

$$\begin{aligned}\dot{\widetilde{A}}_1 &= \dot{A}_1 - \dot{\widehat{A}}_1 = -e_1\left(x_2 - x_1\right) + e_4 x_4 - \widetilde{A}_1,\\ \dot{\widetilde{A}}_2 &= \dot{A}_2 - \dot{\widehat{A}}_2 = -e_3 x_4 - \widetilde{A}_2,\\ \dot{\widetilde{A}}_3 &= \dot{A}_3 - \dot{\widehat{A}}_3 = -e_2 x_1 - \widetilde{A}_3,\\ \dot{\widetilde{A}}_4 &= \dot{A}_4 - \dot{\widehat{A}}_4 = e_3 x_3 - \widetilde{A}_4.\end{aligned} \tag{36}$$

The initial values of estimates for uncertain parameters are $\widehat{A}_1(0) = 0$, $\widehat{A}_2(0) = 0$, $\widehat{A}_3(0) = 0$, and $\widehat{A}_4(0) = 0$. Through (35) and (36), the appropriate controllers can be designed as

$$\begin{aligned}u_1 &= \widehat{A}_1\left(e_2 - e_1 - w_2 + w_1\right) + a\left(w_2 - w_1\right) + e_1,\\ u_2 &= \widehat{A}_3\left(e_1 - w_1\right) + c w_1 + x_1\left(w_3 - e_3\right) + w_1\left(x_3 - e_3\right)\\ &\quad - e_1\left(x_3 + w_3 - e_3\right) - e_2 + e_2,\\ u_3 &= \widehat{A}_2\left(e_4 - w_4\right) + b w_4 - \widehat{A}_4\left(e_3 - w_3\right) - d w_3\\ &\quad - x_1\left(w_2 - e_2\right) - w_1\left(x_2 - w_2\right)\\ &\quad + e_1\left(x_2 + w_2 - e_2\right) + e_3,\\ u_4 &= -\widehat{A}_1\left(e_4 - w_4\right) - a w_4 - e_1 + e_4.\end{aligned} \tag{37}$$

Substituting (36) and (37) into (35), we obtain

$$\begin{aligned}&\dot{V}\left(e_1, e_2, e_3, e_4, \widetilde{A}_1, \widetilde{A}_2, \widetilde{A}_3, \widetilde{A}_4\right)\\&\quad = -e_1^2 - e_2^2 - e_3^2 - e_4^2 - \widetilde{A}_1^2(t) - \widetilde{A}_2^2(t) - \widetilde{A}_3^2(t) - \widetilde{A}_4^2(t)\\&\quad < 0.\end{aligned} \tag{38}$$

Since the Lyapunov function $V(e_1, e_2, e_3, e_4, \widetilde{a}, \widetilde{b}, \widetilde{c}, \widetilde{d})$ is positive definite and its derivative $\dot{V}(e_1, e_2, e_3, e_4, \widetilde{a}, \widetilde{b}, \widetilde{c}, \widetilde{d})$ is negative definite in the neighborhood of the zero solutions for (12a) and (12b), according to the Lyapunov stability theory, the zero solutions of error states dynamic and parameters error vector are asymptotically stable; namely, the slave system equation (11) can asymptotically converge to its master system equation (10) with the adaptive control law equation (36) and the estimation parameter update law equation (37). The generalized adaptive synchronization with uncertain parameters concept proof had to be completed. The numerical simulation results are shown in Figures 8, 9, 10, and 11.

4. Conclusion

A generalized adaptive synchronization with uncertain chaotic parameters is new chaos synchronization concept.

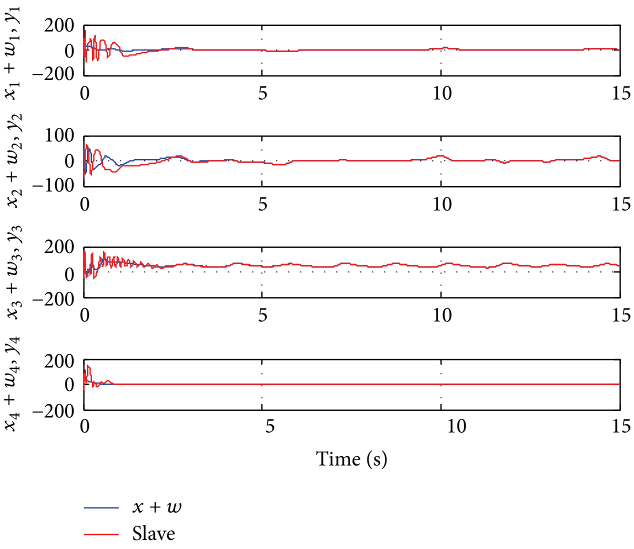


FIGURE 8: Time histories of the generalized signal (master x system plus given function w) and slave system for Case III.

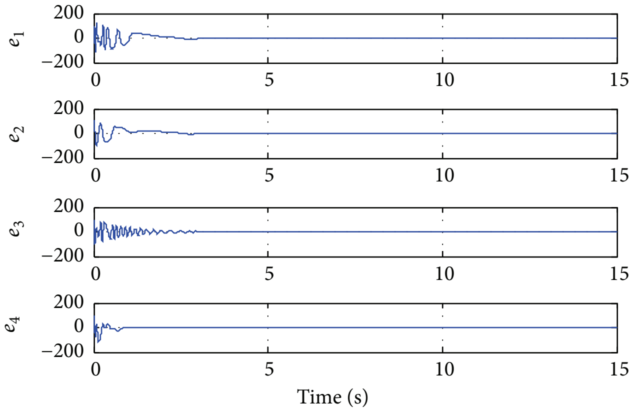


FIGURE 9: Time histories of error states for Case III.

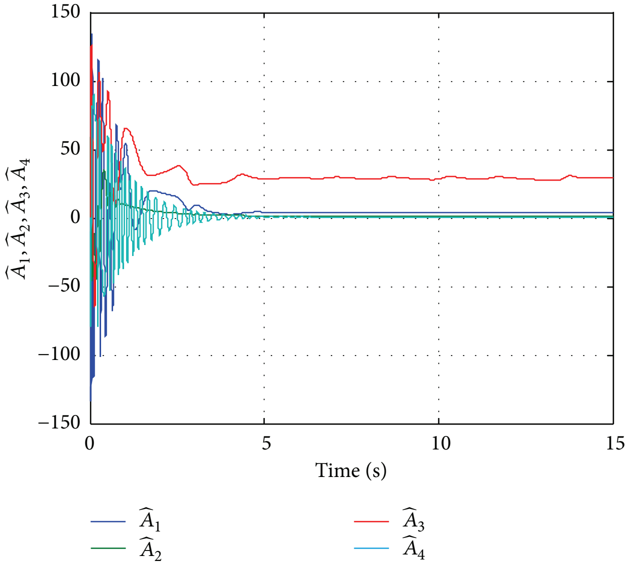
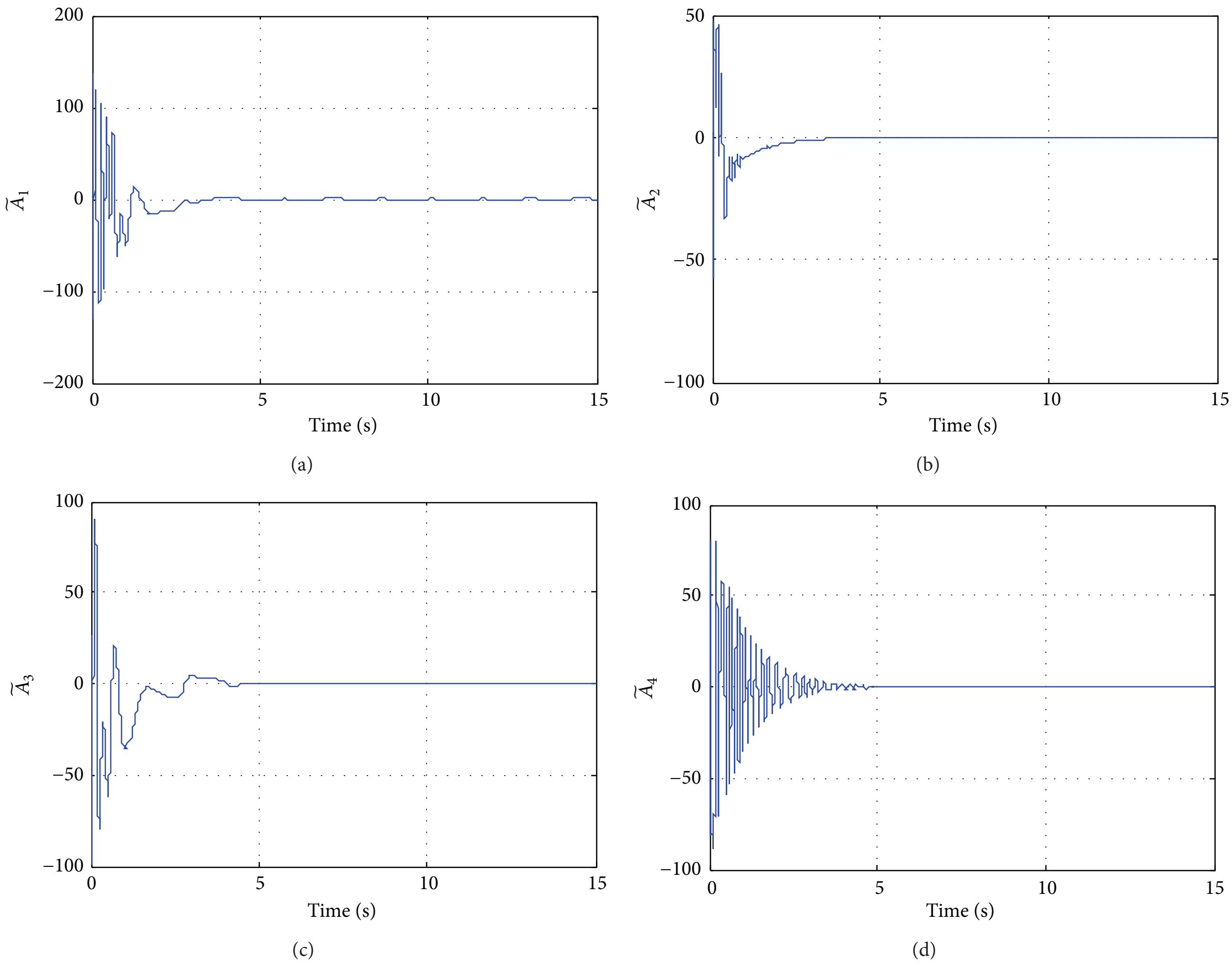


FIGURE 10: Time histories of estimated parameters for Case III.



(a) (b) (c) (d)

FIGURE 11: Time histories of different parameters for Case III.

The theoretical analysis and numerical simulation results of three cases, adaptive synchronization, generalized adaptive synchronization, and generalized adaptive synchronization with uncertain parameters, are shown in the corresponding figures which imply that the adaptive controllers and update laws we designed are feasible and effective. In this paper, the three examples can be used to increase the security of secret communication system.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This research was supported by the National Science Council, China, under Grant no. NSC 102-2221-E-011-034.

References

[1] L. M. Pecora and T. L. Carroll, "Synchronization in chaotic systems," *Physical Review Letters*, vol. 64, no. 8, pp. 821–824, 1990.

[2] J. R. Terry and G. D. Vanwiggeren, "Chaotic communication using generalized synchronization," *Chaos, Solitons and Fractals*, vol. 12, no. 1, pp. 145–152, 2001.

[3] W. Guo Xu and Q. Shu Li, "Chemical chaotic schemes derived from NSG system," *Chaos, Solitons and Fractals*, vol. 15, no. 4, pp. 663–671, 2003.

[4] Z.-M. Ge and C.-H. Yang, "Symplectic synchronization of different chaotic systems," *Chaos, Solitons and Fractals*, vol. 40, no. 5, pp. 2532–2543, 2009.

[5] Z.-M. Ge and C.-H. Yang, "The generalized synchronization of a Quantum-CNN chaotic oscillator with different order systems," *Chaos, Solitons and Fractals*, vol. 35, no. 5, pp. 980–990, 2008.

[6] Z.-M. Ge and C.-H. Yang, "Synchronization of complex chaotic systems in series expansion form," *Chaos, Solitons & Fractals*, vol. 34, no. 5, pp. 1649–1658, 2007.

[7] U. E. Vincent, "Synchronization of identical and non-identical 4-D chaotic systems using active control," *Chaos, Solitons and Fractals*, vol. 37, no. 4, pp. 1065–1075, 2008.

[8] L. Stenflo, "Generalized Lorenz equations for acoustic-gravity waves in the atmosphere," *Physica Scripta*, vol. 53, no. 1, pp. 83–84, 1996.

[9] C.-H. Yang, "Enhanced symplectic synchronization between two different complex chaotic systems with uncertain parameters," *Abstract and Applied Analysis*, vol. 2013, Article ID 193138, 12 pages, 2013.

[10] S.-Y. Li, C.-H. Yang, and C.-T. Lin, "Chaotic motions in the real fuzzy electronic circuits," *Abstract and Applied Analysis*, vol. 2013, Article ID 875965, 8 pages, 2013.

[11] C.-H. Yang, "Symplectic synchronization of Lorenz-Stenflo system with uncertain chaotic parameters via adaptive control," *Abstract and Applied Analysis*, vol. 2013, Article ID 528325, 14 pages, 2013.

[12] S.-Y. Li, C.-H. Yang, L.-W. Ko, C.-T. Lin, and Z.-M. Ge, "Implementation on electronic circuits and RTR pragmatical adaptive synchronization: time-reversed uncertain dynamical systems' analysis and applications," *Abstract and Applied Analysis*, vol. 2013, Article ID 909721, 10 pages, 2013.

[13] C.-H. Yang, "Chaos hybrid generalized synchronization of liu-chen system by GYC partial region stability theory," *Journal of Computational and Theoretical Nanoscience*, vol. 10, no. 4, pp. 825–831, 2013.

[14] C. H. Yang, P. C. Tsen, S. Y. Li, and Z. M. Ge, "Pragmatical adaptive synchronization by variable strength linear coupling," *Journal of Computational and Theoretical Nanoscience*, vol. 10, no. 4, pp. 1007–1013, 2013.

[15] C.-H. Yang, "Chaos control of the Quantum-CNN systems," *Journal of Computational and Theoretical Nanoscience*, vol. 10, no. 1, pp. 171–176, 2013.

[16] Z. Liu, "The first integrals of nonlinear acoustic gravity wave equations," *Physica Scripta*, vol. 61, no. 5, article 526, 2000.

[17] C. Zhou, C. H. Lai, and M. Y. Yu, "Bifurcation behavior of the generalized Lorenz equations at large rotation numbers," *Journal of Mathematical Physics*, vol. 38, no. 10, pp. 5225–5239, 1997.

[18] S. Banerjee, P. Saha, and A. R. Chowdhury, "Chaotic scenario in the Stenflo equations," *Physica Scripta*, vol. 63, no. 3, pp. 177–180, 2001.

[19] Y. Chen, X. Wu, and Z. Gui, "Global synchronization criteria for two Lorenz-Stenflo systems via single-variable substitution control," *Nonlinear Dynamics*, vol. 62, no. 1-2, pp. 361–369, 2010.

[20] L. Yang and J. Jiang, "Adaptive synchronization of drive-response fractional-order complex dynamical networks with uncertain parameters," *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 5, pp. 1496–1506, 2014.

[21] D. Zhu, C. Liu, and B. Yan, "Modeling and adaptive pinning synchronization control for a chaotic-motion motor in complex network," *Physics Letters A*, vol. 378, no. 5-6, pp. 514–518, 2014.

[22] W. Xiao-Qun and L. Jun-An, "Parameter identification and backstepping control of uncertain Lü system," *Chaos, Solitons and Fractals*, vol. 18, no. 4, pp. 721–729, 2003.

[23] Y. Yu and S. Zhang, "Controlling uncertain Lü system using backstepping design," *Chaos, Solitons and Fractals*, vol. 15, no. 5, pp. 897–902, 2003.

[24] J. A. Laoye, U. E. Vincent, and S. O. Kareem, "Chaos control of 4D chaotic systems using recursive backstepping nonlinear controller," *Chaos, Solitons and Fractals*, vol. 39, no. 1, pp. 356–362, 2009.

[25] M. T. Yassen, "Chaos control of chaotic dynamical systems using backstepping design," *Chaos, Solitons & Fractals*, vol. 27, no. 2, pp. 537–548, 2006.

[26] W. M. Bessa, A. S. de Paula, and M. A. Savi, "Chaos control using an adaptive fuzzy sliding mode controller with application to a nonlinear pendulum," *Chaos, Solitons and Fractals*, vol. 42, no. 2, pp. 784–791, 2009.

[27] J.-F. Chang, M.-L. Hung, Y.-S. Yang, T.-L. Liao, and J.-J. Yan, "Controlling chaos of the family of Rössler systems using sliding mode control," *Chaos, Solitons & Fractals*, vol. 37, no. 2, pp. 609–622, 2008.

[28] S. Zhankui and K. Sun, "Nonlinear and chaos control of a micro-electro-mechanical system by using second-order fast terminal sliding mode control," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 9, pp. 2540–2548, 2013.

[29] D. Chen, W. Zhao, J. C. Sprott, and X. Ma, "Application of Takagi-Sugeno fuzzy model to a class of chaotic synchronization and anti-synchronization," *Nonlinear Dynamics*, vol. 73, no. 3, pp. 1495–1505, 2013.

[30] Y. Nian and Y. Zheng, "Controlling discrete time T-S fuzzy chaotic systems via adaptive adjustment," *Physics Procedia*, vol. 24, pp. 1915–1921, 2012.

[31] H. Hua, Y. Liu, J. Lu, and J. Zhu, "A new impulsive synchronization criterion for T-S fuzzy model and its applications," *Applied Mathematical Modelling*, vol. 37, pp. 8826–8835, 2013.

[32] S. Yanchuka and T. Kapitaniak, "Chaos—hyperchaos transition in coupled Rössler systems," *Physics Letters A*, vol. 290, pp. 139–144, 2001.

[33] P. Perlikowski, A. Stefanski, and T. Kapitaniak, "Ragged synchronizability and clustering in a network of coupled oscillators," in *Advances in Nonlinear Dynamics and Synchronization*, pp. 49–75, 2009.

[34] M. Chadlia, A. Abdob, and S. X. Dingb, "$H_/H_\infty$ fault detection filter design for discrete-time Takagi–Sugeno fuzzy system," *Automatica*, vol. 49, no. 7, pp. 1996–2005, 2013.

[35] M. Chadli and H. R. Karimi, "Robust observer design for unknown inputs takagi-sugeno models," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 1, pp. 158–164, 2013.

[36] S. Aouaouda, M. Chadli, V. Cocquempot, and M. Tarek Khadir, "Multi-objective H_H_∞ fault detection observer design for Takagi-Sugeno fuzzy systems with unmeasurable premise variables: descriptor approach," *International Journal of Adaptive Control and Signal Processing*, vol. 27, no. 12, pp. 1031–1047, 2013.

[37] M. Chadlia, I. Zelinkab, and T. Youssef, "Unknown inputs observer design for fuzzy systems with application to chaotic system reconstruction," *Computers & Mathematics with Applications*, vol. 66, no. 2, pp. 147–154, 2013.

[38] I. Zelinka, M. Chadli, D. Davendra, R. Senkerik, and R. Jasek, "An investigation on evolutionary reconstruction of continuous chaotic systems," *Mathematical and Computer Modelling*, vol. 57, no. 1-2, pp. 2–15, 2013.

[39] C.-H. Yang and C.-L. Wu, "Nonlinear dynamic analysis and synchronization of four-dimensional Lorenz-Stenflo system and its circuit experimental implementation," *Abstract and Applied Analysis*, vol. 2014, Article ID 213694, 17 pages, 2014.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 435935, 10 pages
http://dx.doi.org/10.1155/2014/435935

Research Article

Surface Evaluation by Estimation of Fractal Dimension and Statistical Tools

Vlastimil Hotar[1] and Petr Salac[2]

[1] *Department of Glass Producing Machines and Robotics, Technical University of Liberec, Studentská 1402/2, 461 17 Liberec, Czech Republic*
[2] *Department of Mathematics and Didactics of Mathematics, Technical University of Liberec, Studentská 1402/2, 461 17 Liberec, Czech Republic*

Correspondence should be addressed to Vlastimil Hotar; vlastimil.hotar@tul.cz

Received 23 May 2014; Revised 4 July 2014; Accepted 17 July 2014; Published 27 August 2014

Academic Editor: M. Chadli

Structured and complex data can be found in many applications in research and development, and also in industrial practice. We developed a methodology for describing the structured data complexity and applied it in development and industrial practice. The methodology uses fractal dimension together with statistical tools and with software modification is able to analyse data in a form of sequence (signals, surface roughness), 2D images, and dividing lines. The methodology had not been tested for a relatively large collection of data. For this reason, samples with structured surfaces produced with different technologies and properties were measured and evaluated with many types of parameters. The paper intends to analyse data measured by a surface roughness tester. The methodology shown compares standard and nonstandard parameters, searches the optimal parameters for a complete analysis, and specifies the sensitivity to directionality of samples for these types of surfaces. The text presents application of fractal geometry (fractal dimension) for complex surface analysis in combination with standard roughness parameters (statistical tool).

1. Introduction

Due to continuously increasing pressure from competitors to improve the quality of products, there is a demand for objective measurement and control methods for materials, processes, and production processes. However, it is almost impossible to describe many structures using conventional methods (e.g., defects, surfaces, cracks, and time series from dynamic processes) because they are complex and irregular. One approach is the application of fractal dimension which is successfully used in science.

The fractal dimension is closely connected to fractals that were defined by Mandelbrot [1], though scientists found some geometric problems with specific objects (e.g., the measurement of coast lines using different lengths of rulers by Richardson). A potentially powerful property of the fractal dimension is the ability to describe complexity by using a single number that defines and quantifies structures [2, 3]. The number is mostly a noninteger value and the fractal dimension is higher than the topological dimension. For example, the Koch curve (one of the most famous mathematical deterministic fractals) has the topological dimension $D_T = 1$, but the fractal dimension $D_F = 1.2619$. A smooth curve as a line has the topological dimension $D_T = 1$ and the fractal dimension $D_F = 1$. The fractal dimension can be computed for a set of points, curves, surfaces, topological 3D objects, and so forth and if the fractal dimension is higher than the topological dimension, we name the objects fractals.

Fractal dimension is part of a wider theory, fractal geometry. Fractal geometry is closely connected to chaos theory. Furthermore, the obtained structures were produced by real dynamic systems and the obtained data was influenced by these dynamic systems [2, 4, 5]. The data can also be tested to chaotic properties (future work) and also simulated. Chaotic system can be identified by standard tools like Lyapunov coefficient, Hurst coefficient, and also fractal dimension. An alternative and promising way to identify chaotic system is evolutionary reconstruction [6]. Application of chaotic system reconstruction can be practically used for a chaotic cryptosystem procedure [7].

Table 1: List of analysed samples with their production properties.

Sample	Technology of surfaces production
1	Polished surface to maximum gloss
2	Ballotini (glass beads) blasting, grain size F120 (mean diameter 0.109 mm)
3	Corundum blasting, grain size F36 (mean diameter 0.525 mm)
4	Corundum blasting, grain size F12 (mean diameter 1.765 mm)
5	Electro-erosion machining 29A
6	Electro-erosion machining 42A
7	Electro-erosion machining 54A
8	Sandpaper, K400
9	Emery cloth, 120
10	Emery cloth, 80
11	Vertical milling machine, milling cutter 20 mm, 120 rpm, feed 30 mm/min
12	Grinding wheel, 98A 60J 9V C40
13	Grinding wheel, 96A 36P 5V
14	Vertical milling machine, milling cutter 20 mm, 120 rpm, feed 240 mm/min

Even though applications of fractal dimension in industry are quite rare and experimental [8], it is possible to find a promising test and applications [9–14]. Fractal dimension in conjunction with statistics can be used as a useful and powerful tool for an explicit, objective, and automatic description of production process data (laboratory, off-line, and potentially on-line). Fractal dimension does not substitute other tools like statistics and should be used with other parameters for complete analysis. Here, we carry out research into the mentioned tools on a methodology that uses standard and nonstandard parameters to evaluate complex data from industrial practice [15, 16] and laboratories [17–19]. The methodology finds suitable parameters for a complete analysis of the data from a set of parameters. Only the chosen parameters should be used in order to reduce processing time in industrial practice. The chosen parameters can also be recalculated to one number and the number can be used in quality assessment, for example, [16].

However, no research based on a deeper analysis of a relatively large data set has been conducted yet. For reliable usage of the methodology and analysis used, their properties and limitations have to be defined. We also wanted to analyse one source of data with different measured methods. The given theoretical results will be used for improvement of the methodology and finding relationships among the parameters and results of various tests. The main motivation is to answer whether parameters like fractal dimension are useful and beneficial for a complex description of the data from industrial practice.

For this purpose we analysed 14 surfaces produced by 5 different processes and in different conditions, Table 1. Figure 1 shows 28 samples (with 14 surfaces). The analysed structures were chosen so as to be different and to cover the most common surfaces in industrial practice. The chosen samples were made purposely from identical material. This allows us to subsequently ignore material properties and to analyse the change of technological parameters and the influence of technology used.

Figure 1: Analysed samples with machined surfaces.

The samples were measured using 3 methods: with a surface roughness tester, by image-capturing with an electron microscope, and by image-capturing of metallographic samples using an optical microscope, Figure 2. These three methods generate three data types that are the most common types in industrial practice (sequences, signals, 2D images, and dividing lines). The measurements were analysed using the developed methodology with 30 parameters. Results comparison of a surface roughness description, 2D images, and dividing lines seem to be interesting topics for future work.

This paper presents the first results of conducted research and it focuses on data from a surface roughness tester. Nine parameters were chosen for detailed analysis. Further measurements and comparison of the measurements will be published later.

The aims of this phase of the presented research are

(i) to compare standard and nonstandard parameters;

(ii) to find the optimal parameters for a complete analysis;

(iii) to specify the sensitivity to directionality of samples for these types of surfaces.

2. Methodology and Tools Used

The unfiltered reading (raw data) from a surface roughness tester is called a profile (curve). The profile can be evaluated using various methods. The parameters obtained can be divided into three groups, as follows:

(i) *parameters of amplitude,* useful for depth characterization (Std: standard deviation, R_a: average roughness, R_t: maximum roughness, R_z: mean roughness depth, *etc.);*

(ii) *parameters of frequency,* used to describe surface profile spacing parameters and for corrugation frequency characterization (e.g., S_m: mean spacing);

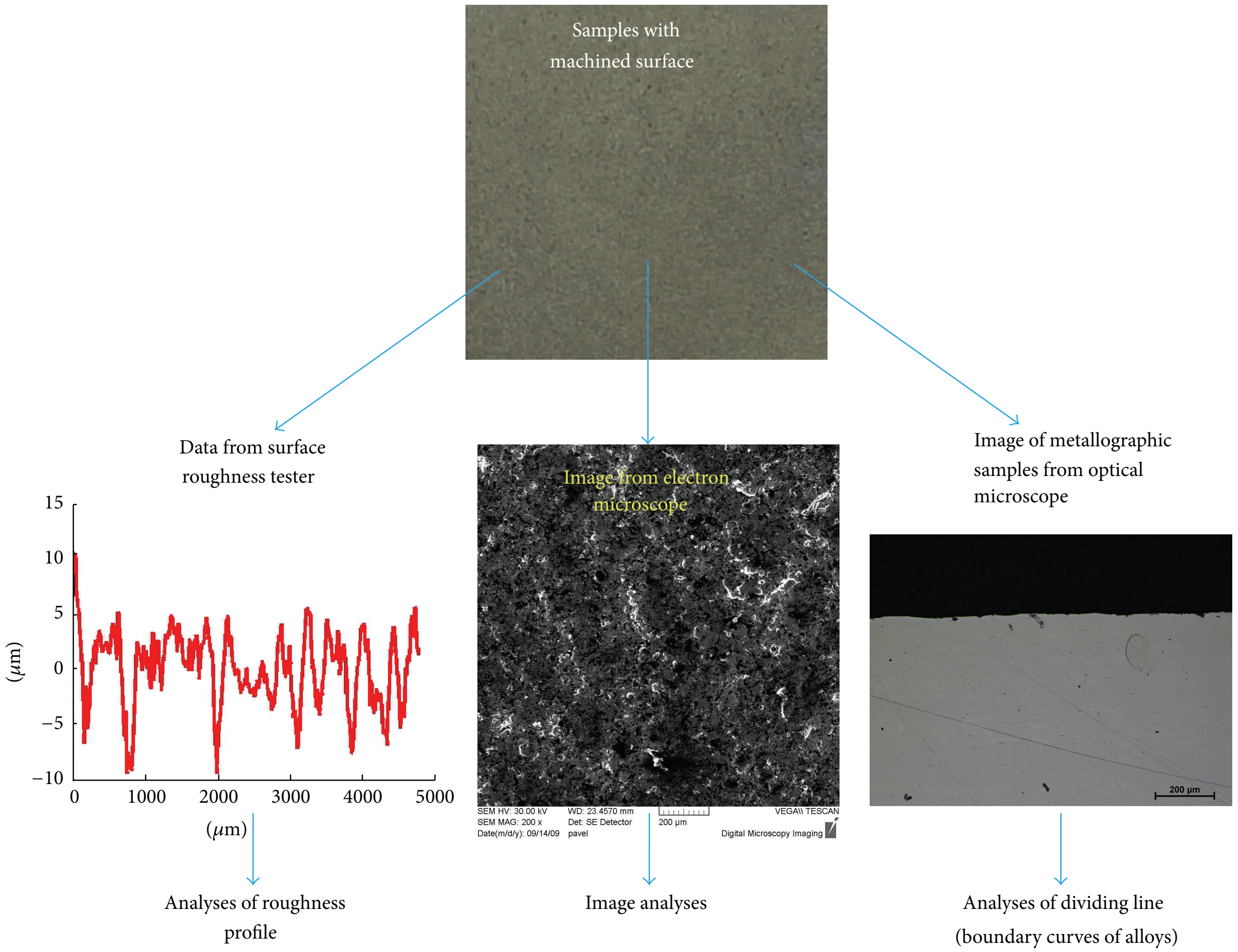


Figure 2: Measurement of samples, obtained data, and analyses.

(iii) *parameters of complexity and deformation,* estimation of fractal dimension by compass dimension (DC) [1–3], by EEE method [20], or by relative length (L_R) and proportional length (L_P) of the profile.

The mentioned parameters of amplitude and frequency are commonly used in industrial practice. These parameters are based on statistics. Average roughness, maximum roughness, mean roughness depth, and mean spacing are surface profile parameters defined by standard ISO 4287-1997 [21]. The parameters of complexity and deformation were selected based on previous experiences.

Average roughness (R_a) is also known as the arithmetical mean roughness. The Average roughness is the area between the roughness profile and its mean line or the integral of the absolute value of the roughness profile height over the evaluation length:

$$R_a = \frac{1}{l} \int_0^l |z(x)|\, dx, \tag{1}$$

where l is the evaluation length and z is the deviation from the center line m, Figure 3. When evaluated from digital data, the integral is normally approximated by a trapezoidal rule, as follows:

$$R_a = \frac{1}{n} \sum_{i=1}^{n} |z_i|, \tag{2}$$

where n is the number of measurements. Graphically, the average roughness is the area (yellow in Figure 3) between the roughness profile and its centre line m divided by the evaluation length. In this field of research, a filtered profile is not being used. For this reason the average roughness is called P_a.

Maximum roughness (R_t), also maximum height, or total roughness, is the vertical distance from the deepest trough to the highest peak, Figure 3. For the unfiltered profile, maximum roughness is denoted by P_t.

Mean roughness depth (R_{z5}) is the arithmetic mean of the single distance from the deepest trough to the highest peak from 5 sampling lengths (l_1–l_5), Figure 3. For the unfiltered profile, mean roughness depth is denoted byP_{z5}.

S_m is the mean spacing between peaks, now with a peak defined relative to the mean line. A peak must cross above the mean line and then cross back below it. If the width of each

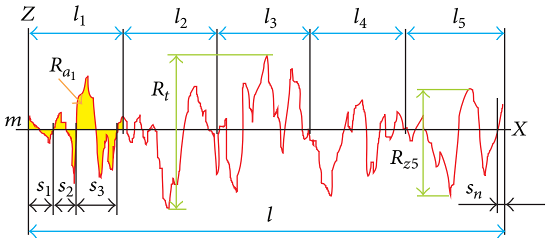


Figure 3: Parameters R_a, R_t, R_{z5}, S_m with the centre line m.

peak is denoted as S_i, then the mean spacing is the average width of a peak over the evaluation length, Figure 3:

$$S_m = \frac{1}{n} \sum_{i=1}^{n} S_i. \tag{3}$$

The estimated compass dimension expresses the degree of complexity of the profile by means of a single number [1]. A compass method [1–3] is based on measuring the profile (curve) using different ruler sizes (Figure 4(a)) according to

$$L_i(r_i) = N_i(r_i) \cdot r_i, \tag{4}$$

where L_i is the length in i-step of the measurement, r_i is the ruler size, and N_i is the number of steps needed for the measurement. If the profile is fractal, and hence the estimated fractal dimension is larger than the topological dimension, then the length measured increases as the ruler size is reduced. The logarithmic dependence between $\log_2 N(r_i)$ and $\log_2 r_i$ is called the Richardson-Mandelbrot plot (Figure 4(b)). The compass dimension is then determined from the slope s of the regression line, as follows:

$$D_C = 1 - s = 1 - \frac{\Delta \log_2 L(r)}{\Delta \log_2 r}. \tag{5}$$

For better comparison of the results, the dimension is multiplied by 1000 ($D_{C\,1000}$). The fractal dimension can also be estimated using a different method [2, 3].

The rate of profile deformation can be evaluated from its relative length L_R. This fast and reliable method measures the ratio of the profile length l_{PIXEL} (red curve in Figure 3) using the smallest ruler (1 pixel) r_{PIXEL} and the length of the projection l (Figure 3), as follws:

$$L_R = \frac{l_{\text{PIXEL}}}{l}. \tag{6}$$

Another similar approach is to compute the proportional length of the profile L_P. The proportional length is the ratio of the profile length measured with a defined ruler l_r (e.g., green line in Figure 4(a)) and the length measured with the maximum ruler $l_{r\,\max}$ (the length between the first and the last point of the profile):

$$L_P = \frac{l_r}{l_{r\,\max}}. \tag{7}$$

The EEE method (evaluation of length changes with elimination of insignificant extremes) [20] stems from an estimation of the fractal dimension, so it measures changes of lengths in sequential steps. The method does not use a fixed “ruler” for its measurement in every step, but the line is defined by local extremes (maxima and minima). The method is based on the length evaluation of a profile (curve or signal).

The profile is defined by measured values, which are isolated points $x_1, x_2, \ldots, x_n$ in the range $z(x_1), z(x_2), \ldots, z(x_n)$. The points represent local extremes (maxima and minima). On the profile, unnecessary extremes are classified with a defined ruler and a new simplified function is defined by the remaining points. A relative length L_{R1} of the new function is measured and the result is saved.

The procedure for the elimination of insignificant extremes is applied to the simplified function (profile). The function obtained is also measured and the process is reiterated. The last function is formed from the global maximum and minimum of all functions, at which point the analysis is stopped. The steps i of the analysis are plotted against the computed relative lengths L_{Ri} of the functions. The relation between the relative lengths L_{Ri} and the steps of elimination i is evaluated by a suitable regression function that can be a regression line, a quadratic function, or a hyperbolic function. In the case of using a regression line, the dimension can be computed from the slope s by the following equation:

$$D_{\text{EEE}} = 1 + |s|. \tag{8}$$

For better comparison of the results the dimension is multiplied by 1000 ($D_{\text{EEE}\,1000}$). More information can be found in [20].

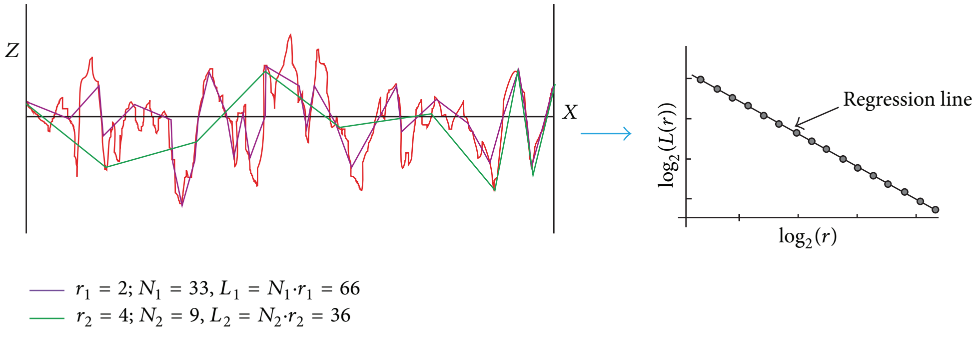
3. Measurement of Samples

The surface roughness tester Mitutoyo SV 2000 was used for taking measurements (parameters: traverse range: 50 mm; linearity of traverse: 0.3 μm/50 mm; stylus speed measuring: 0.5 mm/s; positioning: 2 mm/s). A standard type of stylus with a 60° angle was used with a measuring force: 0.75 mN.

All samples (2 samples with the same surface) were measured in 9 positions, each position in 3 directions, x, y, and transverse. The length of measurement is 4800 μm and the sampling interval is 0.5 μm. All data obtained is in the form of unfiltered profiles. A software tool for a data evaluation was developed in Matlab.

4. Results

The samples analysed have clearly different structural characters. In Figure 1, the samples are ordered from the smoothest to the most structured surface (from left to right). The two upper lines represent blasted and electroeroded surfaces (random surfaces) and the two bottom lines represent the classically machined surfaces. Graphs in Figures 5, 6, 7, 8, 9, 10, 11, 12, and 13 show the results of analysis for the surfaces from the measurement of the profiles in one direction (x-axis). A correlation between the chosen parameters is



(a)

(b)

FIGURE 4: Estimation of the fractal dimension by the compass method.

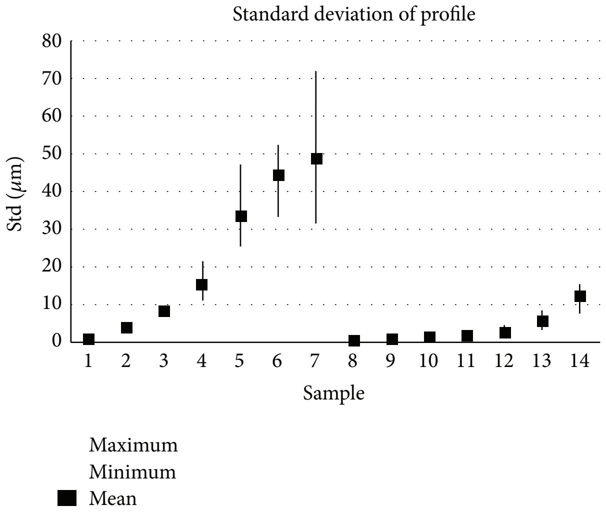


FIGURE 5: Results of parameter Standard deviation (x-axis).

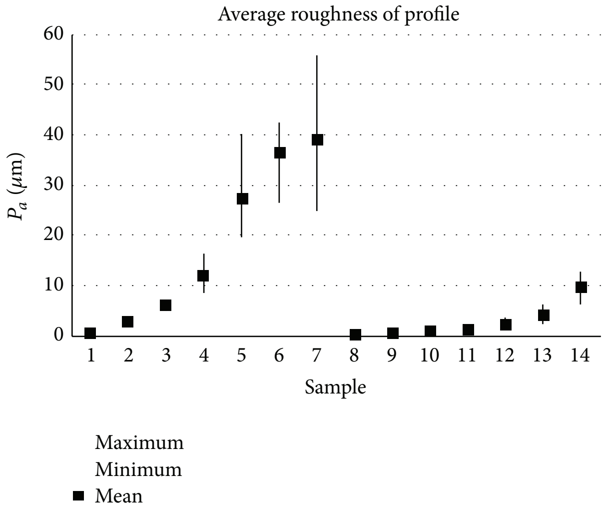


FIGURE 6: Results of P_a parameter (x-axis).

clearly visible (P_a, $D_{\mathrm{EEE}\,1000}$, and L_R). The $D_{C\,1000}$ parameter correlates lower and the S_m parameter does not correlate. To evaluate the parameters objectively, Pearson's correlation coefficients were computed, see Table 2 (the parameters are normally distributed). The aim is to specify the appropriate parameters for fast and reliable analysis for industrial data evaluation [15] (e.g., production control or quality monitoring). Only the chosen parameters should be used for a complete analysis of the data in order to reduce processing time. Some parameters linearly correlate with others (they provide similar information about the data), Table 2. If the situation is simplified and a linear correlation is assumed, we can specify suitable parameters for evaluation of these types of data as follows: average roughness, P_a (*parameter of amplitude*), Mean Spacing, S_m (*parameter of frequency*), compass dimension, and $D_{C\,1000}$ (*parameter of complexity and deformation*). These 3 parameters provide diverse information about the data.

A decisive number (a testing number, a quality number) is required in several applications. Typically, during a subjective testing by an operator (mostly by human eyes), one tested number is obtained, based on subjective comparison with etalons [15, 16]. The demand for only one testing number for quality evaluation comes from industrial practice. Three parameters that fully describe the data can be used for objective evaluation. In these cases the single number has to be calculated from the 3 parameters by weight coefficients and can be converted to a specified quality scale. The weight coefficients for each of the three parameters have to be specified using an appropriate methodology.

Measurements were taken at 9 different measurement points in the x, y, and transverse direction for each of the 28 samples examined. This was done for all 9 presented methods.

The mean values of the data obtained from individual samples of x-axis directions (μ_1), y-axis directions (μ_2), and the transverse directions (μ_3) were compared for each sample. Conformity of the mean values was tested by one-way analysis of variance (ANOVA) [22] at significance level $\alpha = 0.05$ using Matlab software. Thus,

$$\begin{aligned} &H_0 : \mu_1 = \mu_2 = \mu_3 \\ &H_1 : \text{non } H_0. \end{aligned} \tag{9}$$

Table 2: Correlation coefficients of parameters (x-axis).

	L_P proportional length, [—]	L_R relative length, [—]	$D_{\text{EEE}\,1000}$ EEE dimension, [—]	$D_{C\,1000}$ compass dimension, [—]	S_m mean spacing, [μm]	P_{z5} mean roughness depth, [μm]	P_t maximum roughness, [μm]	P_a Average roughness, [μm]	Std standard deviation, [μm]
Standard deviation, Std [μm]	0.92	0.92	0.90	0.74	0.37	0.98	0.99	1.00*	1.00
Average roughness, P_a [μm]	0.92	0.92	0.89	0.74	0.37	0.98	0.98	1.00	
Maximum roughness, P_t [μm]	0.93	0.93	0.92	0.79	0.34	0.99	1.00		
Mean roughness Depth, P_{z5} [μm]	0.95	0.95	0.94	0.81	0.33	1.00			
Mean spacing, S_m [μm]	0.25	0.25	0.22	0.15	1.00				
Compass dimension, $D_{C\,1000}$ [—]	0.85	0.85	0.94	1.00					
EEE dimension $D_{\text{EEE}\,1000}$ [—]	0.96	0.96	1.00						
Relative length, L_R [—]	1.00*	1.00							
Proportional length, L_P [—]	1.00								

*The correlation coefficient is rounded up to 1.00, but is not equal to 1.

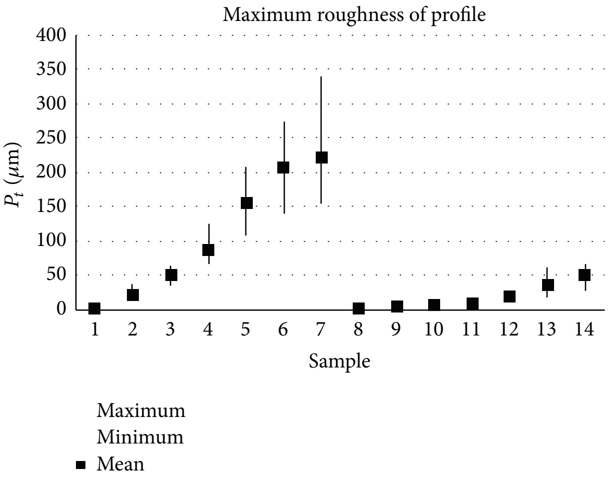


FIGURE 7: Results of P_t parameter (x-axis).

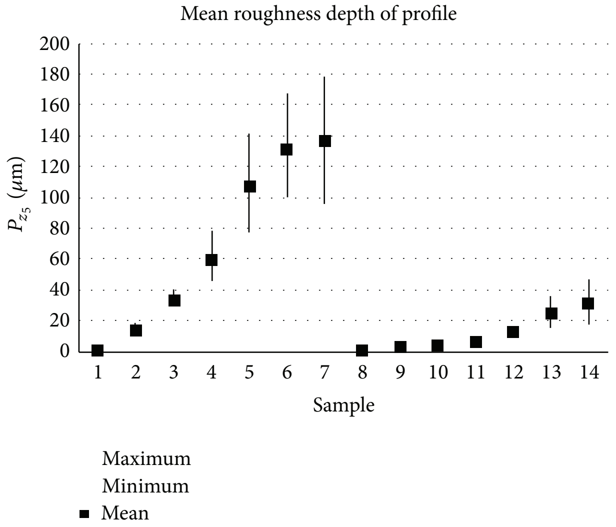


FIGURE 8: Results of P_{z5} parameter (x-axis).

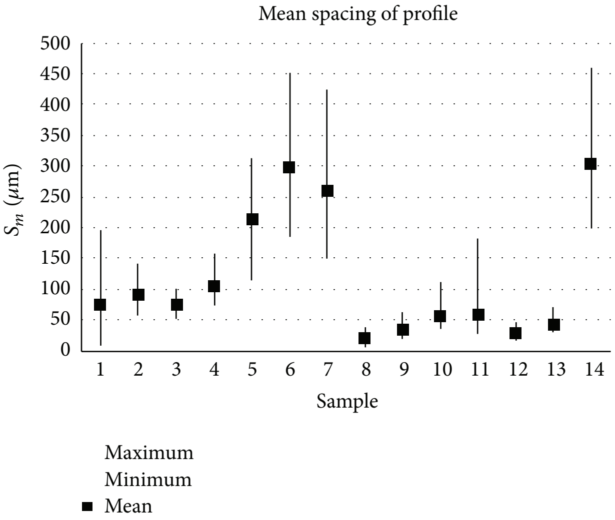


FIGURE 9: Results of S_m parameter (x-axis).

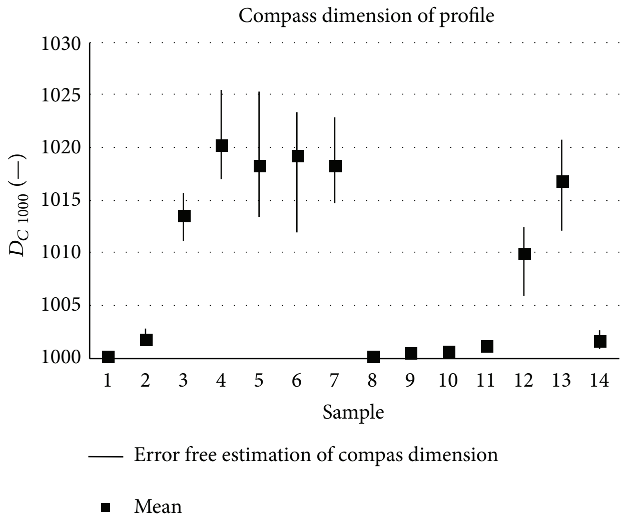


FIGURE 10: Results of fractal dimension estimation, $D_{C\,1000}$ (x-axis).

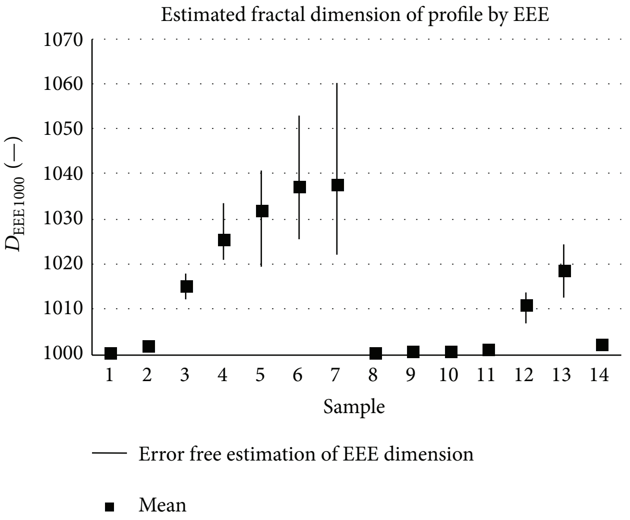


FIGURE 11: Results of fractal dimension estimation, $D_{EEE\,1000}$ (x-axis).

The test results are shown in Table 3, where value 0 means a rejection H_0 and a benefit H_1 (results of measurement are dependent according to direction). Value 1 does not constitute rejection H_0 (results are independent on the direction). P values for rejection of the hypothesis H_0 in favor of the alternatives H_1 are also shown in Table 3.

Samples 1 to 7 were prepared by technologies that produce random structures. Samples 8 to 14 were produced by a standard machining method that generates directionally visible structures (Figure 1). Samples 8, 9, 10, 12, and 13 have linearly oriented structures. Samples 11 and 14 have rotationally oriented structures, because of the milling technology. Parameters $D_{C\,1000}$, $D_{\mathrm{EEE}\,1000}$, L_R, and L_P (parameters of complexity and deformation) show good results in recognition of the directionality. The only exception was for samples 11. These samples have a smooth rotationally

Table 3: One-way analysis of variance (ANOVA).

	Standard deviation, Std [mm]		Average roughness, P_a [mm]		Maximum roughness, P_t[mm]		Mean roughness depth, P_{z5} [mm]		Mean spacing, S_m [mm]		Compass dimension, $D_{C\,1000}$ [—]		EEE dimension, $D_{\mathrm{EEE}\,1000}$ [—]		Relative length, L_R [—]		Proportional length, L_P [—]	
	H_0 (0,95)*	P value**	H_0 (0,95)*	P value**	H_0 (0,95)*	P-value**	H_0 (0,95)*	P value**	H_0 (0,95)*	P value**	H_0 (0,95)*	P value**	H_0 (0,95)*	P value**	H_0 (0,95)*	P value**	H_0 (0,95)*	P value**
1A	1	0.726	1	0.766	1	0.533	1	0.615	1	0.517	1	0.839	1	0.805	1	0.819	1	0.819
1B	1	0.529	1	0.482	1	0.408	1	0.425	1	0.781	1	0.286	1	0.278	1	0.287	1	0.287
2A	1	0.930	1	0.808	1	0.821	1	0.628	1	0.723	1	0.299	1	0.071	1	0.282	1	0.282
2B	1	0.903	1	0.920	1	0.273	1	0.449	1	0.500	1	0.807	1	0.775	1	0.967	1	0.967
3A	1	0.849	1	0.908	1	0.448	1	0.842	1	0.545	1	0.596	1	0.494	1	0.494	1	0.494
3B	1	0.562	1	0.873	0	0.006	1	0.382	1	0.434	1	0.930	1	0.770	1	0.950	1	0.950
4A	1	0.861	1	0.897	1	0.979	1	0.567	1	0.866	1	0.804	1	0.853	1	0.945	1	0.945
4B	1	0.990	1	0.998	1	0.924	1	0.824	1	0.455	1	0.850	1	0.958	1	0.933	1	0.933
5A	1	0.336	1	0.326	1	0.773	1	0.795	0	0.033	0	0.032	1	0.167	1	0.625	1	0.626
5B	1	0.102	1	0.227	0	0.035	1	0.135	1	0.484	1	0.945	1	0.767	1	0.699	1	0.700
6A	0	0.049	1	0.070	1	0.260	1	0.347	1	0.414	1	0.847	1	0.296	1	0.728	1	0.728
6B	1	0.798	1	0.926	1	0.889	1	0.903	1	0.294	1	0.439	1	0.984	1	0.708	1	0.710
7A	1	0.951	1	0.808	1	0.983	1	0.326	1	0.399	1	0.415	1	0.753	1	0.635	1	0.635
7B	1	0.502	1	0.484	1	0.345	1	0.580	1	0.392	1	0.562	1	0.214	1	0.503	1	0.506
8A	1	0.566	1	0.669	1	0.596	1	0.250	0	0.048	0	0.007	0	0.003	0	0.003	0	0.003
8B	1	0.085	1	0.289	0	0.037	1	0.336	1	0.299	0	0.005	0	0.000	0	0.001	0	0.001
9A	1	0.074	1	0.091	1	0.538	1	0.372	0	0.001	0	0.000	0	0.000	0	0.000	0	0.000
9B	0	0.000	0	0.000	1	0.707	1	0.156	0	0.000	0	0.000	0	0.000	0	0.000	0	0.000
10A	1	0.541	1	0.567	0	0.001	0	0.000	0	0.000	0	0.000	0	0.000	0	0.000	0	0.000
10B	1	0.442	1	0.626	0	0.001	0	0.000	0	0.000	0	0.000	0	0.000	0	0.000	0	0.000
11A	1	0.192	1	0.188	1	0.620	1	0.908	0	0.004	1	0.062	1	0.141	1	0.072	1	0.072
11B	1	0.454	1	0.241	1	0.998	1	0.895	1	0.151	1	0.220	1	0.153	1	0.210	1	0.210
12A	0	0.001	0	0.002	0	0.000	0	0.000	0	0.000	0	0.000	0	0.000	0	0.000	0	0.000
12B	1	0.272	1	0.671	1	0.993	1	0.207	0	0.000	0	0.000	0	0.000	0	0.000	0	0.000
13A	1	0.988	1	0.978	1	0.892	1	0.106	0	0.000	0	0.000	0	0.000	0	0.000	0	0.000
13B	1	0.926	1	0.959	1	0.282	1	0.171	0	0.005	0	0.000	0	0.000	0	0.000	0	0.000
14A	1	0.706	1	0.785	1	0.666	1	0.128	1	0.110	0	0.010	0	0.010	0	0.012	0	0.012
14B	1	0.200	1	0.167	1	0.225	1	0.169	1	0.534	0	0.025	0	0.038	0	0.038	0	0.038

*Null hypothesis H_0: value 1 means results are independent of direction; value 0 means results are dependent on direction.

**P values for one-way analysis of variance (ANOVA) for the test (9).

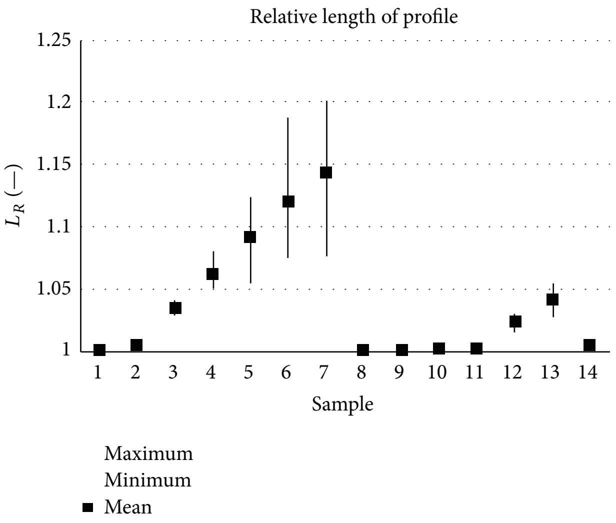


Figure 12: Results of relative length measurement, L_R (x-axis).

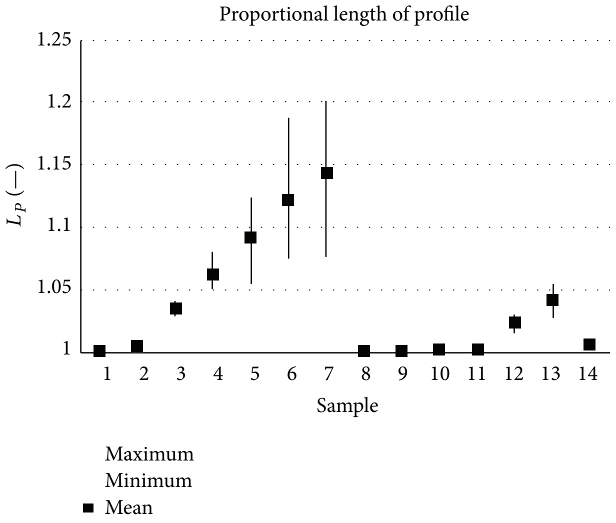


Figure 13: Results of proportional length measurement, L_P (x-axis).

oriented structure that is identified as a random structure. The results show that the mentioned parameters are useful for finding directionally dependent and independent structures. However, the conclusion is valid only for the analysed data and must be verified in further research.

5. Conclusions

The methodology for evaluation of complex and irregular data was developed and applied in industrial practice. The fractal dimension is used in combination with statistical tools; thus commonly used parameters and relatively new parameters are used simultaneously. This methodology searches appropriated parameters for a complex evaluation of data. Only the chosen parameters are used for a complete analysis of the data in order to reduce processing time.

We conducted this research to verify and find properties of the methodology on data measured from 14 samples. The samples were produced by 5 different technologies (commonly used in industry) under different production properties. The samples were measured using 3 methods: by a surface roughness tester, by an electron microscope, and by an optical microscope.

In the first phase of the research we analysed data sets obtained from a surface roughness tester. The nine used parameters were divided into sets: parameters of amplitude, parameters of frequency, parameters of complexity, and deformation. One parameter in each set was determined using the correlation coefficient to evaluate these data types: average roughness, P_a(*parameter of amplitude*), Mean Spacing, S_m (*parameter of frequency*), and compass dimension, $D_{C\,1000}$ (*parameter of complexity and deformation*). These 3 parameters provide diverse information about the data and can be used for a complete data analysis. Within the framework of the research, sensitivity to sample directionality for these types of surfaces was determined. Parameters of complexity and deformation: compass dimension ($D_{C\,1000}$), EEE dimension ($D_{\mathrm{EEE}\,1000}$), relative length (L_R), and proportional length (L_P) can be used for linear structure recognition of the presented data. Based on these results, it can be inferred that the tools represented here are suitable for recognition directionally dependent and independent structures. De facto one-way analysis of variance (ANOVA) illustrates the parameter sensitivity of complexity and deformation to the detection of random structures. Verification of whether the structure is chaotic and also if the structure must be chaotic for detection with the specified procedure will be carried out.

Our future work will focus on two other forms of data: 2D images and dividing lines. Further research will also compare data analyses in various forms (sequences, signals, 2D images, and dividing lines). The potential of the mentioned methodology for industrial practise will be verified. Subsequently, verification of whether the description of complex data is only possible with the use of fractal dimension or sufficient "standard tools" (especially statistical tools) will be executed. Chaotic properties of obtained data will also be studied, because they come from real dynamic systems that can be chaotic.

The fractal dimension is widely used in science, but industrial applications are rather rare. Data analysis using the fractal dimension has great potential in combination with statistical and other measurements in industry. This and previously presented results show possibilities of application in practical use in industry and production laboratories. Structured surface, complex time series, and difficulty describing dividing lines are much more common than can be expected.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The results of this project LO1201 were obtained through the financial support of the Ministry of Education, Youth and Sports in the framework of the targeted support of the "National Programme for Sustainability I", the OPR&DI Project Centre for Nanomaterials, Advanced Technologies and Innovation CZ.1.05/2.1.00/01.0005, and the Grant of Students Grant Contest of the Technical University of Liberec, no. SGS 21006/115, which use special-purpose support for the university research and is financed by the Ministry of Education, Czech Republic.

References

[1] B. B. Mandelbrot, *The Fractal Geometry of Nature*, W. H. Freeman and Co., San Francisco, Calif, USA, 1982.

[2] H. Peitgen, H. Jürgens, and D. Saupe, *Chaos and Fractals: New Frontiers of Science*, Springer, New York, NY. USA, 1992.

[3] C. J. G. Evertsz, H. O. Peitgen, and R. F. Voss, *Fractal Geometry and Analysis*, World Scientific Publishing, Singapore, 1996.

[4] R. C. Hilborn, *Chaos and Nonlinear Dynamics*, 2003.

[5] D. Gulick, *Encounters with Chaos*, McGraw-Hill, 1992.

[6] M. Chadli, I. Zelinka, and T. Youssef, "Unknown inputs observer design for fuzzy systems with application to chaotic system reconstruction," *Computers & Mathematics with Applications*, vol. 66, no. 2, pp. 147–154, 2013.

[7] I. Zelinka, M. Chadli, D. Davendra, R. Senkerik, and R. Jasek, "An investigation on evolutionary reconstruction of continuous chaotic systems," *Mathematical and Computer Modelling*, vol. 57, no. 1-2, pp. 2–15, 2013.

[8] V. J. Levy, E. Lutton, and C. Tricot, *Fractals in Engineering*, Springer, New York, NY, USA, 1997.

[9] A. Conci and C. B. Proenca, "A fractal image analysis system for fabric inspection based on a box-counting method," *Computer Networks and ISDN Systems*, vol. 30, no. 20-21, pp. 1887–1895, 1998.

[10] T. Wang, X. G. Liu, and Z. Y. Zhang, "Characterization of chaotic multiscale features on the time series of melt index in industrial propylene polymerization system," *Journal of the Franklin Institute—Engineering and Applied Mathematics*, vol. 351, pp. 878–906, 2014.

[11] A. Nurkkala, F. Pettersson, and H. Saxén, "Nonlinear modeling method applied to prediction of hot metal silicon in the ironmaking blast furnace," *Industrial and Engineering Chemistry Research*, vol. 50, no. 15, pp. 9236–9248, 2011.

[12] L. Yu and D. Qi, "Applying multifractal spectrum combined with fractal discrete Brownian motion model to wood defects recognition," *Wood Science and Technology*, vol. 45, no. 3, pp. 511–519, 2011.

[13] J. N. Muguthu and D. Gao, "Profile fractal dimension and dimensional accuracy analysis in machining metal matrix composites (MMCs)," *Materials and Manufacturing Processes*, vol. 28, no. 10, pp. 1102–1109, 2013.

[14] C. X. Zheng, D. W. Sun, and L. Y. Zheng, "Recent applications of image texture for evaluation of food qualities—a review," *Trends in Food Science & Technology*, vol. 17, no. 3, pp. 113–128, 2006.

[15] V. Hotař, "Fractal geometry for industrial data evaluation," *Computers and Mathematics with Applications*, vol. 66, no. 2, pp. 113–121, 2013.

[16] V. Hotař, F. Novotný, and H. Reinischová, "Objective evaluation of the corrugation test for sheet glass surfaces," *Glass Technology: European Journal of Glass Science and Technology A*, vol. 52, no. 6, pp. 197–202, 2011.

[17] V. Hotař and F. Novotný, "Surface profile evaluation by fractal dimension and statistic tools," in *Proceedings of the 11th International Conference on Fracture*, CCI Centro Congressi Internazionale, Turin, Italy, 2005.

[18] V. Hotar and F. Novotny, "Evaluation of surface defects by fractal geometry and statistical analysis," *Glass Science and Technology*, vol. 77, pp. 230–237, 2004.

[19] A. Hotar, P. Kratochvil, and V. Hotar, "The corrosion resistance of Fe3Al-based iron aluminides in molten glasses," *Kovove Materialy—Metallic Materials*, vol. 47, pp. 247–252, 2009.

[20] V. Hotař, "EEE Method: improved approach of compass dimension calculation," in *Proceedings of the Advances in Intelligent Systems and Computing (Nostradamus '13)*, I. Zelinka, V. Snasel, G. Chen, A. Abraham, and E. Rossler, Eds., vol. 210, pp. 343–351, Ostrava, Czech Republic, 2013.

[21] ISO 4287, *Geometrical Product Specifications (GPS)—Surface Texture: Profile Method—Terms, Definitions and Surface Texture Parameters*, International Organization for Standardization, Geneva, Switzerland, 1997.

[22] I. Statistics, *Introduction to ANOVA, Regression, and Logistic Regression*, Sas Inst., 2005.