ADVANCES IN INTELLIGENT SYSTEMS AND COMPUTING 210

Ivan Zelinka
Guanrong Chen
Otto E. Rössler
Václav Snášel
Ajith Abraham (Eds.)

Nostradamus 2013: Prediction, Modeling and Analysis of Complex Systems

Springer

Advances in Intelligent Systems and Computing

Volume 210

Series Editor

J. Kacprzyk, Warsaw, Poland

For further volumes:
http://www.springer.com/series/11156

Ivan Zelinka · Guanrong Chen
Otto E. Rössler · Václav Snášel
Ajith Abraham
Editors

Nostradamus 2013: Prediction, Modeling and Analysis of Complex Systems

Springer

Editors

Ivan Zelinka
VŠB-TUO
Faculty of Electrical Eng. and Comp. Sci
Department of Computer Science
Ostrava-Poruba
Czech Republic

Václav Snášel
VŠB-TUO
Faculty of Electrical Eng. and Comp. Sci.
Department of Computer Science
Ostrava-Poruba
Czech Republic

Guanrong Chen
Department of Electronic Engineering
City University of Hong Kong
Hong Kong
Kowloon
China, People's Republic

Ajith Abraham
Machine Intelligence Research Labs
Scientific Network for Innovation and
Research Excellence
Auburn Washington
USA

Otto E. Rössler
Institute of Physical and
Theoretical Chemistry
University of Tuebingen
Tuebingen
Germany

ISSN 2194-5357 ISSN 2194-5365 (electronic)
ISBN 978-3-319-00541-6 ISBN 978-3-319-00542-3 (eBook)
DOI 10.1007/978-3-319-00542-3
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013937614

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This proceeding book of Nostradamus conference (http://nostradamus-conference.org) contains accepted papers presented at this event in June 2013. Nostradamus conference was held in one of the biggest cities-Ostrava (the Czech Republic, http://www.ostrava.cz/en).

Conference topics are focused on classical as well as modern methods for prediction of dynamical systems with applications in science, engineering and economy. Topics are (but not limited to): *prediction by classical and novel methods, predictive control, deterministic chaos and its control, complex systems, modeling and prediction of its dynamics, interdisciplinary fusion of chaos, randomness and evolution* and much more.

Prediction of behavior of the dynamical systems and modeling of its structure is vitally important problem in engineering, economy and science today. Examples of such systems can be seen in the world around us and of course in almost every scientific discipline, including such "exotic" domains like the earth's atmosphere, turbulent fluids, economies (exchange rate and stock markets), information flow in social networks and its dynamics, population growth, physics (control of plasma), chemistry and complex networks. The main aim of the conference is to create periodical possibility for students, academics and researchers to exchange their ideas and novel methods. This conference will establish a forum for the presentation and discussion of recent trends in the area of applications of various modern as well as classical methods for researchers, students and academics.

The selection of papers was extremely rigorous in order to maintain the high quality of the conference that is supported by grantno. CZ.1.07/2.3.00/20.0072 funded by Operational Programme Education for Competitiveness, co-financed by ESF. Regular as well as larger amount of student's papers has been submitted to the conference and topics, that are in accordance with ESF support as well as with conference topics, has been accepted after positive review. Based on accepted papers structure and topics, proceeding book consist of sections like: *Chaos, Evolution and Complexity* discussion topics from the field of evolutionary algorithms, deterministic chaos, its complex dynamics and mutual intersections of all three topics (chaos powered evolutionary algorithms, engineering of mathematical chaotic circuits, etc.). Section *Nature-Inspired Algorithms and Nonlinear Systems* contain participations about use of bio-inspired algorithms on various problems (forecasting, EEG signal modeling, battleship game strategy, evolutionary synthesis of complex structures, etc.). Section *Nonlinearand Predictive Control* and *Nonlinear Dynamicsand Complex Systems* contain papers about controlling modeling and analysis of complex and nonlinear systems and the last section, *Various Topics*, contains a few borderline papers that seem to be still interesting and belonging to the conference topics.

For this year, as a follow-up of the conference, we anticipate further publication of selected papers in a special issue of the prestigious journal Soft Computing, International Journal of Automation and Computing, special book in Emergence Complexity and Computationseries and more.

We would like to thank the members of the Program Committees and reviewers for their hard work. We believe that Nostradamus conference represents high standard conference in the domain of prediction and modeling of complex systems. Nostradamus 2013 enjoyed outstanding keynote speeches by distinguished guest speakers:**Guanrong Chen** (Hong Kong), **Miguel A. F. Sanjuan** (Spain), **Gennady Leonov and Nikolay Kuznetsov** (Russia), **Petr Škoda** (Czech Republic).

Particular thanks goes as well to the Workshop main Sponsors, IT4Innovations, VŠB-Technical University of Ostrava, MIR labs (USA), Centre for Chaos and Complex Networks (Hong Kong), Journal of Unconventional Computing (UK). Special thanks belong to Ministry of Education of the Czech Republic. This conference was supported by the Development of human resources in research and development of latest soft computing methods and their application in practice project, reg. no. CZ.1.07/2.3.00/20.0072 funded by Operational Programme Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic.

We would like to thank all the contributing authors, as well as the members of the Program Committees and the Local Organizing Committee for their hard and highly valuable work. Their work has contributed to the success of the Nostradamus conference.

The editors
Ivan Zelinka
Guanrong Chen
Otto E. Rössler
Václav Snášel
Ajith Abraham

Organization

This conference was supported by the Development of human resources in research and development of latest soft computing methods and their application in practice project, reg. no. CZ.1.07/2.3.00/20.0072 funded by Operational Programme Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic.

esf european social fund in the czech republic

EUROPEAN UNION

MŠMT MINISTRY OF EDUCATION, YOUTH AND SPORTS

2007-13 OP Education for Competitiveness

INVESTMENTS IN EDUCATION DEVELOPMENT

International Conference Committee

Edward Ott (USA)
Ivan Zelinka (Czech Republic)
Guanrong Chen (Hong Kong)
Otto E. Rössler (Germany)
Sergej Čelikovsky (Czech Republic)
Mohammed Chadli (France)
Ajith Abraham (MIR Labs, USA)
Václav Snášel (Czech Republic)
Emilio Corchado (Spain)
Andy Adamatzky (UK)
Jiří Pospíchal (Slovakia))
Jouni Lampinen (Finland)
Juan Carlos Burguillo-Rial (Spain)
Pandian Vasant (Malysia)
Petr Dostál (Czech Republic)
Davendra Donald (Fiji, Czech Republic)
Bernabé Dorronsoro (Luxembourg)
Oplatková Zuzana (Czech Republic)
Linqiang Pan (China)
Šenkeřík Roman (Czech Republic)
Fečkan Michal (Slovakia)
Jašek Roman (Czech Republic)
Joanna Kolodziej (Poland)
Radek Matoušek (Czech Republic)
Hendrik Richter (Germany)
Zdeněk Beran (Czech Republic)
Ana Peleteiro (Spain)
Vadim Strijov (Russia)
Oldřich Zmeškal (Czech Republic)
Masoud Mohammadian (Australia)
Miguel A.F. Sanjuan (Spain)
Gennady Leonov (Russia)
Nikolay Kuznetsov (Russia)
René Lozi (France)

Local Conference Committee

Jan Martinovič
Lenka Skanderová
Jan Platoš
Eliška Odchodková
Martin Milata
Pavel Krömer
Michal Krumnikl

Miloš Kudělka
Pavel Moravec
Jiří Dvorský
Tilkova Ludmila
Kvapulinska Petra

Contents

Keynote Speakers

Constructing a Simple Chaotic System with an Arbitrary Number of Equilibrium Points or an Arbitrary Number of Scrolls 1
Guanrong Chen

Dynamics of Partial Control of Chaotic Systems 3
Miguel A.F. Sanjuan

Prediction of Hidden Oscillations Existence in Nonlinear Dynamical Systems: Analytics and Simulation 5
Gennady A. Leonov, Nikolay V. Kuznetsov

Astroinformatics: Getting New Knowledge from the Astronomical Data Avalanche 15
Petr Škoda

Chaos, Evolution and Complexity

Engineering of Mathematical Chaotic Circuits 17
René Lozi

Utilising the Chaos-Induced Discrete Self Organising Migrating Algorithm to Schedule the Lot-Streaming Flowshop Scheduling Problem with Setup Time 31
Donald Davendra, Roman Senkerik, Ivan Zelinka, Michal Pluhacek, Magdalena Bialic-Davendra

Hidden Periodicity – Chaos Dependance on Numerical Precision 47
Ivan Zelinka, Mohammed Chadli, Donald Davendra, Roman Senkerik, Michal Pluhacek, Jouni Lampinen

Do Evolutionary Algorithms Indeed Require Random Numbers? Extended Study. 61
Ivan Zelinka, Mohammed Chadli, Donald Davendra, Roman Senkerik, Michal Pluhacek, Jouni Lampinen

New Adaptive Approach for Chaos PSO Algorithm Driven Alternately by Two Different Chaotic Maps – An Initial Study 77
Michal Pluhacek, Roman Senkerik, Ivan Zelinka, Donald Davendra

On the Performance of Enhanced PSO Algorithm with Dissipative Chaotic Map in the Task of High Dimensional Optimization Problems 89
Michal Pluhacek, Roman Senkerik, Ivan Zelinka, Donald Davendra

Utilization of Analytic Programming for Evolutionary Synthesis of the Robust Controller for Set of Chaotic Systems 101
Roman Senkerik, Zuzana Kominkova Oplatkova, Ivan Zelinka, Michal Pluhacek

Chaos Powered Selected Evolutionary Algorithms 111
Lenka Skanderova, Ivan Zelinka, Petr Šaloun

Case Study of Evolutionary Process Visualization Using Complex Networks 125
Patrik Dubec, Jan Plucar, Lukáš Rapant

Stabilization of Chaotic Logistic Equation Using HC12 and Grammatical Evolution 137
Radomil Matousek and Petr Minar

Hypervolume-Driven Analytical Programming for Solar-Powered Irrigation System Optimization 147
T. Ganesan, I. Elamvazuthi, Ku Zilati Ku Shaari, P. Vasant

Nature-Inspired Algorithms and Nonlinear Systems

Forecasting of Time Series with Fuzzy Logic 155
Petr Dostál

Unknown Input Proportional Integral Observer Design for Chaotic TS Fuzzy Models 163
T. Youssef, Mohammed Chadli, Ivan Zelinka, M. Zelmat

Modeling of EEG Signal with Homeostatic Neural Network 175
Martin Ruzek

Model Identification from Incomplete Data Set Describing State Variable Subset Only – The Problem of Optimizing and Predicting Heuristic Incorporation into Evolutionary System 181
Tomas Brandejsky

Supervised and Reinforcement Learning in Neural Network Based Approach to the Battleship Game Strategy 191
Ladislav Clementis

Evolutionary Algorithms for Parameter Estimation of Metabolic Systems 201
Anastasia Slustikova Lebedik, Ivan Zelinka

Evolutionary Synthesis of Complex Structures – Pseudo Neural Networks for the Task of Iris Dataset Classification 211
Zuzana Kominkova Oplatkova, Roman Senkerik

Speech Emotions Recognition Using 2-D Neural Classifier 221
Pavol Partila, Miroslav Voznak

Nonlinear and Predictive Control

Predictive Control of Radio Telescope Using Multi-layer Perceptron Neural Network 233
Sergej Jakovlev, Miroslav Voznak, Kestutis Ruibys, Arunas Andziulis

Modeling and Simulation of a Small Unmanned Aerial Vehicle 245
Ozan Eren Yuceol, Ahmet Akbulut

Mathematical Models of Controlled Systems 257
Vladimír Jehlička

Effect of Weighting Factors in Adaptive LQ Control 265
Jiri Vojtesek, Petr Dostal

Unstable Systems Database: A New Tool for Students, Teachers and Scientists 275
František Gazdoš, Jiří Marholt, Jaroslav Kolařík

Nonlinear State Estimation and Predictive Control of pH Neutralization Process 285
Jakub Novák, Petr Chalupa

State Observers for Model Predictive Control 295
Petr Chalupa, Peter Januška, Jakub Novák

Nonlinear Dynamics and Complex Systems

Characteristics of the Chen Attractor 305
Petra Augustová, Zdeněk Beran

Message Embedded Synchronization for the Generalized Lorenz System and Its Use for Chaotic Masking 313
Sergej Čelikovský, Volodymyr Lynnyk

Using Complex Network Topologies and Self-Organizing Maps for Time Series Prediction 323
Juan C. Burguillo, Bernabé Dorronsoro

Initial Errors Growth in Chaotic Low-Dimensional Weather Prediction Model 333
Hynek Bednar, Ales Raidl, Jiri Miksovsky

EEE Method: Improved Approach of Compass Dimension Calculation 343
Vlastimil Hotař

Chaotic Analysis of the GDP Time Series 353
Radko Kříž

Daily Temperature Profile Prediction for the District Heating Application 363
Juraj Koščák, Rudolf Jakša, Rudolf Sepeši, Peter Sinčák

Adaptive Classifier of Candlestick Formations for Prediction of Trends 373
Ján Vaščák, Peter Sinčák, Karol Prešovský

Identification of Economic Agglomerations by Means of Accounting Data from ERP Systems of Business Entities 385
Petr Hanzal, Ivana Faltová Leitmanová

Nonlinear Spatial Analysis of Dynamic Behavior of Rural Regions 401
Yi Chen, Guanfeng Zhang, Bin Zheng, Ivan Zelinka

Complex System Simulation Parameters Settings Methodology 413
Michal Janošek, Václav Kocian, Eva Volná

Simulation Analysis of the Complex Production System with Interoperation Buffer Stores 423
Bronislav Chramcov, Robert Bucki, Sabina Marusza

Usage of Modern Exponential-Smoothing Models in Network Traffic Modelling 435
Roman Jašek, Anna Szmit, Maciej Szmit

On Approaches of Assessment of Tribo Data from Medium Lorry Truck 445
David Valis, Libor Zak, Agata Walek

Various Topics

Energy and Entropy of Fractal Objects: Application to Gravitational Field 455
Oldrich Zmeskal, Michal Vesely, Petr Dzik, Martin Vala

Wavelet Based Feature Extraction for Clustering of Be Stars 467
Pavla Bromová, Petr Škoda, Jaroslav Zendulka

Upcoming Features of SPLAT-VO in Astroinformatics 475
Petr Šaloun, David Andrešič, Petr Škoda, Ivan Zelinka

Mobile Sensor Data Classification Using GM-SOM 487
Petr Gajdoš, Pavel Moravec, Pavel Dohnálek, Tomáš Peterek

Tensor Modification of Orthogonal Matching Pursuit Based Classifier in Human Activity Recognition 497
Pavel Dohnálek, Petr Gajdoš, Tomáš Peterek

Global Motion Estimation Using a New Motion Vector Outlier Rejection Algorithm 507
Burak Yıldırım and Hakkı Alparslan Ilgın

Searching for Dependences within the System of Measuring Stations by Using Symbolic Regression 517
Petr Gajdoš, Michal Radecký, Miroslav Vozňák

The Effect of Sub-sampling on Hyperspectral Dimension Reduction 529
Ali Ömer Kozal, Mustafa Teke, Hakkı Alparslan Ilgın

Author Index 539

Constructing a Simple Chaotic System with an Arbitrary Number of Equilibrium Points or an Arbitrary Number of Scrolls

Guanrong Chen

Abstract. In a typical 3D smooth autonomous chaotic system, such as the Lorenz and the Rssler systems, the number of equilibria is three or less and the number of scrolls in their attractors is two or less. Today, we are able to construct a relatively simple smooth 3D autonomous chaotic system that can have any desired number of equilibria or any desired number of scrolls in its chaotic attractor. Nowadays it is known that a 3D quadratic autonomous chaotic system can have no equilibrium, one equilibrium, two equilibria, or three equilibria. Starting with a chaotic system with only one stable equilibrium, by adding symmetry to it via a suitable local diffeomorphism, we are able to transform it to a locally topologically equivalent chaotic system with an arbitrary number of equilibria. In so doing, the stability of the equilibria can also be easily adjusted by tuning a single parameter. Another interesting issue of constructing a 3D smooth autonomous chaotic system with an arbitrary number of scrolls is discussed next. To do so, we first establish a basic system that satisfies Shilnikovs inequalities. We then search for a heteroclinic orbit that connects the two equilibria of the basic system. Finally, we use a copy and lift technique and a switching control method to timely switch the dynamics between nearby sub-systems, thereby generating a chaotic attractor with multiple scrolls. Not only the number but also the positions of the scrolls in the chaotic attractor can be determined by our design method. This talk will briefly introduce the ideas and methodologies.

Guanrong Chen
Department of Electronic Engineering, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong SAR, P.R. China
e-mail: eegchen@cityu.edu.hk

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, p. 1.
DOI: 10.1007/978-3-319-00542-3_1

Dynamics of Partial Control of Chaotic Systems

Miguel A.F. Sanjuan

Abstract. In our chaotic lives we usually do not try to specify our plans in great detail, or if we do, we should be prepared to make major modifications. Our plans for what we want to achieve are accompanied with situations we must avoid. Disturbances often disrupt our immediate plans, so we adapt to new situations. We only have partial control over our futures. Partial control aims at providing toy examples of chaotic situations where we try to avoid disasters, constantly revising our trajectories. More mathematically, partial control of chaotic systems is a new kind of control of chaotic dynamical systems in presence of disturbances. The goal of partial control is to avoid certain undesired behaviors without determining a specific trajectory. The surprising advantage of this control technique is that it sometimes allows the avoidance of the undesired behaviors even if the control applied is smaller than the external disturbances of the dynamical system. A key ingredient of this technique is what we call safe sets. Recently we have found a general algorithm for finding these sets in an arbitrary dynamical system, if they exist. The appearance of these safe sets can be rather complex though they do not appear to have fractal boundaries. In order to understand better the dynamics on these sets, we introduce in this paper a new concept, the asymptotic safe set. Trajectories in the safe set tend asymptotically to the asymptotic safe set. We present two algorithms for finding such sets. We illustrate all these concepts for a time-2π map of the Duffing oscillator. This is joint work with James A. Yorke (USA), Samuel Zambrano (Italy) and Juan Sabuco (Spain).

Miguel A.F. Sanjuan
Professor of Physics, Director of the Department of Physics, Nonlinear Dynamics,
Chaos and Complex Systems Group, Universidad Rey Juan Carlos,
28933 Mostoles, Madrid, Spain
e-mail: miguel.sanjuan@urjc.es

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 3–4.
DOI: 10.1007/978-3-319-00542-3_2

References

1. Zambrano, S., Sanjuan, M.A.F., Yorke, J.A.: Partial Control of Chaotic Systems. Phys. Rev. E 77, 055201(R) (2008)
2. Zambrano, S., Sanjuan, M.A.F.: Exploring Partial Control of Chaotic Systems. Phys. Rev. E 79, 026217 (2009)
3. Sabuco, J., Zambrano, S., Sanjuan, M.A.F.: Partial control of chaotic transients and escape times. New J. Phys. 12, 113038 (2010)
4. Sabuco, J., Zambrano, S., Sanjuan, M.A.F., Yorke, J.A.: Finding safety in partially controllable chaotic systems. Commun Nonlinear Sci. Numer. Simulat. 17, 4274–4280 (2012)
5. Sabuco, J., Sanjuan, M.A.F., Yorke, J.A.: Dynamics of Partial Control. Chaos 22(4), 047507 (2012)

Prediction of Hidden Oscillations Existence in Nonlinear Dynamical Systems: Analytics and Simulation

Gennady A. Leonov and Nikolay V. Kuznetsov

Abstract. From a computational point of view, in nonlinear dynamical systems, attractors can be regarded as self-excited and hidden attractors. Self-excited attractors can be localized numerically by a standard computational procedure, in which after a transient process a trajectory, starting from a point of unstable manifold in a neighborhood of equilibrium, reaches a state of oscillation, therefore one can easily identify it. In contrast, for a hidden attractor, a basin of attraction does not intersect neighborhoods of equilibria. While classical attractors are self-excited, attractors can therefore be obtained numerically by the standard computational procedure, for localization of hidden attractors it is necessary to develop special procedures, since there are no similar transient processes leading to such attractors. This keynote lecture is devoted to affective analytical-numerical methods for localization of hidden oscillations in nonlinear dynamical systems and their application to well known fundamental problems and applied models.

1 Introduction

An oscillation in dynamical system can be easily localized numerically if initial conditions from its open neighborhood lead to long-time behavior that approaches the oscillation. Such oscillation (or a set of oscillations) is called an attractor, and its attracting set is called the basin of attraction [49, 27]. Thus, from a computational

Gennady A. Leonov
Faculty of Mathematics and Mechanics, Saint Petersburg State University,
Universitetski pr. 28, Saint-Petersburg, 198504, Russia
e-mail: leonov@math.spbu.ru

Nikolay V. Kuznetsov
Dept. of Mathematical Information Technology, University of Jyväskylä,
P.O. Box 35 (Agora), FIN-40014, Finland
e-mail: nkuznetsov239@gmail.com

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 5–13.
DOI: 10.1007/978-3-319-00542-3_3

point of view in applied problems of nonlinear analysis of dynamical models, it is essential to regard attractors [44, 45, 40, 32] as *self-excited* and *hidden attractors* depending on simplicity of finding its basin of attraction in the phase space.

For a self-excited attractors its basin of attraction is connected with an unstable equilibrium: self-excited attractors can be localized numerically by *standard computational procedure*, in which after a transient process a trajectory, started from a point of unstable manifold in a neighborhood of equilibrium, reaches a state of oscillation therefore one can easily identify it. In contrast, for a *hidden attractor, its basin of attraction does not intersect with small neighborhoods of equilibria*.

For numerical localization of hidden attractors it is necessary to develop special analytical-numerical procedures, since there are no similar transient processes leading to such attractors from neighborhoods of equilibria. Remark, that one cannot guarantee finding of an attractor (especially for multidimensional systems) by the integration of trajectories with random initial conditions since basin of attraction can be very small.

2 Self-excited Attractor Localization

In the first half of the last century during the initial period of the development of the theory of nonlinear oscillations [55, 19, 3, 53], a main attention was given to analysis and synthesis of oscillating systems, for which the problem of the existence of oscillations can be solved with relative ease.

These investigations were encouraged by the applied research of periodic oscillations in mechanics, electronics, chemistry, biology and so on (see, e.g., [54]) The structure of many applied systems (see, e.g., Duffing [13], van der Pol [51], Tricomi [56], Beluosov-Zhabotinsky [4] systems) was such that the existence of oscillations was "almost obvious" since the oscillations were excited from unstable equilibria (*self-excited oscillation*). This allowed one, following Poincare's advice "*to construct the curves defined by differential equations*" [50], to visualize periodic oscillations by standard computational procedure.

Then, in the middle of 20th century, it was found numerically the existence of chaotic oscillations [57, 47], which were also excited from an unstable equilibrium and could be computed by the standard computational procedure. Nowadays there is enormous number of publications devoted to the computation and analysis of self-exited chaotic oscillations (see, e.g., [52, 11, 9] and others).

In Fig. 1 numerical localization of classical self-exited oscillation are shown: van der Pol oscillator [51], one of the modification of Belousov-Zhabotinsky reaction [4], two prey and one predator model [14].

In Fig. 2 examples of visualization of classical self-excited chaotic attractors are presented: Lorenz system [47], Rössler system [52], "double-scroll" attractor in Chua's circuit [5].

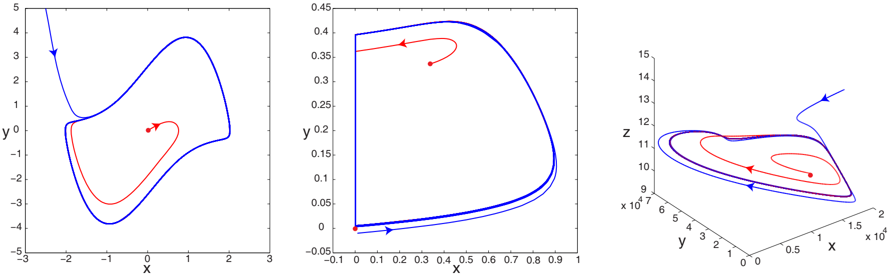


Fig. 1 Standard computation of classical self-excited periodic oscillations

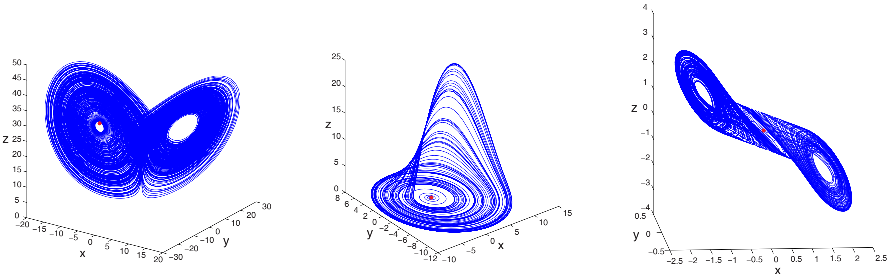


Fig. 2 Standard computation of classical self-excited chaotic attractors

3 Hidden Attractor Localization

While classical attractors are self-exited attractors therefore can be obtained numerically by the standard computational procedure, for localization of hidden attractors it is necessary to develop special procedures, since there are no similar transient processes leading to such attractors. At first, the problem of investigating hidden oscillations arose in the second part of Hilbert's 16th problem (1900) on the number and possible dispositions of limit cycles in two-dimensional polynomial systems (see, e.g., [46] and authors' works [34, 23, 42, 36, 43, 21, 32]). The the problem is still far from being resolved even for a simple class of quadratic systems.

Later, the problem of analyzing hidden oscillations arose from engineering problems in automatic control. In 50s of the last century in M.Kapranov's works [17] on stability of phase locked-loops (PLL) systems, widely used nowadays in telecommunications and computer architectures, the qualitative behavior of systems was studied and the estimate of stability domain was obtained. In these investigations Kapranov assumed that in PLL systems there were self-excited oscillations only. However, in 1961, N.Gubar' [32] revealed a gap in Kapranov's work and showed analytically the possibility of the existence of hidden oscillations in two-dimensional system of phase locked-loop: thus from the computational point of view the system considered was globally stable (all the trajectories tend to equilibria), but, in fact, there was a bounded domain of attraction only.

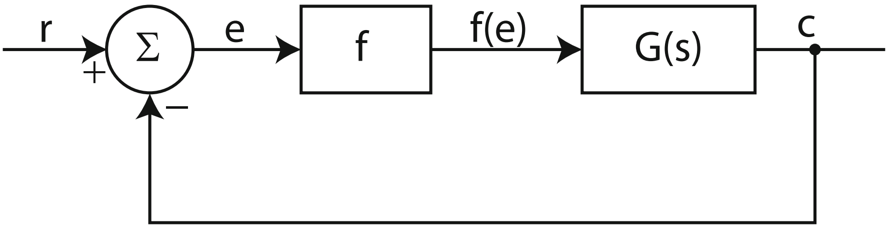


Fig. 3 Nonlinear control system. $G(s)$ is a linear transfer function, $f(e)$ is a single-valued, continuous, and differentiable function [16].

In 1957 year R.E. Kalman stated the following [16]: "*If $f(e)$ in Fig. 1* [see Fig. 3] *is replaced by constants K corresponding to all possible values of $f'(e)$, and it is found that the closed-loop system is stable for all such K, then it intuitively clear that the system must be monostable; i.e., all transient solutions will converge to a unique, stable critical point.*" Kalman's conjecture is a strengthening of Aizerman's conjecture [2], in which in place of condition on the derivative of nonlinearity it is required that the nonlinearity itself belongs to linear sector.

In the last century the investigations of widely known Aizerman's, and Kalman's conjectures on absolute stability have led to the finding of hidden oscillations in automatic control systems with a unique stable stationary point and with a nonlinearity, which belongs to the sector of linear stability (see, e.g., [31, 30, 6, 37, 7, 24, 38, 41, 32]).

The generalization of Kalman's conjecture to multidimensional nonlinearity is known as Markus-Yamabe conjecture [48] (which is also proved to be false [12]).

At the end of the last century the difficulties the difficulties of numerical analysis of hidden oscillations arose in simulations of aircraft's control systems (anti-windup) and drilling systems which caused crashes [26, 29, 8, 18, 33, 32].

Further investigations on hidden oscillations were greatly encouraged by the present authors' discovery, in 2009-2010 (for the first time), of *chaotic hidden attractor* in Chua's circuits (simple electronic circuit with nonlinear feedback) [25, 44, 7].

Until recently, only self excited attractors have been found in Chua circuits. Note that L. Chua himself, in analyzing various cases of attractors existence in Chua's circuit [10], does not admit the existence of hidden attractor (discovered later) in his circuits. Now, it is shown that Chua's circuit and its various modifications [44, 45, 20] can exhibit hidden chaotic attractors (see, Fig.4, b), Fig.5) with positive largest Lyapunov exponent[22, 35][1].

[1] While there are known effects of the largest Lyapunov exponent (LE) sign inversion ([35, 22]) for nonregular time-varying linearizations, computation of Lyapunov exponents for linearization of nonlinear autonomous system along non stationary trajectories is widely used for investigation of chaos, where positiveness of the largest Lyapunov exponent is often considered as indication of chaotic behavior in considered nonlinear system.

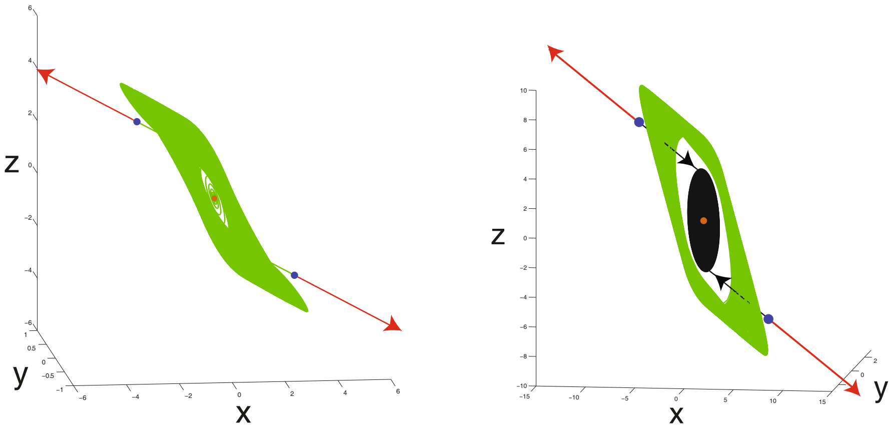


Fig. 4 a) Self-excited and b) Hidden Chua attractor with similar shapes

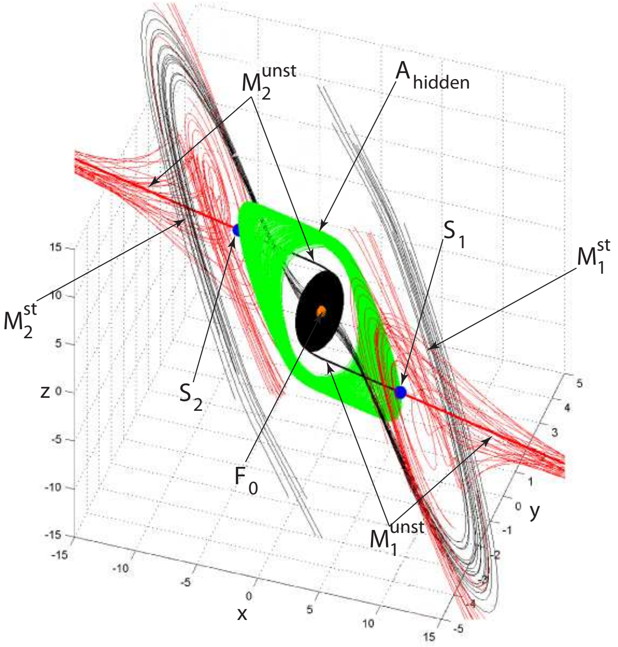


Fig. 5 Hidden chaotic attractor (green domain) in classical Chua circuit: locally stable zero equilibrium F_0 attracts trajectories (black) from stable manifolds $M^{st}_{1,2}$ of two saddle points $S_{1,2}$; trajectories (red) from unstable manifolds $M^{unst}_{1,2}$ tend to infinity

4 Conclusion

Since one cannot guarantee revealing complex oscillations regime by linear analysis and standard simulation, rigorous nonlinear analysis and special numerical methods should be used for investigation of nonlinear dynamical systems.

It was found [28, 38, 24, 39, 41, 32] that the effective methods for the numerical localization of hidden attractors in multidimensional dynamical systems are the methods based on special modifications of *describing function method*[2] and *numerical continuation*: it is constructed a sequence of similar systems such that for the first (starting) system the initial data for numerical computation of possible oscillating solution (starting oscillation) can be obtained analytically and then the transformation of this starting oscillation when passing from one system to another is followed numerically.

Also some recent examples of hidden attractors can be found in [63, 60, 59, 58, 61, 62, 1, 15].

Acknowledgements. This work was supported by the Academy of Finland, Russian Ministry of Education and Science (Federal target programm), Russian Foundation for Basic Research and Saint-Petersburg State University.

References

1. Abooee, A., Yaghini-Bonabi, H., Jahed-Motlagh, M.: Analysis and circuitry realization of a novel three-dimensional chaotic system. Commun Nonlinear Sci. Numer. Simulat. 18, 1235–1245 (2013)
2. Aizerman, M.A.: On a problem concerning the stability in the large of dynamical systems. Uspekhi Mat. Nauk 4, 187–188 (1949) (in Russian)
3. Andronov, A.A., Vitt, A.A., Khaikin, S.E.: Theory of Oscillators. Pergamon, Oxford (1966)
4. Belousov, B.P.: A periodic reaction and its mechanism. In: Collection of short papers on radiation medicine for 1958,.Med. Publ., Moscow (1959) (in Russian)
5. Bilotta, E., Pantano, P.: A gallery of Chua attractors, vol. Series A. 61. World Scientific (2008)
6. Bragin, V.O., Kuznetsov, N.V., Leonov, G.A.: Algorithm for counterexamples construction for Aizerman's and Kalman's conjectures. IFAC Proceedings Volumes (IFAC-PapersOnline) 4(1), 24–28 (2010), doi:10.3182/20100826-3-TR-4016.00008
7. Bragin, V.O., Vagaitsev, V.I., Kuznetsov, N.V., Leonov, G.A.: Algorithms for finding hidden oscillations in nonlinear systems. The Aizerman and Kalman conjectures and Chua's circuits. Journal of Computer and Systems Sciences International 50(4), 511–543 (2011), doi:10.1134/S106423071104006X

[2] In engineering practice for the analysis of the existence of periodic solutions it is widely used classical harmonic linearization and describing function method (DFM). However these classical method is not strictly mathematical reasonable and can lead to untrue results (e.g., DFM proves validity of Aizerman's and Kalman's conjectures on absolute system if control system, while counterexample with hidden oscillation are well known).

8. de Bruin, J., Doris, A., van de Wouw, N., Heemels, W., Nijmeijer, H.: Control of mechanical motion systems with non-collocation of actuation and friction: A Popov criterion approach for input-to-state stability and set-valued nonlinearities. Automatica 45(2), 405–415 (2009)
9. Chen, G., Ueta, T.: Yet another chaotic attractor. International Journal of Bifurcation and Chaos 9(7), 1465–1466 (1999)
10. Chua, L.: A zoo of strange attractors from the canonical Chua's circuits. In: Proceedings of the IEEE 35th Midwest Symposium on Circuits and Systems (Cat. No.92CH3099-9), vol. 2, pp. 916–926 (1992)
11. Chua, L.O., Komuro, M., Matsumoto, T.: The double scroll family. IEEE Transactions on Circuits and Systems CAS-33(11), 1072–1118 (1986)
12. Cima, A., van den Essen, A., Gasull, A., Hubbers, E., Mañosas, F.: A polynomial counterexample to the Markus-Yamabe conjecture. Advances in Mathematics 131(2), 453–457 (1997)
13. Duffing, G.: Erzwungene Schwingungen bei Veranderlicher Eigenfrequenz. F. Vieweg u. Sohn, Braunschweig (1918)
14. Fujii, K.: Complexity-stability relationship of two-prey-one-predator species system model; local and global stability. Journal of Theoretical Biology 69(4), 613–623 (1977)
15. Jafari, S., Sprott, J., Golpayegani, S.: Elementary quadratic chaotic flows with no equilibria. Physics Letters A 377, 699–702 (2013)
16. Kalman, R.E.: Physical and mathematical mechanisms of instability in nonlinear automatic control systems. Transactions of ASME 79(3), 553–566 (1957)
17. Kapranov, M.: Locking band for phase-locked loop. Radiofizika 2(12), 37–52 (1956) (in Russian)
18. Kiseleva, M.A., Kuznetsov, N.V., Leonov, G.A., Neittaanmäki, P.: Drilling systems failures and hidden oscillations. In: Proceedings of IEEE 4th International Conference on Nonlinear Science and Complexity, NSC 2012, pp. 109–112 (2012), doi:10.1109/NSC.2012.6304736
19. Krylov, A.N.: The Vibration of Ships. GRSL, Moscow (1936) (in Russian)
20. Kuznetsov, N., Kuznetsova, O., Leonov, G., Vagaitsev, V.: Analytical-numerical localization of hidden attractor in electrical Chua's circuit. In: Ferrier, J.-L., Bernard, A., Gusikhin, O., Madani, K. (eds.) Informatics in Control, Automation and Robotics. LNEE, vol. 174, pp. 149–158. Springer, Heidelberg (2013)
21. Kuznetsov, N.V., Kuznetsova, O.A., Leonov, G.A.: Visualization of four normal size limit cycles in two-dimensional polynomial quadratic system. Differential Equations and Dynamical Systems 21(1-2), 29–34 (2013), doi:10.1007/s12591-012-0118-6
22. Kuznetsov, N.V., Leonov, G.A.: On stability by the first approximation for discrete systems. In: Proceedings of International Conference on Physics and Control, PhysCon 2005, vol. 2005, pp. 596–599 (2005), doi:10.1109/PHYCON.2005.1514053
23. Kuznetsov, N.V., Leonov, G.A.: Lyapunov quantities, limit cycles and strange behavior of trajectories in two-dimensional quadratic systems. Journal of Vibroengineering 10(4), 460–467 (2008)
24. Kuznetsov, N.V., Leonov, G.A., Seledzhi, S.M.: Hidden oscillations in nonlinear control systems. IFAC Proceedings Volumes (IFAC-PapersOnline) 18(1), 2506–2510 (2011), doi:10.3182/20110828-6-IT-1002.03316
25. Kuznetsov, N.V., Leonov, G.A., Vagaitsev, V.I.: Analytical-numerical method for attractor localization of generalized Chua's system. IFAC Proceedings Volumes (IFAC-PapersOnline) 4(1), 29–33 (2010), doi:10.3182/20100826-3-TR-4016.00009
26. Lauvdal, T., Murray, R., Fossen, T.: Stabilization of integrator chains in the presence of magnitude and rate saturations: a gain scheduling approach. In: Proc. IEEE Control and Decision Conference, vol. 4, pp. 4404–4005 (1997)

27. Leonov, G.A.: Strange attractors and classical stability theory. St.Petersburg University Press, St.Petersburg (2008)
28. Leonov, G.A.: Effective methods for periodic oscillations search in dynamical systems. App. Math. & Mech. 74(1), 24–50 (2010)
29. Leonov, G.A., Andrievskii, B.R., Kuznetsov, N.V., Pogromskii, A.Y.: Aircraft control with anti-windup compensation. Differential equations 48(13), 1700–1720 (2012), doi:10.1134/S001226611213
30. Leonov, G.A., Bragin, V.O., Kuznetsov, N.V.: Algorithm for constructing counterexamples to the Kalman problem. Doklady Mathematics 82(1), 540–542 (2010), doi:10.1134/S1064562410040101
31. Leonov, G.A., Bragin, V.O., Kuznetsov, N.V.: On problems of Aizerman and Kalman. Vestnik St. Petersburg University. Mathematics 43(3), 148–162 (2010), doi:10.3103/S1063454110030052
32. Leonov, G.A., Kuznetsov, G.V.: Hidden attractors in dynamical systems. From hidden oscillations in Hilbert-Kolmogorov, Aizerman, and Kalman problems to hidden chaotic attractors in Chua circuits. International Journal of Bifurcation and Chaos 23(1), Art. no. 1330002 (2013), doi:10.1142/S0218127413300024
33. Leonov, G.A., Kuznetsov, G.V.: Hidden oscillations in drilling systems: torsional vibrations. Journal of Applied Nonlinear Dynamics 2(1), 83–94 (2013), doi:10.5890/JAND.2012.09.006
34. Leonov, G.A., Kuznetsov, N.V.: Computation of the first lyapunov quantity for the second-order dynamical system. IFAC Proceedings Volumes (IFAC-PapersOnline) 3, 87–89 (2007), doi:10.3182/20070829-3-RU-4912.00014
35. Leonov, G.A., Kuznetsov, N.V.: Time-varying linearization and the Perron effects. International Journal of Bifurcation and Chaos 17(4), 1079–1107 (2007), doi:10.1142/S0218127407017732
36. Leonov, G.A., Kuznetsov, N.V.: Limit cycles of quadratic systems with a perturbed weak focus of order 3 and a saddle equilibrium at infinity. Doklady Mathematics 82(2), 693–696 (2010), doi:10.1134/S1064562410050042
37. Leonov, G.A., Kuznetsov, N.V.: Algorithms for searching for hidden oscillations in the Aizerman and Kalman problems. Doklady Mathematics 84(1), 475–481 (2011), doi:10.1134/S1064562411040120
38. Leonov, G.A., Kuznetsov, N.V.: Analytical-numerical methods for investigation of hidden oscillations in nonlinear control systems. IFAC Proceedings Volumes (IFAC-PapersOnline) 18(1), 2494–2505 (2011), doi:10.3182/20110828-6-IT-1002.03315
39. Leonov, G.A., Kuznetsov, N.V.: Hidden oscillations in dynamical systems: 16 Hilbert's problem, Aizerman's and Kalman's conjectures, hidden attractors in Chua's circuits. Journal of Mathematical Sciences (in print, 2013)
40. Leonov, G.A., Kuznetsov, N.V.: Hidden oscillations in dynamical systems. from hidden oscillation in 16th hilbert, aizerman and kalman problems to hidden chaotic attractor in chua circuits. In: 2012 Fifth International Workshop on Chaos-Fractals Theories and Applications, IWCFTA, pp. XV–XVII (2013), doi:10.1109/IWCFTA.2012.8
41. Leonov, G.A., Kuznetsov, N.V.: Analytical-numerical methods for hidden attractors localization: the 16th Hilbert problem, Aizerman and Kalman conjectures, and Chua circuits. In: Numerical Methods for Differential Equations, Optimization, and Technological Problems, Computational Methods in Applied Sciences, vol. 27, part I, pp. 41–64. Springer (2013), doi:10.1007/978-94-007-5288-7_3
42. Leonov, G.A., Kuznetsov, N.V., Kudryashova, E.V.: Cycles of two-dimensional systems: Computer calculations, proofs, and experiments. Vestnik St.Petersburg University. Mathematics 41(3), 216–250 (2008), doi:10.3103/S1063454108030047

43. Leonov, G.A., Kuznetsov, N.V., Kudryashova, E.V.: A direct method for calculating Lyapunov quantities of two-dimensional dynamical systems. Proceedings of the Steklov Institute of Mathematics 272(suppl. 1), S119–S127 (2011), doi:10.1134/S008154381102009X
44. Leonov, G.A., Kuznetsov, N.V., Vagaitsev, V.I.: Localization of hidden Chua's attractors. Physics Letters A 375(23), 2230–2233 (2011), doi:10.1016/j.physleta.2011.04.037
45. Leonov, G.A., Kuznetsov, N.V., Vagaitsev, V.I.: Hidden attractor in smooth Chua systems. Physica D: Nonlinear Phenomena 241(18), 1482–1486 (2012), doi:10.1016/j.physd.2012.05.016
46. Li, J.: Hilberts 16-th problem and bifurcations of planar polynomial vector fields. International Journal of Bifurcation and Chaos 13(1), 47–106 (2003)
47. Lorenz, E.N.: Deterministic nonperiodic flow. J. Atmos. Sci. 20(2), 130–141 (1963)
48. Markus, L., Yamabe, H.: Global stability criteria for differential systems. Osaka Math. J. 12, 305–317 (1960)
49. Ott, E.: Application of the differential transformation method for the solution of the hyperchaotic Rossler system. Scholarpedia 1(8), 1701 (2006)
50. Poincaré, H.: Mémoire sur les courbes définies par les équations différentielles. Journal de Mathématiques 37, 375–422 (1881)
51. van der Pol, B.: On relaxation-oscillations. Philosophical Magazine and Journal of Science 7(2), 978–992 (1926)
52. Rossler, O.E.: An equation for continuous chaos. Physics Letters A 57(5), 397–398 (1976)
53. Stoker, J.J.: Nonlinear Vibrations in Mechanical and Electrical Systems. Interscience, N.Y (1950)
54. Strogatz, H.: Nonlinear Dynamics and Chaos. With Applications to Physics, Biology, Chemistry, and Engineering. Westview Press (1994)
55. Timoshenko, S.: Vibration Problems in Engineering. Van Nostrand, N.Y (1928)
56. Tricomi, F.: Integrazione di unequazione differenziale presentatasi in elettrotechnica. Annali della R. Shcuola Normale Superiore di Pisa 2(2), 1–20 (1933)
57. Ueda, Y., Akamatsu, N., Hayashi, C.: Computer simulations and non-periodic oscillations. Trans. IEICE Japan 56A(4), 218–255 (1973)
58. Wang, X., Chen, G.: A chaotic system with only one stable equilibrium. Commun Nonlinear Sci. Numer. Simulat. 17, 1264–1272 (2012)
59. Wang, X., Chen, G.: Constructing a chaotic system with any number of equilibria. Nonlinear Dyn. (2012), doi:10.1007/s11071-012-0669-7
60. Wei, Z.: Dynamical behaviors of a chaotic system with no equilibria. Physics Letters A 376, 102–108 (2011)
61. Wei, Z.: Complex dynamics of a new chaotic system without equilibria. In: 2012 Fifth International Workshop on Chaos-fractals Theories and Applications, pp. 79–82 (2012)
62. Wei, Z., Pehlivan, I.: Chaos, coexisting attractors, and circuit design of the generalized sprott c system with only two stable equilibria. Optoelectronics And Advanced Materials. Rapid Communications 6(7-8), 742–745 (2012)
63. Yang, Q., Wei, Z., Chen, G.: An unusual 3d autonomous quadratic chaotic system with two stable node-foci. International Journal of Bifurcation and Chaos 20(4), 1061–1083 (2010)

Astroinformatics: Getting New Knowledge from the Astronomical Data Avalanche

Petr Škoda

Abstract. The research in almost all natural sciences is facing the data avalanche represented by an exponential growth of information produced by big digital detectors and large-scale multi-dimensional computer simulations stored in the worldwide network of distributed archives. As the data volumes have been growing faster than computer technology can cope with, a qualitatively new research methodology called Data Intensive Science or X-informatics is required, based on an advanced statistics and data mining methods, as well as on a new approach to sharing huge databases in a seamless way by global research communities. This approach, sometimes presented as a Fourth Paradigm of contemporary science, promises new scientific discoveries as a result of understanding hidden dependencies and finding rare outliers in common statistical patterns extracted by machine learning methods from Peta-scale data archives. The implementation of X-informatics in astronomy, Astroinformatics, is a new emerging discipline, integrating computer science, advanced statistics, and astrophysics to yield new discoveries and better understanding of nature of astronomical objects. It has been fully benefitting from the long-term skill of astronomy of building well documented astronomical catalogues and automatically processed telescope and satellite data archives. The astronomical Virtual Observatory project plays a key role in this effort, being the global infrastructure of federated astronomical archives, web-based services, and powerful client tools supported by supercomputer grids and clusters. It is driven by strict standards describing all astronomical resources worldwide, enabling the standardized discovery and access to these collections as well as advanced visualization and analysis of large data sets. In our talk, we give a overview of the motivations, early history, and technological principles of Virtual Observatory, as well as a more philosophical view of data mining, Citizen Science, and new promises of Astroinformatics in the age of data-flooded astronomy.

Petr Škoda
Astronomical Institute of the Academy of Sciences, Czech Republic
e-mail: skoda@sunstel.asu.cas.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, p. 15.
DOI: 10.1007/978-3-319-00542-3_4

Engineering of Mathematical Chaotic Circuits

René Lozi

Abstract. We introduce the paradigm of chaotic mathematical circuitry which shows some similarity to the paradigm of electronic circuitry especially in the frame of chaotic attractors for application purpose (cryptography, generic algorithms in optimization, control, …).

Keywords: Chaos dynamics inside soft computing algorithms, Mathematical chaotic circuits.

1 Introduction

The purpose of this communication is to build an analogous of paradigm of electronic circuitry, which is the design of electronic circuit: the paradigm of chaotic mathematical circuitry, in order to improve easily the performance of well known chaotic attractors for application purpose (cryptography, generic algorithms in optimization, control, ...).

An electronic circuit is composed of individual electronic components, such as resistors, transistors, capacitors, inductors and diodes, connected by conductive wires through which electric current can flow. The combination of components and wires allows various simple and complex operations to be performed: signals can be amplified, computations can be accomplished, and data can be moved from one place to another. We introduce in the same way mathematical circuits which are composed of individual components (generators, couplers, samplers, mixers, and reducers, ...) connected through streams of data. The combination of such mathematical components leads to several news applications such as improving the performance of well known chaotic attractors (Hénon, Chua, Lorenz, Rössler, ...) for application purpose (chaotic cryptography, evolutionary and genetic algorithms in optimization, control,...). In Sec. 2 we present the symbols we introduce in the paradigm of mathematical circuits (generators, couplers, samplers, mixers, and reducers). In Sec. 3 we consider the design of two circuits, the first one for Chaotic

René Lozi
Laboratoire J.A. Dieudonné, UMR CNRS 7351, Université de Nice-Sophia Antipolis,
Parc Valrose, 06108 NICE Cedex 02, France
e-mail: rlozi@unice.fr

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 17–29.
DOI: 10.1007/978-3-319-00542-3_5

multistream PseudoRandom Number Generators, the second one for Noise-resisting cryptographic transmitter and receiver. The conclusion is given in Sec. 4.

2 Elementary Components of Mathematical Circuits

Analog circuits are very commonly represented in schematic diagrams, in which wires are shown as lines, and each component has a unique symbol. We present in this section the first symbols we design in order to draw mathematical schematic diagrams.

2.1 Generator

The first class of symbol we describe, generator symbols, are, from a mathematical point of view, equivalent to a battery or a current generator in electronic circuit. However we consider that they generate a numerical signal (in one or several dimensions) rather than a voltage or an intensity variation (nonetheless, a voltage or intensity variation can be considered as a physical signal which can be discretized). This signal can be either continuous, as in Chua's circuit, Lorenz or Rössler attractors or discrete as in Hénon or logistic map.

2.1.1 Chua’s Circuit: A Prototype for Continuous Generator

Nowadays there is a worldwide tradition of use of the circuit invented by L.O. Chua in October 1983 for several purposes [2]. This circuit of Fig. 1(a) contains three linear energy-storage elements (an inductor and two capacitors), a linear resistor, and a single nonlinear resistor, namely, Chua's diode (Fig. 1(b)) with three segment linear characteristic defined by

$$f(v_R)=m_0 v_R+\frac{1}{2}(m_1-m_0)\left[\left|v_R+B_p\right|-\left|v_R-B_p\right|\right] \tag{1}$$

where the slopes in the inner and the outer regions are m_0 and m_0, respectively, and $\pm B_p$ denotes the breakpoints.

The dynamics of Chua's circuit is governed by Eq. (2) where V_{C_1}, V_{C_2}, and I_L are respectively the voltages across the capacitors C_1 and C_2, and the intensity of the electrical current through the inductor L.

$$\begin{cases} C_1\dfrac{dv_{C_1}}{dt}=G\left(v_{C_2}-v_{C_1}\right)-f\left(v_{C_1}\right), \\ C_2\dfrac{dv_{C_2}}{dt}=G\left(v_{C_1}-v_{C_2}\right)+i_L \\ L\dfrac{di_L}{dt}=-v_{C_2} \end{cases} \tag{2}$$

Equation (2) can be transformed into the third-order autonomous differential equation whose dimension-less form is

$$\begin{cases}\dot{x}=\alpha(y-x-f(x)),\\ \dot{y}=x-y+z\\ \dot{z}=-\beta y\end{cases}, \quad f(x)=bx+\frac{1}{2}(a-b)\left[|x+1|-|x-1|\right] \qquad (3)$$

The parameter value

$$\alpha=15.60\ ,\ b=28.58\ ,\ a=-\frac{1}{7}\ \text{and}\ b=\frac{2}{7} \qquad (4)$$

are very often used in order to generate chaotic signal. Even if the scheme of Fig. 1(a) is easily understandable for an electronics engineer, it is of no help to build a device using mathematical properties of chaos (like a secure communication system based on it [6]). This is why it is more useful to represent Chua's circuit like a chaos generator by the diagram of Fig. 2(a).

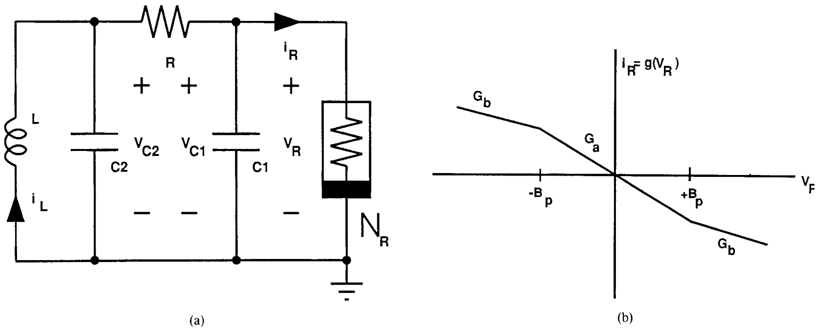


Fig. 1(a) Realization of Chua's circuit. **(b).** Three-segment piecewise-linear *v-i* characteristic of nonlinear voltage controlled resistor (Chua's diode).

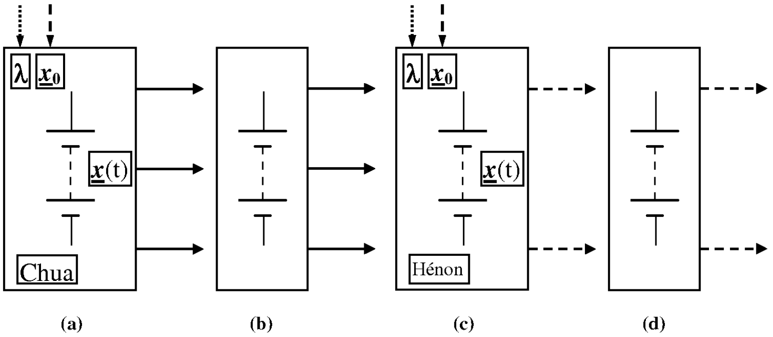


Fig. 2(a) Chua's circuit: continuous generator. **(b).** Simplified symbol of continuous generator, **(c).** Discrete generator (Hénon map, expanded symbol). **(d).** Simplified symbol discrete generator.

On this expanded symbol of continuous generator, the solid line arrows coming out from the generator represent the three components of the signal $\underline{x}(t)=(x(t),y(t),z(t))$, the dashed line arrow points at λ which stands for the parameter value defined by Eq. (4) and the dot line arrow points at $\underline{x}_0=\underline{x}(0)$ the given initial value of the signal.

If there is no ambiguity on the nature of the generator used, the symbol can be simplified as in Fig. 2(b).

2.1.2 Discrete Generator: Hénon and Logistic Map

We need also to design chaotic circuitry for discrete signal. For this purpose some classical generators can be considered: in dimension 2, the Hénon map [4] defined by

$$H_{a,b} : \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} y+1-ax^2 \\ bx \end{pmatrix} \tag{5}$$

with

$$a = 1.4, \quad b = 0.3, \tag{6}$$

which must be associated to the dynamical system

$$\begin{cases} x_{n+1}=y \ +1-ax_n^2 \\ y_{n+1}=bx_n \end{cases} \tag{7}$$

in order to provide a stream of chaotic numbers (Figs. 2(c) and 2(d)). And in 1-dimension, the logistic map

$$f(x)=4x(1-x) \tag{8}$$

associated to the dynamical system

$$x_{n+1}=rx_n(1-x_n) \tag{9}$$

Thereafter, another 1-dimensional chaotic generator, the symmetric tent map, will be, also, represented by the same symbol of Fig. 3.

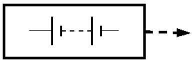


Fig. 3 1-dimensional generator (logistic or tent map)

Remark: In the rest of this article, we use solid line arrow for continuous signal $x(t)$, and dashed line arrow for discrete signal x_n.

2.2 Coupler

The experimental observation of hyperchaotic attractors in open and closed chain of Chua's circuit was reported in 1994 [5]. The layout of the five identical coupled Chua's circuit forming a ring is displayed on Fig. 4.

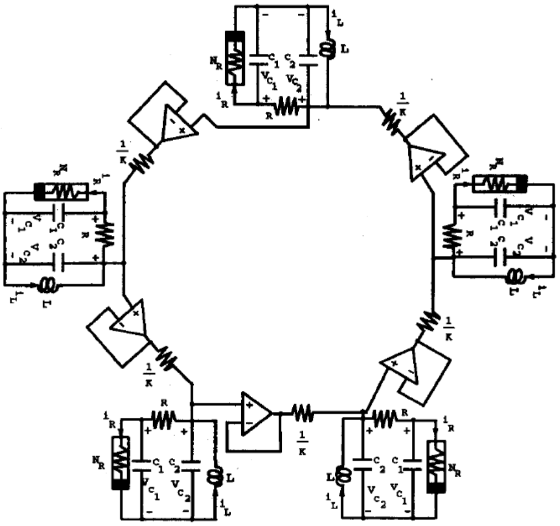


Fig. 4 Five identical coupled Chua's circuits forming a ring

The state equations of this circuit are as follows (10). Identifying the symbols $(V^{(i)}_{C_1}, V^{(i)}_{C_2}, I^{(i)}_L)$ in each Chua's circuit with (x^i, y^i, z^i), the state equations of the circuit can be translated into the differential equations (11), and the electronic circuit is symbolized by the mathematical circuit of Fig. 5.

$$\left\{\begin{array}{l}
C_1\dfrac{dv^{(1)}_{C_1}}{dt}=G\left(v^{(1)}_{C_2}-v^{(1)}_{C_1}\right)-f\left(v^{(1)}_{C_1}\right),\\
C_2\dfrac{dv^{(1)}_{C_2}}{dt}=G\left(v^{(1)}_{C_1}-v^{(1)}_{C_2}\right)+i^{(1)}_L+K_1\left(v^{(2)}_{C_2}-v^{(1)}_{C_2}\right),\\
L\dfrac{di^{(1)}_L}{dt}=-v^{(1)}_{C_2},\\
C_1\dfrac{dv^{(2)}_{C_1}}{dt}=G\left(v^{(2)}_{C_2}-v^{(2)}_{C_1}\right)-f\left(v^{(2)}_{C_1}\right),\\
C_2\dfrac{dv^{(2)}_{C_2}}{dt}=G\left(v^{(2)}_{C_1}-v^{(2)}_{C_2}\right)+i^{(2)}_L+K_2\left(v^{(3)}_{C_2}-v^{(2)}_{C_2}\right),\\
L\dfrac{di^{(2)}_L}{dt}=-v^{(2)}_{C_2},\\
\qquad\qquad\vdots\\
C_1\dfrac{dv^{(5)}_{C_1}}{dt}=G\left(v^{(5)}_{C_2}-v^{(5)}_{C_1}\right)-f\left(v^{(5)}_{C_1}\right),\\
C_2\dfrac{dv^{(5)}_{C_2}}{dt}=G\left(v^{(5)}_{C_1}-v^{(5)}_{C_2}\right)+i^{(5)}_L+K_5\left(v^{(1)}_{C_2}-v^{(5)}_{C_2}\right),\\
L\dfrac{di^{(5)}_L}{dt}=-v^{(5)}_{C_2},
\end{array}\right. \tag{10}$$

In this figure the double rounded arrows symbolize the coupling of one Chua's circuit to the next one. In order to represent the coupling between mathematical equation, depending on the nature of the coupling, we can use two different symbols: the ring coupler corresponding to the coupling of one generator to the next one (Fig. 5) and the full coupler when the coupling involves more connections between the couplers (right hand side of Fig. 6).

$$\begin{cases} \dot{x}^1=\alpha(y^1-x^1-f(x^1)), \\ \dot{y}^1=x^1-y^1+z^1+k_1(y^2-y^1), \\ \dot{z}^1=-\beta y^1, \\ \dot{x}^2=\alpha(y^2-x^2-f(x^2)), \\ \dot{y}^2=x^2-y^2+z^2+k_2(y^3-y^2), \\ \dot{z}^2=-\beta y^2, \\ \vdots \\ \dot{x}^5=\alpha(y^5-x^5-f(x^5)), \\ \dot{y}^5=x^5-y^5+z^5+k_5(y^1-y^5), \\ \dot{z}^5=-\beta y^5, \end{cases} \tag{11}$$

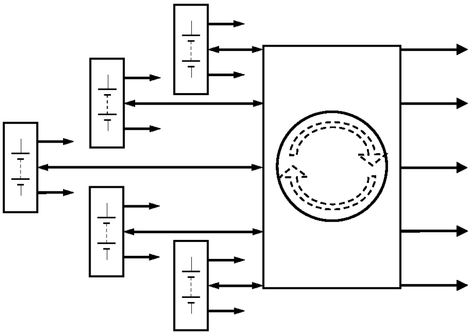


Fig. 5 Numerical circuit corresponding to the electronic circuit of Fig. 4

It has been shown, few years ago [7] that the ultra-weak coupling of several logistic or symmetric tent maps

$$f(x) = 1 - 2\left|x\right| \quad , \quad x_{n+1} = f(x_n) \tag{12}$$

allows the production of long series of chaotic numbers equally distributed over the interval $\left[-1,1\right]$ of the real line.

The system of p-coupled tent map is given by

$$X_{n+1}=F\left(X_n\right)=A\cdot\left(\underline{f}\,(X_n)\right) \tag{13}$$

where

$$X_n=\begin{pmatrix} x_n^1 \\ \vdots \\ x_n^p \end{pmatrix},\ \underline{f}(X_n)=\begin{pmatrix} f(x_n^1) \\ \vdots \\ f(x_n^p) \end{pmatrix} \text{ and,}$$

$$A = \begin{pmatrix} \varepsilon_{1,1}=1-\sum\limits_{j=2}^{j=p}\varepsilon_{1,j} & \varepsilon_{1,2} & \cdots & \varepsilon_{1,p-1} & \varepsilon_{1,p} \\ \varepsilon_{2,1} & \varepsilon_{2,2}=1-\sum\limits_{j=1,j\neq 2}^{j=p}\varepsilon_{2,j} & \cdots & \varepsilon_{2,p-1} & \varepsilon_{2,p} \\ \vdots & \ddots & & \vdots & \vdots \\ \vdots & & \ddots & \vdots & \vdots \\ \varepsilon_{p,1} & \cdots & \cdots & \varepsilon_{p,p-1} & \varepsilon_{p,p}=1-\sum\limits_{j=1}^{j=p-1}\varepsilon_{p,j} \end{pmatrix} \qquad (14)$$

The design of the corresponding mathematical circuit is displayed on Fig. 6.

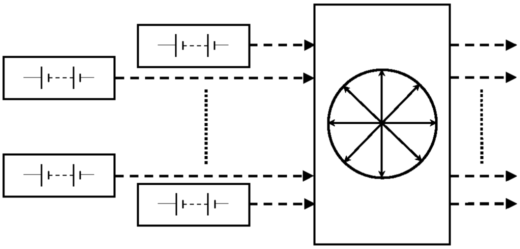


Fig. 6 Circuit of ultra-weak coupling of *p* 1-dimensional chaotic maps

2.3 Sampler

However chaotic numbers are not pseudo-random numbers because the plot of the couples of any component (x_n^l, x_{n+1}^l) of iterated points (X_n, X_{n+1}) in the corresponding phase plane reveals the map f used as one-dimensional dynamical systems to generate them via Eq. (13). Nevertheless we have recently introduced a family of enhanced Chaotic Pseudo Random Number Generators (CPRNG) in order to compute very fast long series of pseudorandom numbers with desktop computer [9]. This family is based on the ultra-weak coupling mechanism which is improved in order to conceal the chaotic genuine function.

In order to hide f of Eq. (13), in the phase space (x_n^l, x_{n+1}^l), the sequence $(x_0^l, x_1^l, x_2^l, \dots x_n^l, x_{n+1}^l, \dots)$ generated by the l-th component x^l, is sampled chaotically, selecting x_n^l every time the value x_n^m of the m-th component x^m, is strictly greater than a threshold T belonging to the interval $[-1,1]$ of the real line.

The pseudo-code, for computing such chaotically sub-sampled iterates is:

```
$X_0 = ( x_0^1, x_0^2, \dots x_0^{p-1}, x_0^p ) \;=\; seed$
$n=0; q=0;$
do { while $n < N$
    do{ while $x_n^m < T$ compute $( x_n^1, x_n^2, \dots x_n^{p-1}, x_n^p ); n++$ }
    compute$( x_n^1, x_n^2, \dots x_n^{p-1}, x_n^p )$; then $n(q) = n;\;\; \bar{x}_q = x_{n(q)}^1; n++; q++ )$ }
```

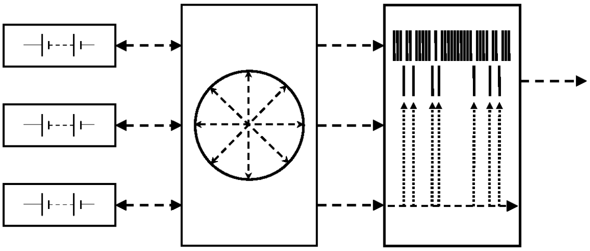


Fig. 7 Circuit of enhanced Chaotic Pseudo Random Number Generator (CPRNG) based on chaotic under-sampling

This chaotic under-sampling is possible due to the independence of each component of the iterated points X_n vs. the others [8]. We introduce the symbol on the right hand side of Fig. 7. in order to give a schematic representation of this chaotic under-sampling process.

2.4 Mixer

A second mechanism can improve the unpredictability of the pseudo-random sequence generated as above, using synergistically all the components of the vector X_n instead of two.

Given $p-1$ thresholds $0 < T_1 < T_2 < \cdots T_{p-1} < 1$ forming a partition $J_1, J_2, \dots, J_{p-1}$ of the interval $[-1,1]$, the pseudo-code, for computing such chaotically sub-sampled iterates is:

```
$X_0 = ( x_0^1, x_0^2, \dots x_0^{p-1}, x_0^p ) \;=\; seed$
$n=0; q=0;$
do { while $n < N$
    do{ while $x_n^m \in J_0$ compute $( x_n^1, x_n^2, \dots x_n^{p-1}, x_n^p ); n++$ }
    compute $( x_n^1, x_n^2, \dots x_n^{p-1}, x_n^p )$;
    let $k$ be such that $x_n^p \in J_k$; then $n(q) = n;\;\; \bar{x}_q = x_{n(q)}^k; n++; q++ )$ }
```

We draw the symbol on the right hand side of Fig. 8. in order to give a schematic representation of the chaotic mixing process. For sake of simplicity we have only displayed a circuit with three 1-dimensional generators. However the mixing process runs better when more generators are coupled.

We can say that the design of mathematical circuit including couplers, samplers or mixers allows the emergence of complexity in chaotic systems which leads to randomness [10].

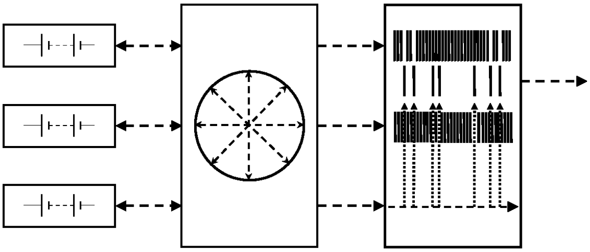


Fig. 8 Circuit of enhanced Chaotic Pseudo Random Number Generator (CPRNG) based

2.5 Reducer

We introduce now, another process which can directly provides random number without sampling or mixing, although it is possible to combine these processes with it. The idea underlying this process is to confine on $[-1,1]^n$ considered as a torus, a ring of p-coupled symmetric tent map (or logistic map) [3].

Consider the equation

$$\begin{cases} x_{n+1}^{1}=1-2\left|x_n^1\right|+k_1x_n^2 \\ \vdots \\ x_{n+1}^{m}=1-2\left|x_n^m\right|+k_mx_n^{m+1} \\ \vdots \\ x_{n+1}^{p-1}=1-2\left|x_n^{p-1}\right|+k_{p-1}x_n^p \\ x_{n+1}^{p}=1-2\left|x_n^p\right|+k_px_n^1 \end{cases} \tag{15}$$

Where the parameters $k_i = \mp 1$. In order to confine the variables x_{n+1}^{i} on this torus, we do, for every iteration, the transform

$$\begin{cases} if(\ x_{n+1}^{j}<-1)\quad add\quad 2 \\ if(\ x_{n+1}^{j}>1)\quad substract\quad 2 \end{cases} \tag{16}$$

We design a new symbol: the reducer, on the right hand side of Fig. 9, in order to give a schematic representation of the projection of the variable on the torus. For sake of simplicity we have only displayed a circuit with three 1-dimensional generators. However this new pseudo-random number generator works better when more generators are coupled.

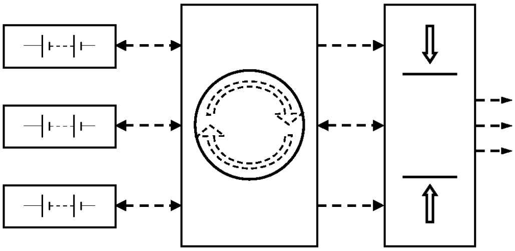


Fig. 9 Reducer for the circuit of Eq. (15) and the transform of Eq. (16)

3 Design of Mathematical Chaotic Circuits

In the limited extend of this paper; it is difficult give examples of fully developed mathematical circuits. We first give the circuit of Cms-PRNG and then we use it for the design of new transmitter and receiver of cryptographic based chaos circuit.

3.1 Chaotic multistream Pseudorandom Number Generators (Cms-PRNG)

It is possible to combine several equations in order to design chaotic multistream pseudo random number generators (Cms-PRNG) and processes in order to generate uncorrelated sequences of pseudo-random numbers, possessing a large number of keys for a cryptographic use.

$$\left\{\begin{array}{l}
x^1_{n+1}=1-2\left|x^1_n\right|+k_1\left(\left(1-\sum\limits_{j=3}^{p}\varepsilon_{1,j}\right)x^2_n+\sum\limits_{j=3}^{p}\varepsilon_{1,j}x^j_n\right)\\
\vdots\\
x^m_{n+1}=1-2\left|x^m_n\right|+k_m\left(\left(1-\sum\limits_{j=1,j\neq m;m+1}^{p}\varepsilon_{m,j}\right)x^{m+1}_n+\sum\limits_{j=1,j\neq m;m+1}^{p}\varepsilon_{m,j}x^j_n\right)\\
\vdots\\
x^{p-1}_{n+1}=1-2\left|x^{p-1}_n\right|+k_{p-1}\left(\left(1-\sum\limits_{j=1}^{p-2}\varepsilon_{p-1,j}\right)x^p_n+\sum\limits_{j=1}^{p-2}\varepsilon_{p-1,j}x^j_n\right)\\
x^p_{n+1}=1-2\left|x^p_n\right|+k_p\left(\left(1-\sum\limits_{j=2}^{p-1}\varepsilon_{p,j}\right)x^1_n+\sum\limits_{j=2}^{p-1}\varepsilon_{p,j}x^j_n\right)
\end{array}\right. \qquad (17)$$

This is simply obtained by adding a full coupler as a keyer as shown in the circuit of Fig. 10, corresponding to Eq. (17) (with the reduction process of Eq.(16)).

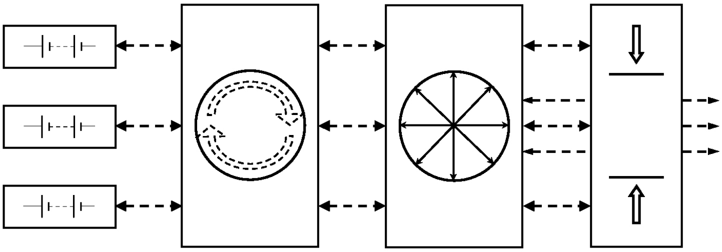


Fig. 10 Circuit of Cms-PRNG with only 3 streams

3.2 Noise-Resisting Cryptographic Transmitter and Receiver Circuits

The Cms-PRNG have been used for a novel ciphering method recently introduced in order to resist to noise which is always present during the transmission of the signal in any channel [1]. The main idea is to establish, between the transmitter and the receiver, a correspondence between the alphabet constituting the plain text and some intervals defining a partition of $[-1,1]$. Some realistic assumption about the noise boundedness allows restricting the bounds of the aforementioned intervals in order to precisely resist to the effects of the noise. An extra scrambling resorting to a co-generated chaotic sequence enhances the ciphering process. Then a new chaotic substitution method is developed: considering a chaotic carrier, belonging to the set of cogenerated and coupled pseudo-random chaotic sequences, the idea is to randomly/chaotically (in fact, this is determined by a second pseudo-random chaotic sequence) replace some elements of the carrier by a ciphered element (a letter here) of the message. At the receiver end, a copy of the Cms-PRNG, with the same parameters (hence we deal with a symmetrical ciphering method) allows to generate the necessary chaotic sequences and therefore to retrieve the initial message.

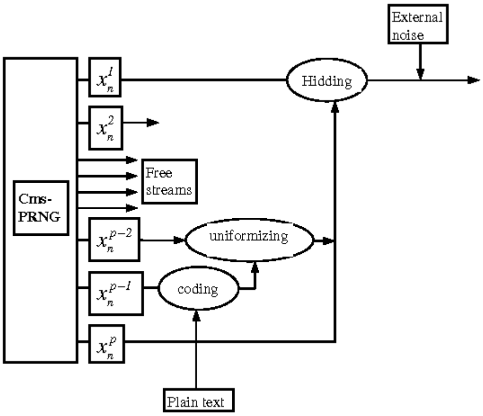


Fig. 11 Transmitter based on circuit of Fig. 10

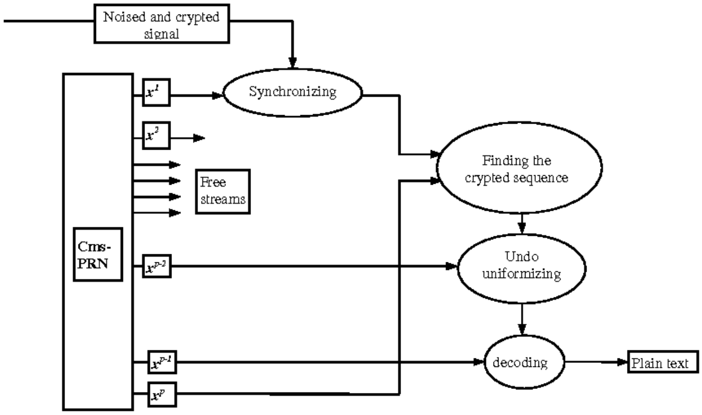


Fig. 12 Receiver of the coded signal

This process can be summarized in both circuits of Figs. 11 and 12. Due again to the limited extend of this paper, we cannot expand these figures in order to show the constituting symbols in each oval shaped region of the circuit. The originality of the method lies in the use of a chaotic pseudo-random number generator: several co-generated sequences can be used at different steps of the ciphering process, since they present the strong property of being uncorrelated. Each letter of the initial alphabet of the plain text is encoded as a subinterval of $[-1,1]$.

The bounds of each interval are defined in function of the known bound of the additive noise. A pseudo-random sequence is used to enhance the complexity of the ciphering. The transmission consists of a substitution technique inside a chaotic carrier, depending on another cogenerated sequence. The efficiency of the proposed scheme is illustrated on some numerical simulations. As further work, some studies should be performed of several sets of unknown parameters, since with the considered Cms-PRNG with 10 states, the number of possible parameters amounts to 90 (the $\varepsilon_{i,j}$ and the k_i).

4 Conclusion

Following the worldwide tradition of use of Chua's circuits for various purposes, we have introduced the paradigm of chaotic mathematical circuitry which shows some similarity to the paradigm of electronic circuitry –the design of electronic circuits. This new paradigm allows, as an example, the building of new chaotic and random number generators.

Alongside to electronic circuits, the new theory of mathematical circuits allows many new applications in chaotic cryptography, genetic algorithms in optimization and in control [11], ... Due to the versatility of the new components we introduce, the combined operation of these chaotic mathematical circuits remains largely unexplored.

References

[1] Cherrier, E., Lozi, R.: Noise-resisting ciphering based on a chaotic multi-stream pseudo-random number generator. In: Proc. IEEE Conference Internet Technology and Secured Transactions (ICITST), Abu Dhabi, December 11-14, pp. 91–96 (2011)

[2] Chua, L.O., Kumoro, M., Matsumoto, T.: The Double Scroll Family. IEEE Trans. Circuit and Systems 32(11), 1055–1058 (1984)

[3] Espinel, A., Taralova, I., Lozi, R.: Dynamical and Statistical Analysis of a New Lozi Function for Random Numbers Generation. In: Proceeding of Physcon 2011, León, Spain, September 5-8. IPACS open Access Electronic Library and Applications (2011)

[4] Hénon, M.: A Two-dimensional mapping with a strange attractor. Commun. Math. Phys. 50, 69–77 (1976)

[5] Kapitaniak, T., Chua, L.O., Zhong, G.-H.: Experimental hyperchaos in coupled Chua's circuits. IEEE Trans. Circuit and Systems 41(7), 499–503 (1994)

[6] Lozi, R., Chua, L.O.: Secure communications via chaotic synchronization II: noise reduction by cascading two identical receivers. Int. J. Bifurcation & Chaos 3(5), 1319–1325 (1993)

[7] Lozi, R.: Giga-periodic orbits for weakly coupled tent and logistic discretized maps. In: Siddiqi, A.H., Duff, I.S., Christensen, O. (eds.) Modern Mathematical Models, Methods and Algorithms for Real World Systems, pp. 80–124. Anamaya Publishers, New Delhi (2006)

[8] Lozi, R.: New Enhanced Chaotic Number Generators. Indian Journal of Industrial and Applied Mathematics 1(1), 1–23 (2008)

[9] Lozi, R.: Complexity leads to randomness in chaotic systems. In: Siddiqi, A.H., Singh, R.C., Manchanda, P. (eds.) Mathematics in Science and Technology: Mathematical Methods, Models and Algorithms in Science and Technology. Proceedings of the Satellite Conference of ICM, Delhi, India, August 15-17, pp. 93–125. World Scientific Publisher, Singapore (2010)

[10] Lozi, R.: Emergence of randomness from chaos. Int. J. Bifurcation & Chaos 22(2), 1250021, 15pages (2012)

[11] Pluhacek, M., Senkerik, R., Davendra, D., Zelinka, I.: Designing PID Controller for DC Motor by Means of Enhanced PSO Algorithm with Dissipative Chaotic Map. In: Snasel, V., Abraham, A., Corchado, E.S. (eds.) SOCO Models in Industrial & Environmental Appl. AISC, vol. 188, pp. 475–483. Springer, Heidelberg (2013)

Utilising the Chaos-Induced Discrete Self Organising Migrating Algorithm to Schedule the Lot-Streaming Flowshop Scheduling Problem with Setup Time

Donald Davendra, Roman Senkerik, Ivan Zelinka,
Michal Pluhacek, and Magdalena Bialic-Davendra

Abstract. The dissipative Lozi chaotic map is embedded in the Discrete Self Organising Migrating (DSOMA) algorithm, as a pseudorandom generator. This novel chaotic based algorithm is applied to the constraint based Lot-Streaming Flowshop scheduling problem. Two new and unique data sets generated using the Lozi and Dissipative maps are used to compare the chaos embedded DSOMA (DSOMA$_\mathrm{C}$) and the generic DSOMA utilising the venerable Mersenne Twister. In total, 100 data sets were tested by the two algorithms, for the idling and the non-idling case. From the obtained results, the DSOMA$_\mathrm{C}$ algorithm is shown to significantly improve the performance of generic DSOMA.

1 Introduction

One of the core premise of EA's is their reliance on *stochasticity*, the ability of generate a random event, which in turn, hopefully, provides the spark of perturbation towards the desired goal. The task of generating these stochasticity is generally in the realm of *pseudorandom number generators (PRNG)*; a structured sequence

Donald Davendra · Ivan Zelinka
VSB-Technical University of Ostrava, 17. listopadu 15 708 33,
Ostrava-Poruba, Czech Republic
e-mail: donald.davendra@vsb.cz

Roman Senkerik
Faculty of Applied Informatics, Tomas Bata University in Zlin
e-mail: senkerik@fai.utb.cz

Michal Pluhacek
Faculty of Applied Informatics, Tomas Bata University in Zlin
e-mail: pluhacek@fai.utb.cz

Magdalena Bialic-Davendra
Centre for Applied Economic Research, Tomas Bata University in Zlin
e-mail: bialic@fame.utb.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 31–45.
DOI: 10.1007/978-3-319-00542-3_6

of mathematical formulation which tries to yield a generally optimal range of distributed numbers between a specified range.

A wide variety of such random number generators exist, however the most common in usage is the *Mersenne Twister* [13]. A number of its variants have been designed; for a full listing please see [12]. Some other common PRNG are the *Mother Of All*, *CryptoAPI*, *Indirection, Shift, Accumulate, Add, and Count* (ISAAC), *Keep it Simple Stupid* (KISS) and *Mutiply-With-Carry* (MWC).

This paper explores a novel approach to generating PRNG, one with a lineage in chaos theory. The term *chaos* describes the complex behaviour of simple, well behaved systems. When casually observed, this behaviour can seem erratic and somewhat random, however these systems are deterministic, whose precise knowledge of future behaviour is well known. The question is then to reconcile the notion of nonlinearity of these systems.

Sudden and dramatic changes in some nonlinear systems may give rise to complex behaviour called chaos. The noun *chaos* and the adjective *chaotic* are used to describe the time behaviour of a system when that behaviour is aperiodic (it *never* exactly repeats) and is appears apparently random or noisy [10].

This aperiodic non-repeating behaviour of chaotic systems is the core foundation of this research. The objective is then to use a valid chaotic system, and embed it in the EA's as a PRNG. Four general branches of chaotic systems exist, which are the dissipative systems, fractals, dissipative and high-dimensional systems and conservative systems. The systems used for this research is the *discrete* dissipative systems of Lozi and Dissipative maps. These two systems are relatively simple, in terms of period density, and therefore easier to obtain data through sectional cropping.

Many chaotic maps in the literature possess certainty, ergodicity and the stochastic property. Recently, chaotic sequences have been adopted instead of random sequences with improved results. They have been used to improve the performance of EA's ([1]) and [2]). They have also been used together with some heuristic optimisation algorithms ([9] and [22]) to express optimisation variables. The choice of chaotic sequences is justified theoretically by their unpredictability, i.e., by their spread-spectrum characteristic, non-periodic, complex temporal behaviour, and ergodic properties [14].

Differential Evolution (DE) has also been modified using chaotic sequences. [9] has applied the canonical DE to solve the PID optimisation problem, whereas [14] applied a sequence of chaotic maps to optimise a range of benchmark problems, with the conclusion that the Sinus map and Circle map have somewhat increased the solution quality, with the ability to escape the local optima. The economic dispatch problem was solved by [11], where the *Tent Map* was utilised as a chaotic local search in order to bypass the local optima.

Concerning combinatorial optimisation problems, [20] has developed a chaotic hybrid DE, where the parameter selection and operation is handled by chaotic sequences, whereas [7] utilised both the population generation and parameter selection using a hybrid DE - Scatter Search algorithm to solve the traveling salesman problem.

The motivation of this work is to ascertain if any improvement can be attained in an EA by embedding a chaos map. Since the current literature has applied chaos to a magnitude of such research with varying application of the chaos maps (population generation, mutation crossover, escaping the local optima etc), we decided to totally replace the PRNG with a chaotic map. The Lozi map was chosen as it is a very simple map, which can be totally described in a few equations, making it easier to implement. Additionally the Lozi map has proven to be a excellent generator in the pervious work on DE for real value optimisation [9].

This research looks at modifying the Discrete Self Organising Migrating algorithm ([4], [8], [6]) with the Lozi chaotic map and utilising it to solve the lot-streaming flowshop problem with setup time. The primary aim of this work, therefore is to ascertain if there is any improvement to the algorithm when comparing the chaotic variant to the standard DSOMA variant, utilising the best PRNG; the Mersenne Twister. Secondly, we have generated two unique data sets using the Lozi and Dissipative systems for testing.

2 Discrete Self-organising Migrating Algorithm

Discrete Self Organising Migrating (DSOMA) ([4], [8], [6]) is the discrete version of SOMA [21], developed to solve permutation based combinatorial optimisation problem. The SOMA ideology of the sampling of the space between two individuals is modified for application in the combinatorial domain. Assume that there are two individuals in a search space. The objective for DSOMA is to transverse from one individual to another, while mapping each discrete space between these two individuals.

The major input of this algorithm is the sampling of the jump sequence between the individuals in the populations, and the procedure of constructing new trial individuals from these sampled jump sequence elements.

The overall outline for DSOMA can be given as:

1. **Initial Phase**
   a. *Population Generation*: An initial number of permutative trial individuals is generated for the initial population.
   b. *Fitness Evaluation*: Each individual is evaluated for its fitness.
2. **DSOMA**
   a. *Creating Jump Sequences*: Taking two individuals, a number of possible jump positions is calculated between each corresponding element.
   b. *Constructing Trial Individuals*: Using the jump positions; a number of trial individuals is generated. Each element is selected from a jump element between the two individuals.
   c. *Repairment*: The trial individuals are checked for feasibility and those, which contain an incomplete schedule, are repaired.

Table 1 DSOMA parameters

Name	Range	Type	
J_{min}	(1+)	Control	Minimum number of jumps
Population	10+	Control	Number of individuals
Migrations	10+	Termination	Total number of iterations

3. **Selection**
   a. *New Individual Selection*: The new individuals are evaluated for their fitness and the best new fitness based individual replaces the old individual, if it improves upon its fitness.
4. **Generations**
   a. *Iteration*: Iterate the population till a specified migration.

DSOMA requires a number of parameters as given in Table 1. The major addition is the parameter J_{min}, which gives the minimum number of jumps (sampling) between two individuals. The SOMA variables PathLength, StepSize and PRT Vector are not initialised as they are dynamically calculated by DSOMA using the adjacent elements between the individuals.

2.1 Initialisation

The population is initialised as a permutative schedule representative of the size of the problem at hand (1). As this is the initial population, the superscript of x its set to 0. The *rand()* function obtains a value between 0 and 1, and the *INT()* function rounds down the real number to the nearest integer. The *if* condition checks to ensure that each element within the individual is unique.

$$\begin{array}{l} x_{i,j}^0 = \begin{cases} 1+INT\left(rand\left(\right)\cdot\left(N-1\right)\right) \\ \quad \text{if } x_{i,j}^0 \notin \left\{x_{i,1}^0,\dots,x_{i,j-1}^0\right\} \end{cases} \\ \qquad\quad i=1,\dots,\beta;\ j=1,\dots,N \end{array} \tag{1}$$

Each individual is vetted for its fitness (2), and the best individual, whose index in the population can be assigned as L (leader) and it is designated the leader as X_L^0 with its best fitness given as C_L^0.

$$C_i^0 = \Im\left(X_i^0\right), \quad i=1,\dots,\beta \tag{2}$$

After the generation of the initial population, the migration counter t is initialised to 1 where $t = 1,\dots,M$ and the individual index i is initialised to 1, where $i = 1,\dots,\beta$. Using these values, the following sections (section 2.2 - section 2.5) are recursively applied, with the counters i and t being updated in section 2.6 and section 2.7 respectively.

2.2 Creating Jump Sequences

DSOMA operates by calculating the number of discrete jump steps that each individual has to circumnavigate. In DSOMA, the parameter of minimum jumps ($J_{\min}$) is used in lieu of PathLength, which states the minimum number of individuals or sampling between the two individuals.

Taking two individuals in the population, one as the incumbent (X_i^t) and the other as the leader (X_L^t), the possible number of jump individuals $J_{\max}$ is the mode of the difference between the adjacent values of the elements in the individual (3). A vector J of size N is created to store the difference between the adjacent elements in the individuals. The *mode ()* function obtains the most common number in J and designates it as J_{max}.

$$\begin{aligned} J_j &= \left| x_{i,j}^{t-1} - x_{L,j}^{t-1} \right|, \quad j = 1, \ldots, N \\ J_{\max} &= \begin{cases} mode\,(J), & \text{if} \quad \text{mode}\,(\text{J}) > 0 \\ 1 & \text{otherwise} \end{cases} \end{aligned} \tag{3}$$

The step size (s), can now be calculated as the integer fraction between the required jumps and possible jumps (4).

$$s = \begin{cases} \left\lfloor \frac{J_{\max}}{J_{\min}} \right\rfloor & \text{if } J_{\max} \geq J_{\min} \\ 1 \quad \text{otherwise} \end{cases} \tag{4}$$

Create a jump matrix **G**, which contains all the possible jump positions, that can be calculated as:

$$\mathbf{G}_{l,j} = \begin{cases} x_{i,j}^{t-1} + s \cdot l, & \text{if} \quad x_{i,j}^{t-1} + s \cdot l < x_{L,j}^{t-1} \text{ and } x_{i,j}^{t-1} < x_{L,j}^{t-1}, \\ x_{i,j}^{t-1} - s \cdot l, & \text{if} \quad x_{i,j}^{t-1} + s \cdot l < x_{L,j}^{t-1} \text{ and } x_{i,j}^{t-1} > x_{L,j}^{t-1}, \\ 0 \quad \text{otherwise.} \end{cases} \\ j = 1, \ldots, N; \; l = 1, \ldots, J_{\min} \tag{5}$$

2.3 Constructing Trial Individuals

For each jump sequence of two individuals, a total of $J_{\min}$ new individuals can now be constructed from the jump positions. Taking a new temporary population H $\left(H = \left\{Y_1, \ldots, Y_{J_{\min}}\right\}\right)$, in which each new individual Y_w ($w = 1, \ldots, J_{\min}$), is constructed piecewise from **G**. Each element $y_{w,j}$ $\left(Y_w = \left\{y_{w,j}, \ldots, y_{w,N}\right\},\ j = 1, 2, \ldots, N\right)$ in the individual, indexes its values from the corresponding j^{th} column in **G**. Each l^{th} ($l = 1, \ldots, J_{\min}$) position for a specific element is sequentially checked in $\mathbf{G}_{l,j}$ to ascertain if it already exists in the current individual Y_w. If this is a new element, it is then accepted in the individual, and the corresponding l^{th} value is set to zero as $\mathbf{G}_{l,j} = 0$. This iterative procedure can be given as in equation (6).

$$y_{w,j} = \begin{cases} \mathbf{G}_{l,j} \begin{cases} \text{if} \quad \mathbf{G}_{l,j} \notin \; \left\{y_{w,1}, \ldots, y_{w,j-1}\right\} \text{ and } \mathbf{G}_{l,j} \neq 0; \\ \text{then } \mathbf{G}_{l,j} = 0; \end{cases} \\ 0 \quad \text{otherwise} \end{cases} \\ l = 1, \ldots, J_{\min}; \; j = 1, \ldots, N; \; w = 1, \ldots, J_{\min} \tag{6}$$

2.4 Repairing Trial Individuals

Some individuals may exist, which may not contain a permutative schedule. The jump individuals Y_w $(w = 1,2,\ldots,J_{min})$, are constructed in such a way, that each infeasible element $y_{w,j}$ is indexed by 0.

Taking each jump individual Y_w iteratively from H, the following set of procedures can be applied recursively.

Take A and B, where A is initialised to the permutative schedule $A = \{1,2,\ldots,N\}$ and B is the complement of individual Y_w relative to A as given in equation (7).

$$B = A \backslash Y_w \tag{7}$$

If after the complement operation, B is an empty set without any elements; $B = \{\}$, then the individual is correct with a proper permutative schedule and does not require any repairment.

However, if B contains values, then these values are the missing elements in individual Y_w. The repairment procedure is now outlined. The first process is to randomise the positions of the elements in set B. Then, iterating through the elements $y_{w,j}$ $(j = 1,\ldots,N)$ in the individual Y_w, each position, where the element $y_{w,j} = 0$ is replaced by the value in B. Assigning B_{size} as the total number of elements present in B (and hence missing from the individual Y_w), the repairment procedure can be given as in equation (8).

$$y_{w,j} = \begin{cases} B_h & \text{if } y_{w,j} = 0 \\ y_{w,j} & \text{otherwise} \end{cases} \tag{8}$$
$$h = 1,\ldots,B_{size};\ j = 1,\ldots,N$$

After each individual is repaired in H, it is then evaluated for its fitness value as in equation (9) and stored in γ, the fitness array of size J_{min}.

$$\gamma_w = \Im\left(Y_w\right), \quad w = 1,\ldots,J_{\min} \tag{9}$$

2.5 Population Update

2 Opt local search is applied to the best individual Y_{best} obtained with the minimum fitness value $(\min(\gamma_w))$. After the local search routine, the new individual is compared with the fitness of the incumbent individual X_i^{t-1}, and if it improves on the fitness, then the new individual is accepted in the population (10).

$$X_i^t = \begin{cases} Y_{best} & \text{if } \Im\left(Y_{best}\right) < C_i^{t-1} \\ X_i^{t-1} & \text{otherwise} \end{cases} \tag{10}$$

If this individual improves on the overall best individual in the population, it then replaces the best individual in the population (11).

$$X_{best}^t = \begin{cases} Y_{best} & \text{if } \Im\left(Y_{best}\right) < C_{best}^t \\ X_{best}^{t-1} & \text{otherwise} \end{cases} \tag{11}$$

2.6 Iteration

Sequentially, incrementing *i*, the population counter by 1, another individual X_{i+1}^{t-1} is selected from the population, and it begins its own sampling towards the designated leader X_L^{t-1} from sections 2.2 - 2.5. It should be noted that the leader does not change during the evaluation of one migration.

2.7 Migrations

Once all the individuals have executed their sampling towards the designated leader, the migration counter *t* is incremented by 1. The individual iterator *i* is reset to 1 (the beginning of the population) and the loop in sections 2.2 - 2.6 is re-initiated.

2.8 2 Opt Local Search

The local search utilised in DSOMA is the 2 Opt local search algorithm. The reason as to why the 2 Opt local search was chosen, is that it is the simplest in the *k*-opt class of routines. As the DSOMA sampling occurs between two individuals in *k*-dimension, the local search refines the individual. This in turn provides a better probability to find a new leader after each jump sequence. The placement of the local search was refined heuristically during experimentation.

The complexity of this local search is $O\left(n^2\right)$. As local search makes up the majority of the complexity time of DSOMA, the overall computational complexity of DSOMA for a single migration is $O\left(n^3\right)$.

3 Lot-Streaming Problem

The lot-steaming problem considered in this paper is a subset of the generic flowshop scheduling problem. Whereas, in the permutative flowshop problem, each job *n* is processed by a single machine *m*, in a lot-streaming variant, each job is divided into smaller tasks called *lots* (l) [15]. Once the processing of a sub-lot on its preceding machine is completed, it can be transferred to the downstream machine immediately. However, all $l(j)$ sub-lots of job *j* should be processed continuously as no intermingling or exchanging is allowed. A separable sequence-dependent setup time is necessary for the first sub-lot of each job *j* before it can be processed on any machine *k* [16].

Two different cases of the problem are considered; the idling and the non-idling case. The idling case is a the simpler variant of the problem, where only the schedule of the lots are taken into consideration. A non-idling case on the other hand is more practical. A non-idle case arise when the machine is not allowed to be idle. This is beneficial, especially in the case where a number of machines are in operation, and resources such as electricity is wasted. Another practical situation is when expensive machinery is employed. Idling on such expensive equipment is often not desired. Other examples come from sectors where less expensive machinery is used but where machines cannot be easily stopped and restarted [19].

A detailed description of the lot-streaming problem is given in [17], [3] and [18].

3.1 Idling Case

The constraint in this case is that at any given time a machine can process only one sub-lot, and each sub-lot can only be assessed individually. Let the processing time of each sub-lot of job on machine m be $P(m, j)$, and the setup time of job j on machine m, after having processed job j is $s(m, j, j)$, which can also represent the setup time of job j if it is the first job to be proceeded in the machine. The objective is to find a sequence with the optimal sub-lot starting and completion times to minimise the makespan.

The permissible job permutation can be presented as $\pi = \{\pi_1, \pi_2, \ldots, \pi_n\}$, and the earliest start and completion time as $S(m, j, r)$ and $C(m, j, r)$, where r represents the specific sub-lot on job j being processed on machine m.

Initially, the first and third directive of equation (12) is used to calculate the earliest start time of the first sub-lot of the first job (π_1). The second and fourth directives calculate the completion times, making sure that preemption of jobs is not allowed. Equation (13) controls the earliest start time and the earliest completion time for the successive sub-lots of job π_1, which ensure that sub-lots of the same job are processed continuously.

Equation sets (14) and (15) are used for the calculations for the sub-lots of the following jobs in the π. When calculating the start time for the first sub-lot of a job in equation (14) , the completion time of the previous job on the current machine must be considered with the completion time of the sub-lot on the previous machine, and the setup time of the job on the current machine.

The makespan can be calculated using equation (16), which is essentially the completion time of the last sub-lot of the last job π_n on the last machine m [16].

$$\begin{array}{l}
S(1,\pi_1,1) = s(1,\pi_1,\pi_1) \\
C(1,\pi_1,1) = S(1,\pi_1,1) + P(1,\pi_1); \\
S(w,\pi_1,1) = \max\{C(w-1,\pi_1,1), s(w,\pi_1,\pi_1)\}, \\
C(w,\pi_1,1) = S(w,\pi_1,1) + +P(w,\pi_1), w = 2,3,\ldots,m
\end{array} \tag{12}$$

$$\begin{array}{l}
S(1,\pi_1,r) = C(1,\pi_1,r-1), \\
C(1,\pi_1,r) = S(1,\pi_1,r) + P(1,\pi_1), \;\; r = 2,3,\ldots,l(\pi_1); \\
S(w,\pi_1,r) = \max\{C(w-1,\pi_1,r), C(,\pi_1,r-1)\}, \\
C(w,\pi_1,r) = S(w,\pi_1,r) + P(w,\pi_1), \; r = 2,3,\ldots,l(\pi_1), \;\; w = 2,3,\ldots,m
\end{array} \tag{13}$$

$$\begin{array}{l}
S(1,\pi_1,1) = C(1,\pi_{i-1},l(\pi_{i-1})) + s(1,\pi_{i-1},\pi_i), \\
C(1,\pi_1,1) = S(1,\pi_1,1) + P(1,\pi_1), i = 2,3,\ldots,n; \\
S(w,\pi_1,1) = \max\left\{ \begin{array}{l} C(w-1,\pi_1,r), \\ C(w,\pi_{i-1},l(\pi_{i-1})) + s(w,\pi_{i-1},\pi_i) \end{array} \right\}, \\
C(w,\pi_1,1) = S(w,\pi_1,1) + P(w,\pi_i), \; i = 2,3,\ldots,n, \;\; w = 2,3,\ldots,m
\end{array} \tag{14}$$

$$\begin{aligned}
&S(1,\pi_1,r) = C(1,\pi_i,r-1)\,,\\
&C(1,\pi_1,r) = S(1,\pi_1,r) + P(1,\pi_1)\ i = 2,3,\ldots,n,\ \ r = 2,3,\ldots,l(\pi_i)\,;\\
&S(w,\pi_1,r) = \max\left\{C(w-1,\pi_1,r)\,,C(w,\pi_{i-1},r-1)\right\},\\
&C(w,\pi_1,r) = S(w,\pi_1,r) + P(w,\pi_i)\,,\ i = 2,3,\ldots,n,\ \ r = 2,3,\ldots,l(\pi_i)\,,\\
&w = 2,3,\ldots,m
\end{aligned} \tag{15}$$

$$C_{\max}(\pi) = C_T(m,\pi_n,l(\pi_n)) \tag{16}$$

3.2 Non-idling Case

For the non-idling case, the earliest start time for the first sub-lot is given in equations (17) and (18), where the start time is the maximum of the setup time of the job in the current machine, the completion time of the first sub-lot on the previous machine, and the difference between the completion time of the whole job on the previous machine and the total processing time of the whole job on the preceding machine except the last sub-lot. This ensures that there is the no idling time between two adjacent sub-lots. The last two directives of these equations calculate the completion time for the first job.

The subsequent processing time of the next job sequence are given in equations (19) and (20).

$$\begin{aligned}
&S(1,\pi_1,1) = s(1,\pi_i,\pi_i)\\
&C(1,\pi_1,l(\pi_1)) = S(1,\pi_1,1) + l(\pi_1)\times P(1,\pi_1)
\end{aligned} \tag{17}$$

$$\begin{aligned}
&S(w,\pi_1,1) = \max\left\{\begin{array}{l} s(w,\pi_1,\pi_1)\,,S(w-1,\pi_1,1) +\ p(w-1,\pi_1)\,,\\ C(k-1,\pi_1,l(\pi_1)) - (l(\pi_1)-1)\times P(1,\pi_1) \end{array}\right\},\\
&C(w,\pi_1,l(\pi_1)) = S(w,\pi_1,1) + l(\pi_1)\times P(w,\pi_i)\,,\ w = 2,3,\ldots,m
\end{aligned} \tag{18}$$

$$\begin{aligned}
&S(1,\pi_1,1) = C(1,\pi_{i-1},l(\pi_{i-1})) + s(1,\pi_{i-1},\pi_i)\,,\\
&C(1,\pi_1,l(\pi_1)) = S(1,\pi_1,1) + l(\pi_1)\times P(1,\pi_1)\,,\ i = 2,3,\ldots,n
\end{aligned} \tag{19}$$

$$\begin{aligned}
&S(w,\pi_1,1) = \max\left\{\begin{array}{l} S(w-1,\pi_i,1) + P(w-1,\pi_1)\,,\\ C(w-1,\pi_1,l(\pi_1)) - (l(\pi_1)-1)\times P(1,\pi_1)\,,\\ C(w-1,\pi_{i-1},l(\pi_{i-1})) + s(1,\pi_{i-1},\pi_i) \end{array}\right\},\\
&\qquad i = 2,3,\ldots,n,\ \ w = 2,3,\ldots,m\\
&C(w,\pi_1,l(\pi_1)) = S(w,\pi_1,1) + l(\pi_1)\times P(w,\pi_i)\,,\ i = 2,3,\ldots,n,\\
&\qquad w = 2,3,\ldots,m
\end{aligned} \tag{20}$$

The makespan for the non-idling case can be then calculated as equation (21).

$$C_{\max}(\pi) = C_T(m,\pi_n,l(\pi_n)) \tag{21}$$

The objective of the lot-streaming flow shop scheduling problem with makespan criterion is to find a permutation π^* in the set of all permutations $\prod$ can be given as in equation (22) [16].

$$C_{\max}(\pi^*) \leq C_{\max}(\pi)\,,\forall \pi \in \Pi \tag{22}$$

4 Data Sets Generation

In keeping with the theme of utilising the chaotic maps in lieu of PRNG, the data sets have been generated using two unique chaotic maps; the Lozi and the Dissipative map. Five unique sizes of data sets have been generated. They are from 10 job x 5 machine, 20 job x 10 machine, 50 job x 25 machine, 75 job x 30 machine and 100 job x 50 machine. There are 5 instances for each data set, therefore, in total 25 data set instances for each of the Lozi and Dissipative data sets.

In order to have *unique* data sets, each *instance* was initialised from a unique *start* position of the respective chaotic system. Additionally, the map was not allowed to be reinitialised. Two different maps were used in order to have more versatility in the data sets and to remove any particular bias.

The datasets are available at [5] for download.

5 Experimentation and Analysis

The main emphasis of this work is to show the improvement of applying the chaotic map to DSOMA. Therefore the experimentation compares the generic and chaos induced DSOMA (DSOMA$_C$) algorithms on the same problem sets, using the identical parameter settings.

The parameter settings for the DSOMA and DSOMA$_C$ are given in Table 2. The experiments was conducted on a MacBook Pro, Sandy Bridge 2.3 GHz Intel Core i7, 8 GB 1333 MHz DDR3 RAM.

Table 2 Parameter Settings

Parameters	Value
Population	40
Migration	20
J_{min}	20
Local Search	2 Opt

5.1 Lozi Generated Data Sets Results

The results of the Lozi generated datasets is given in Table 3 for the idling case and Table 4 for the non-idling case. A total of 5 experiments have been performed on each dataset, and the results are presented as the summation of the 5 experiments for the different problem instance sizes. Four different statistical analysis have been performed on the results. The minimum, maximum, average and execution time computation are presented. Additionally the *t*-test for the different data sets for the two different problems was conducted and is given in Tables 8 - 9. For the idling case in Table 3, DSOMA$_C$ obtains better average minimum (8684) and average (8919.84) value compared to 8899 and 9248.48 for DSOMA. Additionally, it obtains more minimum (5) and average values (5) for the individual data sets. In terms of the *t*-test, DSOMA$_C$ is significantly better than DSOMA on three data sets of 20 x 10, 50 x 25 and 75 x 30 in the idling case, and both algorithms are significantly equal on the remaining two data sets.

Table 3 Lozi Idling results

	DSOMA				**DSOMA$_C$**			
Instance	**Average**	**Min**	**Max**	**Time** (sec)	**Average**	**Min**	**Max**	**Time** (sec)
10 x 5	**429.8**	**383**	464	1.87	**429.8**	**383**	464	2.32
20 x 10	1778	1680	1885	45.03	**1727.8**	**1644**	1822	39.96
50 x 25	24872.2	24170	25404	504.7	**24111.0**	**23361**	24569	642.76
75 x 30	7584.6	7355	7714	1023.7	**7279.0**	**7145**	7324	1065.32
100 x 50	11577.8	10907	11362	2953.21	**11051.6**	**10887**	11216	2855.6
Average	9248.48	8899	9365.8	**905.7**	**8919.84**	**8684**	9079	921.19

For the non-idling case given in Table 4, the DSOMA and DSOMA$_C$ algorithms performs equally. DSOMA manages to obtain better average minimum (9863) and average (10173.24) values compared to 9878 an 10179.32 for DSOMA$_C$. In terms of individual instances, DSOMA and DSOMA$_C$ have almost identical performances; average (2) and minimum (3) for DSOMA compared to average (3) and minimum (3) for DSOMA$_C$. The *t*-test result for all the data instances shows that the two algorithms are significantly not different.

Table 4 Lozi Non-Idling results

	DSOMA				**DSOMA$_C$**			
Instance	**Average**	**Min**	**Max**	**Time** (sec)	**Average**	**Min**	**Max**	**Time** (sec)
10 x 5	513	431	557	2.22	**511.2**	**425**	552	2.46
20 x 10	**2107.6**	**1979**	2210	39.43	2108.0	**1979**	2210	42.86
50 x 25	**27852.4**	27164	28558	528.32	27889.6	**27100**	28407	536.45
75 x 30	8067.6	**7819**	8230	821.40	**8063.4**	7832	8259	846.43
100 x 50	12325.6	**11922**	12951	3688.20	**12324.4**	12054	12954	3865.30
Average	**10173.2**	**9863**	10501.2	**1015.91**	10179.3	9878	10476.4	1058.70

5.2 *Dissipative Generated Data Sets Results*

For the dissipative problem sets, the results are given in Table 5 for the idling case and Table 6 for the non-idling case. The *t*-test results for the different data sets is given in Tables 10 - 11. For the idling case, DSOMA$_C$ is better performing on all problem instances having the better average (10173.24) and minimum (12176) values. Analysing the *t*-test results, DSOMA$_C$ is significantly better than DSOMA on the two instances of 50 x 25 and 100 x 50.

Table 5 Dissipative Idling results

	DSOMA				**DSOMA$_C$**			
Instance	**Average**	**Min**	**Max**	**Time** (sec)	**Average**	**Min**	**Max**	**Time** (sec)
10 x 5	630.6	568	699	2.65	**626.2**	**566**	693	2.34
20 x 10	2049.8	1869	2279	39.54	**2040.6**	**1859**	2264	40.32
50 x 25	13095.0	11633	14392	612.32	**12789.8**	**11390**	14018	623.43
75 x 30	19256.2	18789	19740	967.43	**18998.8**	**18493**	19260	976.32
100 x 50	29868.8	28884	31163	3145.32	**29516.8**	**28572**	30824	3212.22
Average	12980.1	12348.6	13654.6	**953.45**	**12794.4**	**12176**	13411.8	970.93

Table 6 Dissipative Non-Idling results

	DSOMA				**DSOMA$_C$**			
Instance	**Average**	**Min**	**Max**	**Time** (sec)	**Average**	**Min**	**Max**	**Time** (sec)
10 x 5	**700.6**	**613**	762	2.01	701.2	**613**	762	1.97
20 x 10	2274.8	2146	2461	36.32	**2234.2**	**2120**	2416	37.24
50 x 25	14524.0	12897	15705	540.32	**14207.0**	**12816**	15455	544.23
75 x 30	21115.0	20855	21398	877.43	**20535.8**	**20244**	20903	892.43
100 x 50	32752.0	31742	33622	3302.32	**32137.6**	**31106**	33539	3331.4
Average	14273.3	13650.6	14789.6	**951.68**	**13963.2**	**13379.8**	14615	961.45

Table 7 Summarised Results

	DSOMA		DSOMA$_C$	
Instance	**Min**	**Average**	**Min**	**Average**
10 x 5	2	2	**4**	**3**
20 x 10	1	1	**4**	**3**
50 x 25	0	1	**4**	**3**
75 x 30	1	0	**3**	**4**
100 x 50	1	0	**3**	**4**
Total	5	4	**18**	**17**

Table 8 Paired *t*-test for the Lozi Idling results

Data set	*t*-value	*p*-value	$p < 0.05$	Result
10 x 5	-	-	-	-
20 x 10	8.28	0.001	Yes	DSOMA$_C$
50 x 25	14.32	0.0001	Yes	DSOMA$_C$
75 x 30	10.15	0.0005	Yes	DSOMA$_C$
100 x 50	2.22	0.089	No	Same

Table 9 Paired *t*-test for the Lozi Non-Idling results

Data set	*t*-value	*p*-value	$p < 0.05$	Result
10 x 5	1.15	0.313	No	Same
20 x 10	0.196	0.854	No	Same
50 x 25	0.518	0.613	No	Same
75 x 30	0.188	0.859	No	Same
100 x 50	0.026	0.98	No	Same

Table 10 Paired *t*-test for the Dissipative Idling results

Data set	*t*-value	*p*-value	$p < 0.05$	Result
10 x 5	1.53	0.2	No	Same
20 x 10	1.23	0.283	No	Same
50 x 25	5.42	0.005	Yes	DSOMA$_C$
75 x 30	1.49	0.209	No	Same
100 x 50	6.04	0.003	Yes	DSOMA$_C$

Table 11 Paired *t*-test for the Dissipative Non-Idling results

Data set	*t*-value	*p*-value	$p < 0.05$	Result
10 x 5	1	0.37	No	Same
20 x 10	6.38	0.003	Yes	DSOMA$_C$
50 x 25	4.125	0.014	Yes	DSOMA$_C$
75 x 30	7.96	0.0013	Yes	DSOMA$_C$
100 x 50	0.73	0.018	Yes	DSOMA$_C$

The non-idling case also affirms the better performance of the DSOMA$_C$ algorithm. Following the trend in the previous experiment, the chaos based DSOMA has better average minimum (13379.8) and average (13963.16) values and individual minimum (5) and average (4) values. From the *t*-test results DSOMA$_C$ is significantly better than DSOMA on four instances.

6 Conclusion

This research is based on ascertaining the effectiveness of applying the Lozi map to the DSOMA algorithm in order to solve the lot-streaming flowshop scheduling problem. In total, 50 data sets were generated using the Lozi and Dissipative maps, which were solved for both the idling and non-idling case. In total, 5 experimentation for each dataset was conducted, therefore a total of 500 individual experiments was conducted to validate the finding in this paper.

From the summarised results in Table 7, it is obvious that the DSOMA$_C$ algorithm is better performing than the generic DSOMA algorithm. Of all the valid compared parameters, DSOMA$_C$ obtained more minimum values, 18 compared to 5 and better average values, 17 compared to 4 for the DSOMA algorithm. This shows that the DSOMA$_C$ is more robust and is has a better probability of finding a better solution. In terms of *t*-test, DSOMA$_C$ is significantly better than DSOMA on nine data sets and equal in the remaining eleven data sets. This hypothesis raises the main issue regarding the importance of the PRNG in stochastic algorithms. Whereas, the defining argument during the past decades has been the application of better crossover and mutation routines, the generator for selection and mutation, the PRNG now gains more prominence.

The downside of the experiment invariably has been the execution time, where there was a slight increase for the chaos version. However, this can be improved significantly through utilising the GPU for computing. This will be directed as a further aspect of this research, alongside the application of different chaotic maps as generators. The implementation of different chaos maps is important in the deduction aspect of which maps are compatible for the different problem classes.

Acknowledgements. This work was supported by the Technology Agency of the Czech Republic under the Project TE01020197 and the Internal Grant Agency of Tomas Bata University under the project No. IGA/FAI/2013/012.

References

1. Alatas, B., Akin, E., Ozer, A.: Chaos embedded particle swarm optimization algorithms. Chaos, Solitons and Fractals 40(4), 1715–1734 (2009)
2. Caponetto, R., Fortuna, L., Fazzino, S., Xibilia, M.: Chaotic sequences to improve the performance of evolutionary algorithms. IEEE Transactions on Evolutionary Computation 7(3), 289–304 (2003)
3. Chang, J.L., Gong, D.W., Ma, X.P.: A heuristic genetic algorithm for no-wait flowshop scheduling problem. Journal of China University of Mining and Technology 17(4), 582–586 (2007)
4. Davendra, D.: Chaotic attributes and permutative optimization. Ph.D. thesis, Tomas Bata University in Zlin (2009)
5. Davendra, D.: Flowshop lot-streaming problem data sets (2012), http://mrl.cs.vsb.cz/people/davendra/research.html
6. Davendra, D., Bialic-Davendra, M.: Scheduling flow shops with blocking using a discrete self-organising migrating algorithm. International Journal of Production Research 51(8), 2200–2218 (2013), doi:10.1080/00207543.2012.711968
7. Davendra, D., Zelinka, I., Senkerik, R.: Chaos driven evolutionary algorithm for the traveling salesman problem. In: Davendra, D. (ed.) Traveling Salesman Problem, Theory and Applications, pp. 55–70. InTech Publishing, Croatia (2010)
8. Davendra, D., Zelinka, I., Bialic-Davendra, M., Senkerik, R., Jasek, R.: Discrete self-organising migrating algorithm for flow-shop scheduling with no-wait makespan. Mathematical and Computer Modelling 57(12), 100–110 (2013), doi:10.1016/j.mcm.2011.05.029
9. Davendra, D., Zelinka, I., Senkerik, R.: Chaos driven evolutionary algorithms for the task of pid control. Computers & Mathematics with Applications 60(4), 1088–1104 (2010)
10. Hilborn, R.: Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers. OUP Oxford (2000)
11. Lu, Y., Zhou, J., Qin, H., Wang, Y., Zhang, Y.: Chaotic differential evolution methods for dynamic economic dispatch with valve-point effects. Engineering Applications of Artificial Intelligence 24(2), 378–387 (2011)
12. Matsumoto, M.: Mersenne twister webpage (2012), http://www.math.sci.hiroshima-u.ac.jp/ m-mat/ MT/ARTICLES/earticles.html
13. Matsumoto, M., Nishimura, T.: Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. ACM Transaction on Modeling and Computer Simulation 8(1), 3–30 (1998)
14. Ozer, A.: Cide: Chaotically initialized differential evolution. Expert Systems with Applications 37(6), 4632–4641 (2010)
15. Pan, Q.K., Fatih Tasgetiren, M., Suganthan, P.N., Chua, T.J.: A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. Inf. Sci. 181(12), 2455–2468 (2011), doi:10.1016/j.ins.2009.12.025
16. Pan, Q.K., Ruiz, R.: An estimation of distribution algorithm for lot-streaming flow shop problems with setup times. Omega 40(2), 166–180 (2012)
17. Potts, C.N., Baker, K.R.: Flow shop scheduling with lot streaming. Oper. Res. Lett. 8(6), 297–303 (1989)
18. Sarin, S.C., Jaiprakash, P.: Flow Shop Lot Streaming. Springer, Berlin (2007)

19. Yavuz, M.: Fuzzy lead time management. In: Kahraman, C., Yavuz, M. (eds.) Production Engineering and Management under Fuzziness. STUDFUZZ, vol. 252, pp. 77–94. Springer, Heidelberg (2010)
20. Yuan, X., Cao, B., Yang, B., Yuan, Y.: Hydrothermal scheduling using chaotic hybrid differential evolution. Energy Conversion and Management 49(12), 3627–3633 (2008)
21. Zelinka, I.: Soma - self organizing migrating algorithm. In: Onwubolu, G., Babu, B. (eds.) New Optimization Techniques in Engineering. Springer, Germany (2004)
22. Zuo, X., Fan, Y.: A chaos search immune algorithm with its application to neuro-fuzzy controller design. Chaos, Solitons and Fractals 30(1), 94–109 (2006)

Hidden Periodicity – Chaos Dependance on Numerical Precision

Ivan Zelinka, Mohammed Chadli, Donald Davendra, Roman Senkerik,
Michal Pluhacek, and Jouni Lampinen

Abstract. Deterministic chaos has been observed in many systems and seems to be random-like for external observer. Chaos, especially of discrete systems, has been used on numerous occasions in place of random number generators in so called evolutionary algorithms. When compared to random generators, chaotic systems generate values via so called map function that is deterministic and thus, the next value can be calculated, i.e. between elements of random series is no deterministic relation, while in the case of chaotic system it is. Despite this fact, the very often use of chaotic generators improves the performance of evolutionary algorithms. In this paper, we discuss the behavior of two selected chaotic system (logistic map and Lozi system) with dependance on numerical precision and show that numerical precision causes the appearance of many periodic orbits and explain reason why it is happens.

1 Introduction

The term "chaos" covers a rather broad class of phenomena, whose behavior may seem erratic, chaotic at first glance. Often, this term is used to denote

Ivan Zelinka · Donald Davendra
VSB-Technical University of Ostrava, 17. listopadu 15 708 33, Ostrava-Poruba, Czech Republic
e-mail: ivan.zelinka@vsb.cz

Roman Senkerik · Michal Pluhacek
Faculty of Applied Informatics, Tomas Bata University in Zlin
e-mail: senkerik@fai.utb.cz

Mohammed Chadli
Laboratory of Modeling, Information & Systems (MIS), University of Picardie Jules Verne (UPJV), 33 rue Saint-Leu, 80039 Amiens Cedex 1, France
e-mail: mchadli@u-picardie.fr

Jouni Lampinen
University of Vaasa, Department of Computer Science, P.O. Box 700, FI-65101 Vaasa, Finland
e-mail: antlam@uwasa.fi

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 47–59.
DOI: 10.1007/978-3-319-00542-3_7

phenomena, which are of a purely stochastic nature, such as the motion of molecules in a vessel with gas and the like. The discovery of the phenomenon of deterministic chaos brought about the need to identify manifestations of this phenomenon also in experimental data. Deterministically chaotic systems are necessarily nonlinear, and conventional statistical procedures, which are mostly linear, are insufficient for their analysis. If the output of a deterministically chaotic system is subjected to linear methods, such signal will appear as the result of a random process. Examples include the Fourier spectral analysis, which will disclose nonzero amplitudes at all frequencies in a chaotic system, and so chaos can be easily mistaken for random noise.

Deterministic chaos has been used during last decade as a pseudorandom number generator. For example in [8] is discussed possibility of generation of random or pseudorandom numbers by use of the ultra weak multidimensional coupling of p 1-dimensional dynamical systems. In the paper [1] is done deep investigation on logistic map as on possible pseudo-random number generator and is compared with contemporary pseudo-random number generators. A comparison of logistic map results is made with conventional methods of generating pseudorandom numbers. The approach used to determine the number, delay, and period of the orbits of the logistic map at varying degrees of precision (3 to 23 bits). Logistic map, we are using here, was also used in [2] like chaos-based true random number generator embedded in reconfigurable switched-capacitor hardware. Another paper [9] proposed an algorithm of generating pseudorandom number generator, which is called (couple map lattice based on discrete chaotic iteration) and combine the couple map lattice and chaotic iteration. Authors also tested this algorithm in NIST 800-22 statistical test suits and is used in image encryption. In [10] authors exploit interesting properties of chaotic systems to design a random bit generator, called CCCBG, in which two chaotic systems are cross-coupled with each other. For evaluation of the bit streams generated by the CCCBG, the four basic tests are performed: monobit test, serial test, auto-correlation, Poker test. Also the most stringent tests of randomness: the NIST suite tests have been used. Another research is done in [11]. A new binary stream-cipher algorithm based on dual one-dimensional chaotic maps is proposed in this paper with statistic proprieties showing that the sequence is of high randomness. Similar studies are done in [3] (discussing whether chaos work better than noise), [4] (proposes an experimental analysis on the convergence of evolutionary algorithms wit use of logistic map, Gauss map, Lozi map amongst the others), [5] (the Chen chaotic system is proposed as a pseudorandom sequence generator in this research article) and [6], (the effect of a weak random additive noise in a linear chain of N locally coupled logistic maps at the edge of chaos are investigated there. Maps tend to synchronize for a strong enough coupling, but if a weak noise is added, very intermittent fluctuations are observed and investigated with dependance on weak noise and map coupling). All mentioned papers less or more discuss randomness, chaos and its mutual intersections and influences.

More application oriented papers are for example [13] - [14] that shows how chaotic dynamics can be used instead of random number generator in evolutionary algorithms. It is numerically shown that algorithms powered by chaotic system has better performance than algorithms powered by random number generator. In fact, more comprehensive study of mutual intersections between chaos and evolution can be found in [7] and chaos and randomness in [8]. Chaos, especially of discrete systems, has been many times used instead of random number generator in so called evolutionary algorithms, [13] - [14], [7]. Comparing to random generators, chaotic systems generate values via so called map function that is deterministic and thus, next value can be calculated, i.e. between elements of random series there is no deterministic relation, while in the case of chaotic system i.e. despite this fact, the very often use of chaotic "random" generator, improves the performance of evolutionary algorithms. In this paper we discuss the behavior of selected chaotic system with dependance on numerical precision and show that numerical precision cause appearance of many periodic orbits.

In our experiments, the logistic map Eq. 1, (example of chaotic behavior is at Fig. 1) and Lozi system, Eq. 2, has been used. The main aim of this paper is to show how dependent the behavior of the two selected discrete deterministic chaotic systems on numerical precision and discuss the possibility as to whether we really need random number generators for evolutionary algorithms.

$$x_{n+1} = Ax_n\left(1 - x_n\right) \tag{1}$$

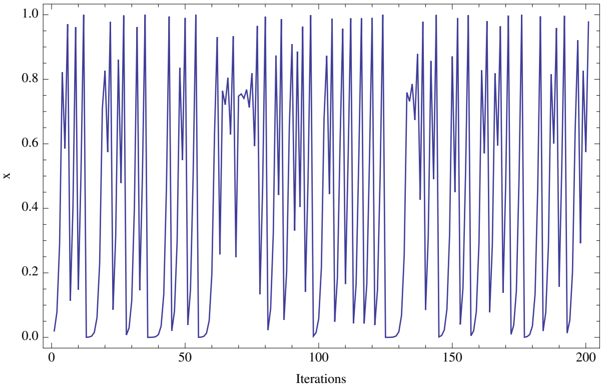


Fig. 1 Chaotic dynamics of logistic map, no restrictions has been given on numerical precision

2 Experiment Design

Similarly as in the [1] is the core object of this study also logistic map (Eq. 1) and Lozi system (Eq. 2) too. Comparing to [1] where period of the orbits of the logistic map were investigated at varying degrees of precision (3 to 23 bits), precision is in this paper is not used on binary level, but on decimal, i.e. the term precision means here how many numbers behind decimal point is used. For example precision 2 means $n.nn$, precision 3 $n.nnn$ etc. Contrary to [1], both chaotic systems discussed here are not studied for their pseudorandomness properties. Appearance of periodicity and its length is studied on the other side, with future use inside evolutionary algorithms instead of pseudo-random generators. Simply, both chaotic systems are used to generate n periodic series with different length.

For all experiments and results reported here has been used MacBook Pro with Intel Core 2 Duo 2.8 GHz and the Mathematica 9.0 software. Both chaotic systems has been used in a few levels of investigation, i.e.

- Identification of n periodic orbits with dependance on numerical precision.
- Global view on logistic map behavior with fixed control parameters.
- Global view on logistic map behavior for various parameter of $A \in [3.4-4]$, logistic map has been iterated for 1000, 10 000, 100 000 iterations.
- Analyze why and when is n periodic behavior generated for both chaotic systems.
- Detail analysis of logistic map behavior for parameter $A = 4$, numerical precision $\in [1, 20]$, initial conditions $x_{start} \in [0.01, 0.99]$ (x was incremented by increment $= 0.01$) and 1 000 000 iterations for each combination of this parameters.
- Detail analysis of Lozi system (Fig. 2. Eq. 2) behavior for parameter $A = 1.75, B = 0.5$, numerical precision $\in [1, 10]$, initial conditions $x_{start} \in [-0.1, 1]$, $y_{start} \in [-0.1, 1]$ (x was incremented by increment $= 0.1$) and 1 000 000 iterations for each combination of this parameters.

Identification of period length was done so that each dataset generated by each experiment was checked whether each value is in dataset only once (no periodicity in the range of used iterations was observed) or more time, that indicate period which certain length has been found. Results are discussed in the next section and summarized in the section Conclusion.

$$\begin{aligned} &x_{n+1} = 1 - A\,|x_n| + By_n \\ &y_{n+1} = x_n \\ &A = 1.7 \\ &B = 0.5 \end{aligned} \tag{2}$$

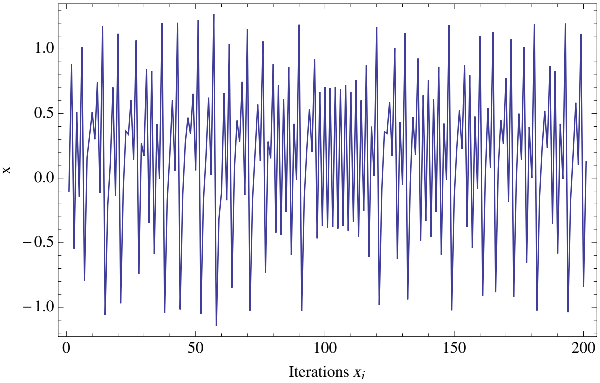


Fig. 2 Chaotic dynamics of Lozi system, no restrictions has been given on numerical precision

3 Results

In this section are described results from both chaotic systems. The first is discussed logistic map and then Lozi system.

3.1 Logistic Map

As described in the previous section, dynamics of the logistic map has been investigated and discussed when, how and what n periodical orbits can be observed in this dynamics and at the end of this paper is also discussed whether the random number generator is really so important for evolutionary algorithms.

Here a few simple demonstrations has been made on the logistic map. Periodic behavior of logistic map, with precision 2 and $x_{start} = 0.02$, (Fig. 3) and precision $= 4$ (Fig. 4) has been generated for example. As clearly visible, different precision caused different periodic data series. Also, the initial value x_{start} has an impact on the appearance of n periodic series, as was observed during experiments.In this part, it has been numerically demonstrated that different precision cause different n periodic series.

To get a more comprehensive picture, we have extend previous experiments so that the behavior of logistic map has been depicted in the Fig. 5. In this figure we can easily see, that for different precisions, the logistic map produces many periodical series. Color on this figure represent the amplitude of the logistic map under different setting.

The next step was focused on the simulation, where the behavior of logistic map has been investigated for different setting of parameter A and numerical precisions. Visualization results are on Fig. 6. Black squares represent settings of logistic map producing periodic behavior, while white are

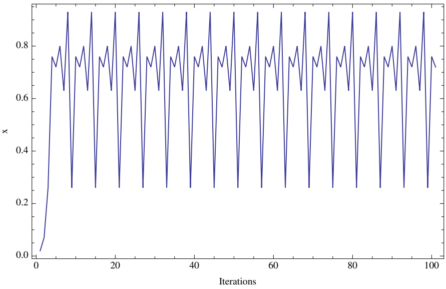


Fig. 3 Periodic behavior of logistic map with precision 2 and $x_{start} = 0.02$

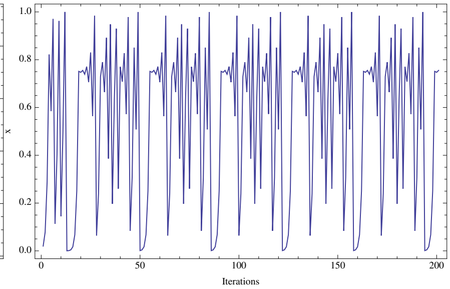


Fig. 4 Periodic behavior of logistic map with precision 4 and $x_{start} = 0.02$

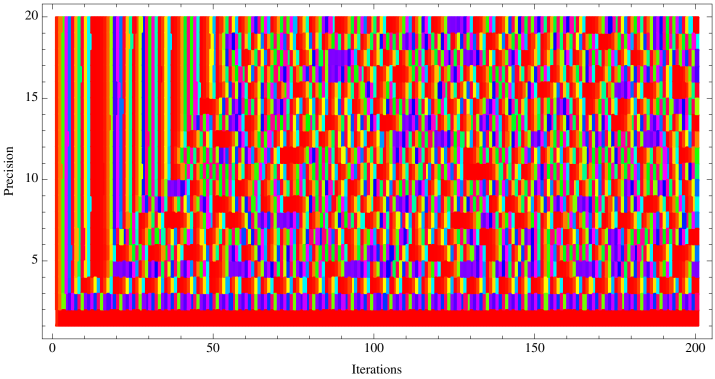


Fig. 5 Numerical demonstration: global view on behavior of logistic map with dependance on precision and iterations

settings for which, in the range of defined iterations, has not been observed periodicity. It is obvious that with increasing number of iterations, number of periodic series increase too and is logically evident, however on Fig. 7 - 8 is demonstrated why it happens.

On Fig. 7 is depicted the Cobweb diagram of the logistic map when precision is set to 1. This numerical precision of the map function is stepwise and thus projection from x_n to x_{n+1} is limited to finite number of possibilities, i.e. no chaotic series can be observed there. On Fig. 7, is depicted the time series with 50 iterations (see also Fig. 9). It is obviously visible when compared with Fig. 8, where for the same conditions, but with high precision, chaotic series has been generated.

Even in the case of higher precision, the mapping function is still discontinuous. The question is thus for what precision, x_{start} and parameter A, we can observe n periodic series. Extended version of previous experiments has been done. Parameter A has been set to 4, x_{start} changed in the interval [0, 1] by increments of 0.01. For each setting has been set precision for value from 1 to 20 (incremented by 1) and has been done 1 000 000 iterations. In the

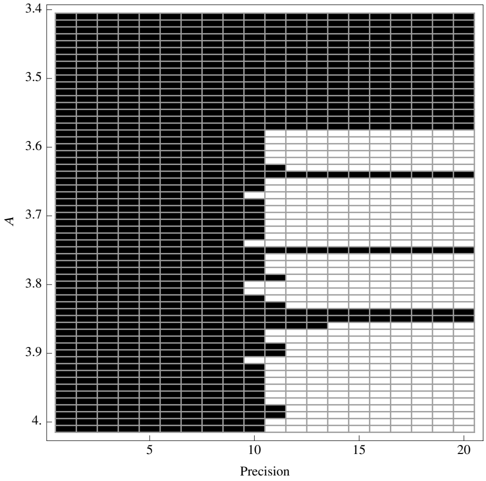


Fig. 6 Numerical demonstration: periodicity and ”chaos”. Logistic map has been iterated for 100 000 iterations, $x_{start} = .02$. Black squares represent settings of logistic map producing periodic behavior. With increasing number of iterations number of black squares increase.

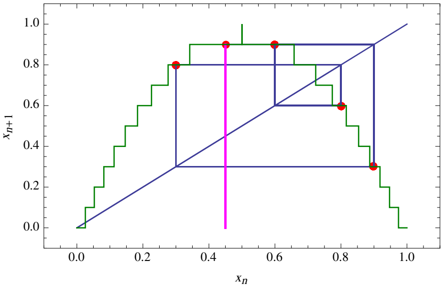


Fig. 7 Cobweb diagram: impact of low precision = 1 (one decimal behind zero) on shape of the mapping function (green stepwise curve). Four periodic orbit can be easily observed, see Fig. 9.

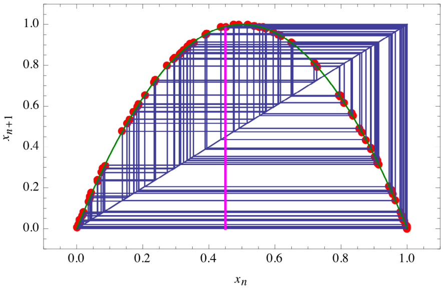


Fig. 8 Cobweb diagram: chaotic dynamics with of logistic map with high precision. Initial conditions of both Cobweb diagrams were the same. Different behavior is caused by different level of precision.

obtained results, a search for periodical series has been performed, see Fig. 10 - 11 and Table 1. The maximal period has been 922 744 and the shortest is 1 (i.e. stable state - constant value).

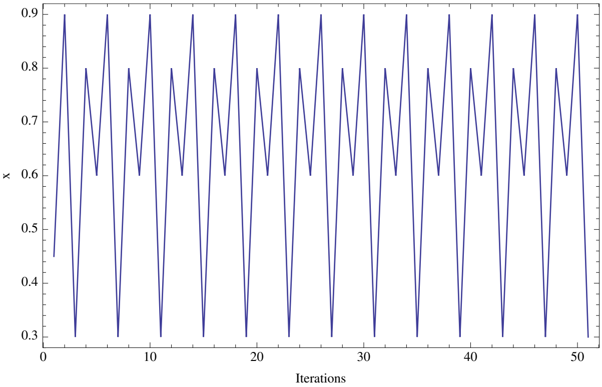


Fig. 9 Dynamic from Fig. 7 in time

Table 1 Periodical time series generated by logistic map. Value 0 means that no periodicity has been observed (in the range of used iterations).

Precision	Minimal Period	Maximal Period
1	4	4
2	2	10
3	10	29
4	15	36
5	67	170
6	143	481
7	421	758
8	1030	4514
9	2277	11227
10	2948	35200
11	9668	57639
12	65837	489154
13	518694	518694
14	75594	316645
15	1	0
16	1	0
17	264446	264336
18	18491	18491
19	46854	46854
20	70767	922744

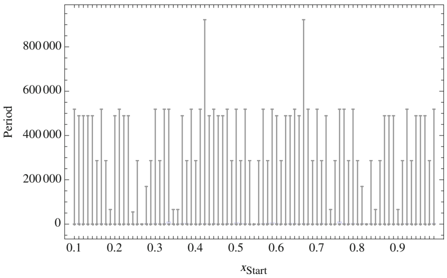


Fig. 10 Dependance of period on initial value x_{start}

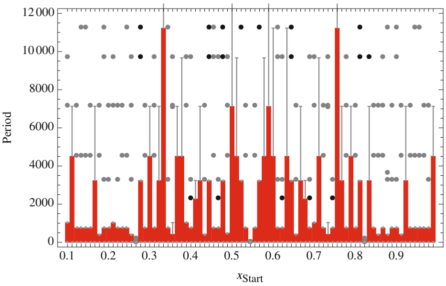


Fig. 11 Dependance of period on initial value x_{start} - detail view with outliers

3.2 Lozi System

Another system, that was used for demonstration of ideas proposed here is so called Lozi system, see Eq. 2. In the case of the Lozi system precision has been changed from 1 to 10, for each precision value was done 1 000 000 iterations and analyzed whether period can be observed under such experimental setting. In the Lozi system was done search for period under parameters $A = 1.75$ and $B = 0.5$.

Results are recorded in Tab. 2 and a few examples visualized on Fig. 12 and Fig. 13, compare with dynamics of Lozi system without any restrictions on numerical precision, see Fig.2.

When compare Tab. 1 and 2, it is visible, that in the case of Lozi system is sensitivity of appearance of periodical dynamics (with dependance on numerical precision) much bigger than in the case of Logistic map. Results from Tab. 2 are also visualized on Fig. 14.

Table 2 Periodical time series generated by Lozi system, see Fig. 14

Precision	Minimal Period	Maximal Period
1	6	11
2	22	50
3	701	919
4	191	3267
5	1351	17765
6	2806	91416
7	267868	482539
8	253954	989400
9	915344	970455
10	676598	885777

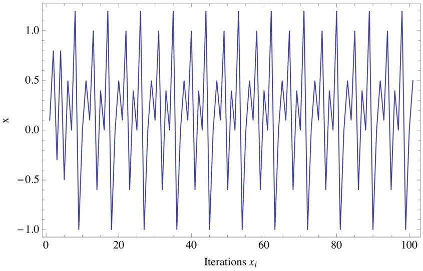


Fig. 12 Period of Lozi system, Eq. 2, for precision 1 and $x_{start} = 0.1$ and $y_{start} = 0.1$

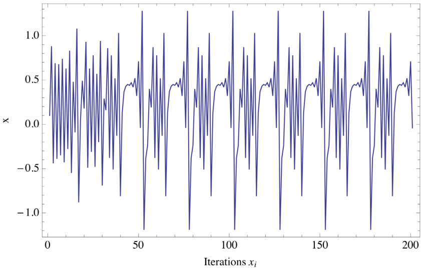


Fig. 13 Period of Lozi system, Eq. 2, for precision 2 and $x_{start} = 0.1$ and $y_{start} = 0.1$

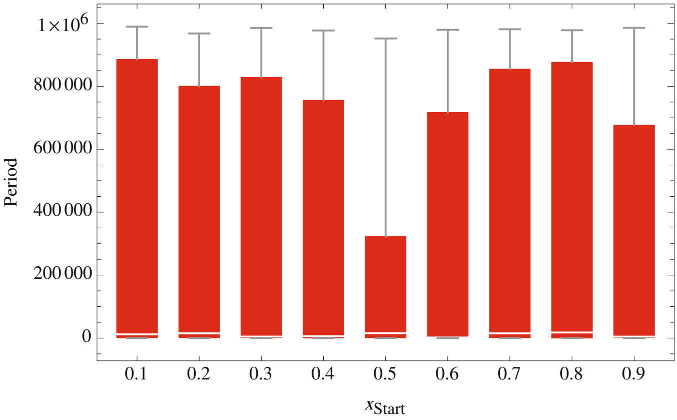


Fig. 14 Period of Lozi system generated by finite numerical precision,see Eq. 2, for different x_{start}, see Tab. 2

4 Conclusion

The main motivation of the research reported in this paper is whether it is possible to generate n periodic series by means of chaotic systems. All previous experiments has demonstrated an impact of numerical precision on logistic map and Lozi system dynamics.

During experiments, it has been observed that periodicity depends on initial conditions and precision (logistic map: Fig. 3 - 4, Tab. 1, Lozi system: Fig. 12 - 13, Tab. 2). It was also demonstrated on Fig. 6 for logistic map. Impact of precision on mapping function is explained on Fig. 7 - 8. Mapping function is due to numerical precision stepwise and dependance x_{n+1} on x_n is not continuous, like on Fig. 8, but discrete. This is the main reason why, when precision is low, we can observe low periodic time series.

At the end, the second set of experiments, using Lozi system (Eq. 2), was done. Results about observed period as well as its visualizations are in Tab.

2 and Fig. 12 - 14. It is clearly visible that numerical precision has also definitely clear impact on periodicity of chaotic systems.

Intensive experiments, focused on numerical investigation of under what conditions are n periodic orbits (series) generated and of what length have been conducted. It has been clearly demonstrated that on computers in fact, deterministic chaos does not exist at all, because computers work with finite numerical precision. A more important issue, that arises during this investigation was, that despite the fact that discrete chaotic systems contain deterministic map function (i.e. **no** pure randomness or pseudo-randomness is present) and its use inside evolutionary algorithms instead of pseudo-random number generators, improves very often its performance, see [13] - [14], [16] - [21].

Next research shall be focused on whether do exist causal relation between period length, precision and level of chaos, that can be measured by Lyapunov exponent. For those purposes can be used for example systems like already discussed logistic map (Eq. 1), Lozi system (Eq. 2) and other systems like Sine map amongst the others. Close relation between chaotic domains and Lyapunov exponent will be used to investigate impact of different level of chaos on evolutionary algorithms performance.

When summarize fact that chaos can improve performance of evolutionary algorithms, see for example [4] - [15], and is generated with finite precision (thus n periodical dynamics can be observed), it arise another question, whether evolutionary algorithms really require pseudo-random number generators or just deterministic periodical series. This is now under investigation.

Acknowledgements. The following two grants are acknowledged for the financial support provided for this research: Grant Agency of the Czech Republic - GACR P103/13/08195S, by the Development of human resources in research and development of latest soft computing methods and their application in practice project, reg. no. CZ.1.07/2.3.00/20.0072 funded by Operational Programme Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic, partially supported by Grant of SGS No. SP2013/114, VŠB - Technical University of Ostrava, Czech Republic, and by European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089.

References

1. Persohn, K.J., Povinelli, R.J.: Analyzing logistic map pseudorandom number generators for periodicity induced by finite precision floating-point representation. Chaos, Solitons and Fractals 45, 238–245 (2012)
2. Drutarovsky, M., Galajda, P.: A robust chaos-based true random number generator embedded in reconfigurable switched-capacitor hardware. In: 17th International Conference Radioelektronika, Brno, Czech Republic, April 24-25, vol. 1, 2, pp. 29–34 (2007)
3. Bucolo, M., Caponetto, R., Fortuna, L., Frasca, M., Rizzo, A.: Does chaos work better than noise? IEEE Circuits and Systems Magazine 2(3), 4–19 (2002)

4. Caponetto, R., Fortuna, L., Fazzino, S., Xibilia, M.: Chaotic sequences to improve the performance of evolutionary algorithms. IEEE Trans. Evol. Comput. 7(3), 289–304 (2003)
5. Hu, H., Liu, L., Ding, N.: Pseudorandom sequence generator based on the Chen chaotic system. Computer Physics Communications 184(3), 765–768 (2013), doi:10.1016/j.cpc.2012.11.017
6. Pluchino, A., Rapisarda, A., Tsallis, C.: Noise, synchrony, and correlations at the edge of chaos. Physical Review E 87(2) (2013), doi:10.1103/PhysRevE.87.022910
7. Zelinka, I., Celikovsky, S., Richter, H., Chen, G.: Evolutionary Algorithms and Chaotic Systems, p. 550. Springer, Germany (2010)
8. Lozi, R.: Emergence Of Randomness From Chaos. International Journal of Bifurcation and Chaos 22(2), 1250021 (2012), doi:10.1142/S0218127412500216
9. Wang, X.-Y., Qin, X.: A new pseudo-random number generator based on CML and chaotic iteration. International Journal of Nonlinear Dynamics and Chaos in Engineering Systems 70(2), 1589–1592 (2012), doi:10.1007/s11071-012-0558-0
10. Pareek, N.K., Patidar, V., Sud, K.K.: A Random Bit Generator Using Chaotic Maps. International Journal of Network Security 10(1), 32–38 (2010)
11. Wang, X.-Y., Yang, L.: Design Of Pseudo-Random Bit Generator Based On Chaotic Maps. International Journal of Modern Physics B 26(32), 1250208 (9 pages) (2012), doi:10.1142/S0217979212502086
12. Davendra, D., Zelinka, I., Senkerik, R.: Chaos driven evolutionary algorithms for the task of PID control. Computers and Mathematics with Applications 60(4), 1088–1104 (2010)
13. Pluhacek, M., Senkerik, R., Davendra, D., Kominkova Oplatkova, Z.: On the Behaviour and Performance of Chaos Driven PSO Algorithm with Inertia Weight. Computers and Mathematics with Applications (in print), ISSN 0898-1221
14. Pluhacek, M., Budikova, V., Senkerik, R., Kominkova Oplatkova, Z., Zelinka, I.: On the Performance of Enhanced PSO Algorithm with Lozi Chaotic Map. In: Application of Modern Methods of Prediction, Modeling and Analysis of Nonlinear Systems. SCI, vol. 1, p. 18. Springer, Heidelberg (November 2012) (accepted for publication) ISSN: 1860-949X
15. Senkerik, R., Davendra, D., Zelinka, I., Oplatkova, Z., Pluhacek, M.: Optimization of the batch reactor by means of chaos driven differential evolution. In: Snasel, V., Abraham, A., Corchado, E.S. (eds.) SOCO Models in Industrial & Environmental Appl. AISC, vol. 188, pp. 93–102. Springer, Heidelberg (2013)
16. Yang, M., Guan, J., Cai, Z., Wang, L.: Self-adapting differential evolution algorithm with chaos random for global numerical optimization. In: Cai, Z., Hu, C., Kang, Z., Liu, Y. (eds.) ISICA 2010. LNCS, vol. 6382, pp. 112–122. Springer, Heidelberg (2010)
17. Coelho, L., Mariani, V.: Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect. IEEE Transactions on Power Systems 21(2), 989–996 (2006), doi:10.1109/TPWRS.2006.873410
18. Hu, G.-W.: Chaos-differential evolution for multiple sequence alignment. In: 3rd International Symposium on Intelligent Information Technology Application, Nanchang, Peoples R China, vol. 2, pp. 556–558., doi:10.1109/IITA.2009.511

19. Zhao, Q., Ren, J., Zhang, Z., Duan, F.: Immune co-evolution algorithm based on chaotic optimization. In: Workshop on Intelligent Information Technology Application (IITA 2007), Zhang Jiajie, Peoples R China, pp. 149–152.
20. Liua, B., Wanga, L., Jina, Y.-H., Tangb, F., Huanga, D.-X.: Improved particle swarm optimization combined with chaos. Chaos, Solitons & Fractals 25(5), 1261–1271 (2005)
21. Gandomi, A., Yun, G., Yang, X., Talatahari, S.: Chaos-enhanced accelerated particle swarm optimization. Communications In Nonlinear Science and Numerical Simulation 18(2), 327–340 (2013), doi:10.1016/j.cnsns.2012.07.017

Do Evolutionary Algorithms Indeed Require Random Numbers? Extended Study

Ivan Zelinka, Mohammed Chadli, Donald Davendra, Roman Senkerik, Michal Pluhacek, and Jouni Lampinen

Abstract. An inherent part of evolutionary algorithms, that are based on Darwin theory of evolution and Mendel theory of genetic heritage, are random processes. In this participation, we discuss whether are random processes really needed in evolutionary algorithms. We use n periodic deterministic processes instead of random number generators and compare performance of evolutionary algorithms powered by those processes and by pseudo-random number generators. Deterministic processes used in this participation are based on deterministic chaos and are used to generate periodical series with different length. Results presented here are numerical demonstration rather than mathematical proofs. We propose that a certain class of deterministic processes can be used instead of random number generators without lowering of evolutionary algorithms performance.

Ivan Zelinka · Donald Davendra
VSB-Technical University of Ostrava, 17. listopadu 15 708 33, Ostrava-Poruba, Czech Republic
e-mail: ivan.zelinka@vsb.cz

Roman Senkerik · Michal Pluhacek
Faculty of Applied Informatics, Tomas Bata University in Zlin
e-mail: senkerik@fai.utb.cz

Mohammed Chadli
Laboratory of Modeling, Information and Systems (MIS), University of Picardie Jules Verne (UPJV), 33 rue Saint-Leu, 80039 Amiens Cedex 1, France
e-mail: mchadli@u-picardie.fr

Jouni Lampinen
University of Vaasa, Department of Computer Science, P.O. Box 700, FI-65101 Vaasa, Finland
e-mail: antlam@uwasa.fi

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 61–75.
DOI: 10.1007/978-3-319-00542-3_8

1 Introduction

The term “chaos” covers a rather broad class of phenomena whose behavior may seem erratic, chaotic at first glance. Often, this term is used to denote phenomena which are of a purely stochastic nature, such as the motion of molecules in a vessel with gas and the like. The discovery of the phenomenon of deterministic chaos brought about the need to identify manifestations of this phenomenon also in experimental data. Deterministically chaotic systems are necessarily nonlinear, and conventional statistical procedures, which are mostly linear, are insufficient for their analysis. If the output of a deterministically chaotic system is subjected to linear methods, such signals will appear as the result of a random process. Examples include the Fourier spectral analysis, which will disclose nonzero amplitudes at all frequencies in a chaotic system, and so chaos can be easily mistaken for random noise. Till now, chaos was observed in many of various systems (including evolutionary one) and in the last few years is also used to replace pseudo-number generators (PRGNs) in evolutionary algorithms (EAs). Lets mention for example research papers like [1] (a comprehensive overview of mutual intersection between EAs and chaos is discussed here), [14], [15], [23], [25], discussing use of deterministic chaos inside particle swarm algorithm instead of PRGNs, [2] - [5] investigating relations between chaos and randomness or the latest one [6] and [7] using chaos with EAs in applications. Another papers using deterministic chaos inside EAs are for example [16], [17], [18], [20], [21], discussing combination of differential evolution and chaotic systems, [19] (chaotic optimization and immune co-evolution algorithm) and [22], [24] that also use chaotic dynamics in evolutionary algorithms.

This publication is focused on use of deterministic chaos to generate n periodic deterministic series, that are used inside evolutionary algorithms instead of pseudo-random number generator.

2 Motivation

The motivation of the proposed experiments here is quite simple. For a long time, various PRNGs were used inside evolutionary algorithms. During the last few years, deterministic chaos systems (DCHS) instead of PRNGs have been used. As was demonstrated in [14]-[15], very often the performance of EAs using DCHS better or fully comparable with EAs using PRNGs. See for example [14]. Used EAs (we do not discuss here the special cases, modified for special experiments) of different kinds, such as genetic algorithms [12], differential evolution [9], particle swarm [13], SOMA [8], scatter search [10], evolutionary strategies [11], etc. do not analyze whether used pseudo-random numbers are really random ones. Random numbers are simply used. On the other side, as demonstrated in mentioned references, EAs with DCHS gives the same or often better performance. Difference between series from PRNGs and DCHS is that in the case of DCHS, one can easily reconstruct/calculate whole series generated by DCHS from one point. In the PRNGs it is impossible. Because DCHS generate periodical series (thanks to final numerical precision) it is obvious that EAs performance shall be from a certain numerical precision of DCHS comparable with performance of classical EAs. This is the main aim of this paper - check whether PRNGs can be replaced by deterministic periodical dynamics.

3 Experiment Design

Our experiments has been set so that periodical deterministic time series based on deterministic chaos generators (see for example Fig. 1) and Eq. 2, were used instead of PRNGs. Based on the fact that numerical precision has impact on existence of periodicity in deterministic chaos, we have selected logistic equation, Eq. 2, and data series generated by this equation for numerical precisions from interval [1, 13] (numbers behind decimal point) with setting $A = 4$, see Tab. 2 which shows only maximal period for current setting. Algorithms selected for our experiments were SOMA [8] and differential evolution (DERand1Bin and DELocalToBest) [9] . Setting of all three algorithms is in Tab. 1. Based on these setting and algorithm architecture, it is easy to calculate how many times has been used periodic data series (PDS) in EAs generated by DCHS. The total cost function evaluation was for SOMA maximally 172 727 and for both DE versions 130 000 times. Tab. 2 summarize the many precision levels were PDS were repeatedly used. All experiments were done in Mathematica 9, on MacBook Pro, 2.8 GHz Intel Core 2 Duo. Test function used in this experiment was 10th dimensional Schwefel's function (see Eq. 1) and each experiment was 100 times repeated for each precision set. The aim was to find global extreme (10 × 418.983 at position 420.969, 420.969, ..., 420.969) of this function as precise as possible. So in total 3900 (3 algorithms × 100 repetitions × 13 different numerical precisions) evolutionary experiments has been done. In each experiment, the PRNGs used only on the start of Eq. 2 to set initial condition x_{start}. Remaining use of Eq. 2 was PRNGs free, i.e. PRNGs was not further in use. There were in fact three case of studies of our experiments.

$$\sum_{i=1}^{n} -x_i \sin(\sqrt{|x_i|}) \tag{1}$$

- **Case 1.** The first one was focused on the use of PDS, instead of PRNG only for mutations, so in DE was used classical PRNG to select individuals from population as well as in special error-correction procedures (when individual leave searched space and has to be returned back).
- **Case 2.** In the second set, all PRNG numbers were fully replaced by PDS numbers (with different precision and thus period length).

- **Case 3.** The last part of our experiments was focused on the use of the same algorithms (SOMA and DE) in its canonical versions with classical PRNGs (standard pseudo-random generator in Wolfram *Mathematica* 9) just to compare performance with both Cases 1 and 2.

$$x_{n+1} = Ax_n\left(1 - x_n\right) \tag{2}$$

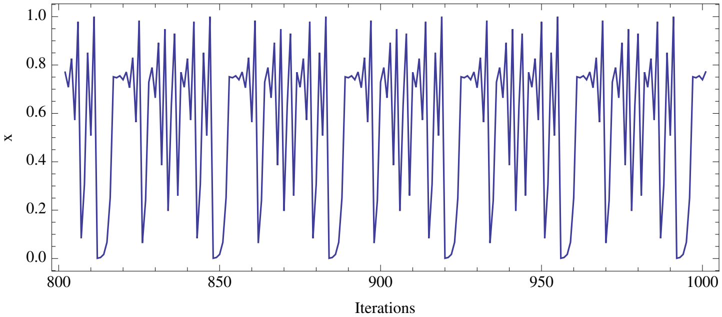


Fig. 1 Time series of period 36 (precision = 4) based on Eq. 2 for $A = 4$, see Tab. 2

Table 1 Algorithms setting. The same settings was used for both versions of DE.

DE		SOMA	
NP	20	PopSize	20
Dimensions	20	Dimensions	20
Generations	500	Migrations	20
F	0.9	PRT	0.1
CR	0.5	PathLength	5
		Step	0.11

4 Results

Results based on all three experiments are reported in Tab. 3 - 10 and Fig. 2 - 8. Tables and figures are organized sequentially according to case examples. Results of the **Case 1** are recorded in Fig. 2 - 4 further in Tab. 3, 5 and 7. Results of **Case 2** are recorded in Fig. 5 - 7 further in Tab. 4, 6 and 8. The last one, **Case 3**, is in Fig. 8 and Tab. 10.

For **Case 1** and **Case 2** has been used precision in DCHS according to Tab. 2 to get PDS. In this table is also recorded how many times was n periodic series repeatedly used in each algorithm. It is visible that in the frame of our experiments it was enough to set precision to 5 (Case 1, Fig. 2, Tab. 3, Tab. 5 and Tab. 7) and 8 (Case 2, Fig. 5, Tab. 4, Tab. 6 and Tab. 8).

Table 2 Periodicity dependance of Eq. 2 on various precision. Table also shows as to how many times was used n periodical series by the discussed algorithm.

Precision	Maximal Period	Repeated in SOMA	Repeated in DE
1	4	43181	32500
2	10	17272	13000
3	29	5956	4482
4	36	4797	3611
5	170	1016	764
6	481	359	270
7	758	227	171
8	4514	38	28
9	11227	15	11
10	35200	4	3
11	57639	2	2
12	489154	0	0
13	518694	0	0

All results from both cases can be compared with Fig. 8 and Tab. 10 from **Case 3** that was using only PRNG (standard pseudo-random generator in Wolfram *Mathematica* 9). When compared with Tab. 10 according to the Median value, then it is visible that in Case 1 it is comparable with precision equal to 4 (SOMA), 3 (DERand1Bin and DELocalToBest). For Case 2, it is enough to have precision to 5 (SOMA), 7 (DERand1Bin) and 8 (DELocalToBest). From Tab. 2 it is visible as to how many times was n periodic PDS used with given precisions. Results are also summarized in Tab. 9.

Table 3 Case 1. SOMA AllToOne.

Precision	Max	75%	Median	25%	Min
1	-585.311	-933.89	-1076.61	-1294.98	-1984.45
2	-771.596	-1225.85	-1392.08	-1572.34	-2324.94
3	-4144.75	-4175.6	-4179.39	-4182.09	-4186.95
4	-4051.62	-4182.91	-4186.16	-4187.5	-4189.39
5	-4070.28	-4186.44	-4187.86	-4188.75	-4189.68
6	-4172.77	-4185.4	-4187.13	-4188.04	-4189.42
7	-4068.95	-4187.24	-4188.75	-4189.03	-4189.65
8	-4058.85	-4184.46	-4187.2	-4188.05	-4189.29
9	-4069.58	-4185.51	-4187.4	-4187.98	-4189.48
10	-4177.38	-4186.37	-4187.5	-4188.46	-4189.55
11	-4174.2	-4185.52	-4187.49	-4188.56	-4189.35
12	-4066.18	-4185.32	-4187.4	-4188.61	-4189.71
13	-4144.51	-4186.16	-4187.58	-4188.7	-4189.51

Table 4 Case 2. SOMA AllToOne.

Precision	Max	75%	Median	25%	Min
1	823.851	687.668	687.668	-12.9572	-3042.29
2	-50.7378	-58.9429	-58.9429	-58.9429	-2146.33
3	220.071	220.071	-4057.14	-4132.14	-4169.95
4	-3761.36	-3987.13	-4044.72	-4068.93	-4182.16
5	-3914.18	-4136.2	-4185.91	-4189.17	-4189.62
6	-4157.77	-4187.88	-4188.76	-4189.12	-4189.60
7	-4069.75	-4187.	-4188.64	-4189.33	-4189.61
8	-4104.29	-4188.22	-4188.99	-4189.39	-4189.75
9	-4181.43	-4187.81	-4188.59	-4189.13	-4189.70
10	-4182.85	-4188.05	-4188.76	-4189.15	-4189.53
11	-4179.54	-4188.1	-4188.89	-4189.23	-4189.67
12	-4185.73	-4188.35	-4188.93	-4189.24	-4189.77
13	-4185.14	-4187.93	-4188.6	-4189.12	-4189.60

Table 5 Case 1. DERand1Bin.

Precision	Max	75%	Median	25%	Min
1	-1518.23	-1844.57	-2030.3	-2267.56	-2850.71
2	-2113.45	-2676.34	-2851.81	-3071.15	-3401.61
3	-4189.83	-4189.83	-4189.83	-4189.83	-4189.83
4	-4071.39	-4189.83	-4189.83	-4189.83	-4189.83
5	-4071.39	-4189.83	-4189.83	-4189.83	-4189.83
6	-4071.39	-4189.83	-4189.83	-4189.83	-4189.83
7	-4071.39	-4189.82	-4189.83	-4189.83	-4189.83
8	-4189.81	-4189.83	-4189.83	-4189.83	-4189.83
9	-4189.75	-4189.83	-4189.83	-4189.83	-4189.83
10	-4071.39	-4189.83	-4189.83	-4189.83	-4189.83
11	-4189.73	-4189.83	-4189.83	-4189.83	-4189.83
12	-4071.39	-4189.83	-4189.83	-4189.83	-4189.83
13	-4071.39	-4189.83	-4189.83	-4189.83	-4189.83

Table 6 Case 2. DERand1Bin.

Precision	Max	75%	Median	25%	Min
1	823.851	687.668	687.668	123.225	-12.9572
2	687.668	-58.9429	-58.9429	-58.9429	-58.9429
3	220.071	220.071	-1615.84	-2050.05	-2635.02
4	-1138.12	-1936.64	-2183.85	-2477.27	-3026.53
5	-1656.53	-2347.12	-2625.79	-3036.31	-4159.5
6	-2602.	-3712.17	-3975.73	-4189.82	-4189.83
7	-2417.19	-4189.32	-4189.81	-4189.83	-4189.83
8	-2591.58	-4189.82	-4189.83	-4189.83	-4189.83
9	-4189.65	-4189.83	-4189.83	-4189.83	-4189.83
10	-4071.39	-4189.83	-4189.83	-4189.83	-4189.83
11	-4189.76	-4189.83	-4189.83	-4189.83	-4189.83
12	-4188.91	-4189.83	-4189.83	-4189.83	-4189.83
13	-4071.39	-4189.83	-4189.83	-4189.83	-4189.83

Table 7 Case 1. DELocalToBest.

Precision	Max	75%	Median	25%	Min
1	-1339.95	-1872.95	-2060.63	-2292.03	-3371.71
2	-834.557	-2592.75	-2767.6	-2918.88	-3428.83
3	-971.065	-3429.78	-4189.83	-4189.83	-4189.83
4	-2004.78	-4172.27	-4188.8	-4189.68	-4189.83
5	-4052.35	-4189.76	-4189.82	-4189.83	-4189.83
6	-3886.55	-4186.71	-4189.71	-4189.82	-4189.83
7	-4071.28	-4189.77	-4189.82	-4189.83	-4189.83
8	-3980.9	-4189.76	-4189.83	-4189.83	-4189.83
9	-4071.39	-4189.82	-4189.83	-4189.83	-4189.83
10	-4071.39	-4189.82	-4189.83	-4189.83	-4189.83
11	-4071.38	-4189.81	-4189.83	-4189.83	-4189.83
12	-3895.89	-4189.8	-4189.83	-4189.83	-4189.83
13	-4015.18	-4189.8	-4189.83	-4189.83	-4189.83

Table 8 Case 2. DELocalToBest.

Precision	Max	75%	Median	25%	Min
1	823.851	687.668	687.668	123.225	-12.9572
2	-58.9429	-58.9429	-58.9429	-848.729	-2146.33
3	220.071	220.071	-2390.42	-2647.82	-3341.34
4	-1700.31	-2360.66	-2587.63	-2836.82	-3402.46
5	-1457.35	-2878.78	-3613.31	-4056.6	-4189.83
6	-2451.4	-3426.59	-3601.66	-3850.79	-4189.83
7	-2567.36	-3694.01	-4079.63	-4189.81	-4189.83
8	-3335.31	-4189.57	-4189.83	-4189.83	-4189.83
9	-4162.43	-4189.81	-4189.83	-4189.83	-4189.83
10	-3109.22	-4189.83	-4189.83	-4189.83	-4189.83
11	-4132.83	-4189.79	-4189.83	-4189.83	-4189.83
12	-4071.39	-4189.81	-4189.83	-4189.83	-4189.83
13	-3985.53	-4189.8	-4189.83	-4189.83	-4189.83

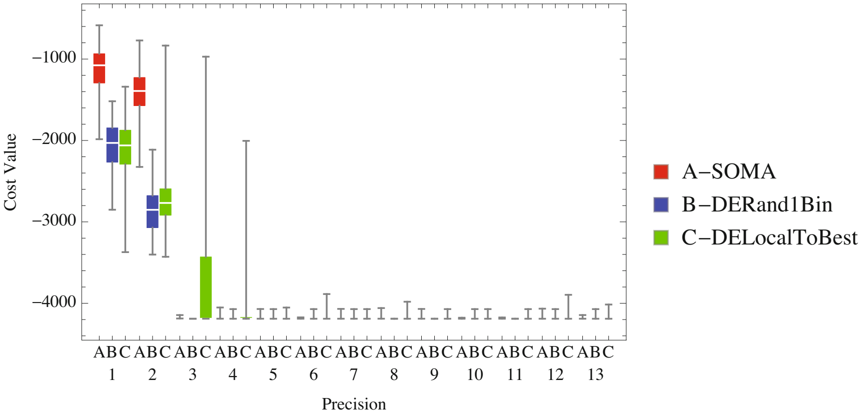


Fig. 2 Case 1. Dependance of algorithm performance on precision.

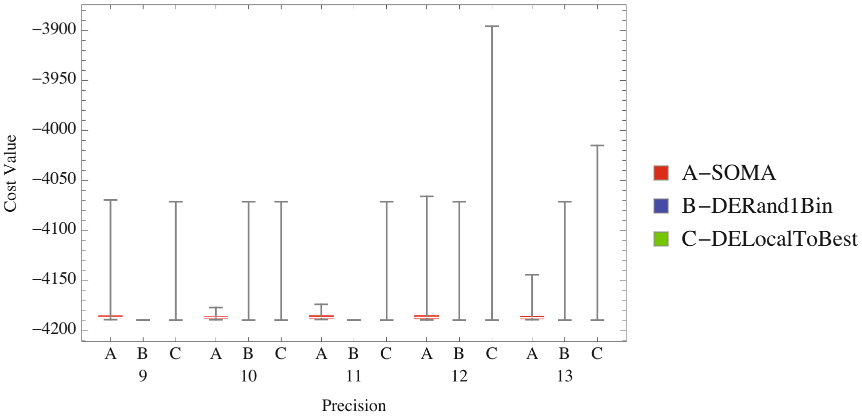


Fig. 3 Case 1. Dependance of algorithm performance on precision - detail.

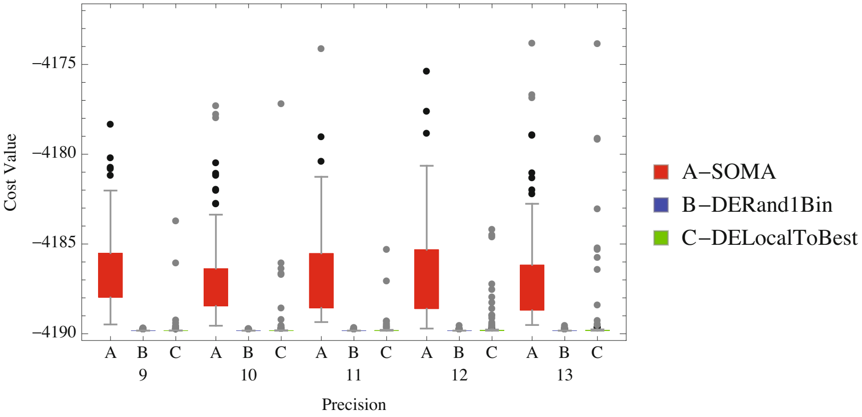


Fig. 4 Case 1. Dependance of algorithm performance on precision - detail with outliers.

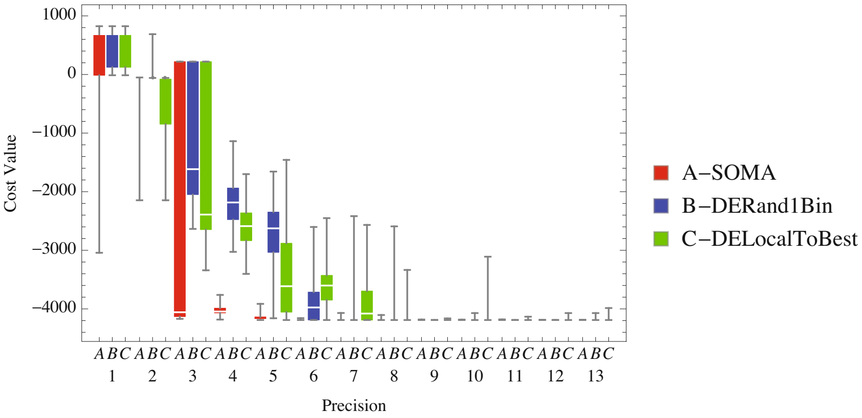


Fig. 5 Case 2. Dependance of algorithm performance on precision.

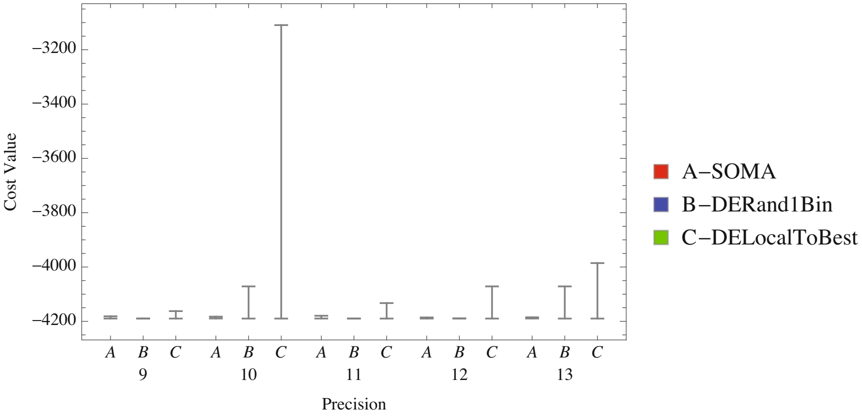


Fig. 6 Case 2. Dependance of algorithm performance on precision - detail.

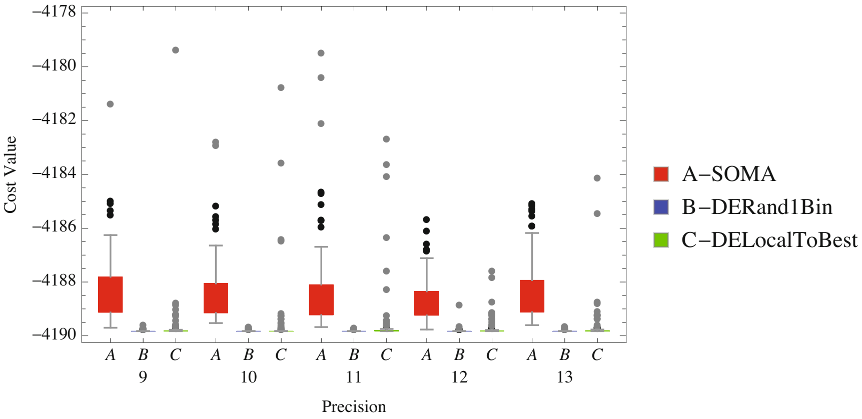


Fig. 7 Case 2. Dependance of algorithm performance on precision - detail with outliers.

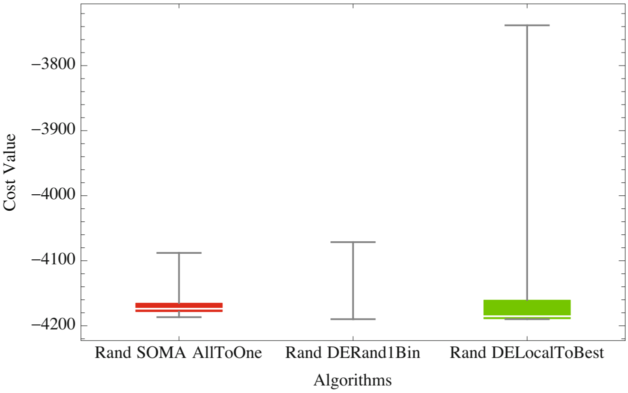


Fig. 8 Case 3. Results of EAs with PRNG, see Tab. 10.

Table 9 Summarization of all the best results from Tab. 10 from experiments based on EAs and PDS

Precision	SOMA		DERand1Bin		DELocalToBest	
	Case 1	Case 2	Case 1	Case 2	Case 1	Case 2
1	-1984.45	-3042.29	-2850.71	-12.9572	-3371.71	-12.9572
2	-2324.94	-2146.33	-3401.61	-58.9429	-3428.83	-2146.33
3	-4186.95	-4169.95	-4189.83	-2635.02	-4189.83	-3341.34
4	-4189.39	-4182.16	-4189.83	-3026.53	-4189.83	-3402.46
5	-4189.68	-4189.62	-4189.83	-4159.5	-4189.83	-4189.83
6	-4189.42	-4189.60	-4189.83	-4189.83	-4189.83	-4189.83
7	-4189.65	-4189.61	-4189.83	-4189.83	-4189.83	-4189.83
8	-4189.29	-4189.75	-4189.83	-4189.83	-4189.83	-4189.83
9	-4189.48	-4189.70	-4189.83	-4189.83	-4189.83	-4189.83
10	-4189.55	-4189.53	-4189.83	-4189.83	-4189.83	-4189.83
11	-4189.35	-4189.67	-4189.83	-4189.83	-4189.83	-4189.83
12	-4189.71	-4189.77	-4189.83	-4189.83	-4189.83	-4189.83
13	-4189.51	-4189.60	-4189.83	-4189.83	-4189.83	-4189.83

Table 10 Case 3. Results of EAs with PRNG.

Algorithm	Max	75%	Median	25%	Min
SOMA AllToOne	-4087.96	-4165.07	-4173.78	-4178.38	-4186.74
DERand1Bin	-4071.39	-4189.83	-4189.83	-4189.83	-4189.83
DELocalToBest	-3738.01	-4160.41	-4185.35	-4189.32	-4189.83

5 Conclusion

The main motivation of the research reported in this paper is whether it is possible to replace random number generators by deterministic processes originated in systems of deterministic chaos. In this participation were done three kind of experiments the first two used deterministic generators inside evolutionary algorithms (SOMA and differential algorithms) instead of pseudo-random number generators and the last one used standard pseudo-random generator in Wolfram *Mathematica* 9 in selected evolutionary algorithms to compare efficiency of proposed and tested methods.

For different numerical precessions were generated periodic series; see Tab. 2, that were used instead of random ones. Based on the obtained results it can be stated that at least in our case studies, all experiments exhibit fact that random number generators can be replaced by deterministic processes with small period (29 - 4514), with the repeated use in evolutionary algorithms with quite big frequency (5956 - 28 times). Results of the best reached minimum of each algorithm are also summarized in Tab. 9. An advantage of the proposed use of deterministic processes is the fact that in such case it is possible to fully repeat runs of given algorithm, analyze its behavior deterministically, including its full path on searched fitness landscape. We also believe that mathematical proofs can be in such case more easily constructed for such class of evolutionary algorithms.

Despite the widely presumed fact that pseudo-random number generators (also for evolutionary algorithms use) has to have as big period as possible (for example Mersenne twister with $2^{19937}-1$) and such as the 2^{32} common in many software packages, we show here that deterministic periodical processes with period 29 - 4514 is enough for our experiments.

Our further research is focused on more extensive and intensive testing of our ideas proposed here. Our aim is to try algorithms like scatter search [10], evolutionary strategies [11], genetic algorithms [12], [28] or particle swarm [13]. Also novel algorithms will be tested for its performance under our proposed approach in [26], [27] and alternative methods of symbolic regression [29].

Wider class of different algorithms, test functions and deterministic processes will be selected for future experiments to prove and specify the domain of validity of our ideas proposed here.

Acknowledgements. The following two grants are acknowledged for the financial support provided for this research: Grant Agency of the Czech Republic - GACR P103/13/08195S, by the Development of human resources in research and development of latest soft computing methods and their application in practice project, reg. no. CZ.1.07/2.3.00/20.0072 funded by Operational Programme Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic, partially supported by Grant of SGS No. SP2013/114, VŠB - Technical University of Ostrava, Czech Republic, and by European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089.

References

1. Zelinka, I., Celikovsky, S., Richter, H., Chen, G.: Evolutionary Algorithms and Chaotic Systems, p. 550. Springer, Germany (2010)
2. Lozi, R.: Emergence Of Randomness From Chaos. International Journal of Bifurcation and Chaos 22(2), 1250021 (2012), doi:10.1142/S0218127412500216
3. Wang, X.-Y., Qin, X.: A new pseudo-random number generator based on CML and chaotic iteration. Nonlinear Dynamics An International Journal of Nonlinear Dynamics and Chaos in Engineering Systems 70(2), 1589–1592 (2012) ISSN 0924-090X, doi:10.1007/s11071-012-0558-0
4. Pareek, N.K., Patidar, V., Sud, K.K.: A Random Bit Generator Using Chaotic Maps. International Journal of Network Security 10(1), 32–38 (2010)
5. Wang, X.-Y., Lei, Y.: Design Of Pseudo-Random Bit Generator Based On Chaotic Maps. International Journal of Modern Physics B 26(32), 1250208 (9 pages) (2012), doi:10.1142/S0217979212502086

6. Zhang, S.Y., Xingsheng, L.G.: A hybrid co-evolutionary cultural algorithm based on particle swarm optimization for solving global optimization problems. In: International Conference on Life System Modeling and Simulation / International Conference on Intelligent Computing for Sustainable Energy and Environment (LSMS-ICSEE), Wuxi, Peoples R China, September 17-20 (2010)
7. Hong, W.-C., Dong, Y., Zhang, Wen, Y., Chen, L.-Y., B.K., P.: Cyclic electric load forecasting by seasonal SVR with chaotic genetic algorithm. International Journal of Electrical Power and Energy Systems 44(1), 604–614, doi:10.1016/j.ijepes.2012.08.010
8. Zelinka, I.: SOMA – Self Organizing Migrating Algorithm. In: Babu, B.V., Onwubolu, G. (eds.) New Optimization Techniques in Engineering, pp. 167–218. Springer, New York
9. Price, K.: An Introduction to Differential Evolution. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, pp. 79–108. McGraw-Hill, London
10. Glover, F., Laguna, M.: Scatter Search. In: Ghosh, A., Tsutsui, S. (eds.) Advances in Evolutionary Computation: Theory and Applications, pp. 519–537. Springer, New York (2003)
11. Beyer, H.-G.: Theory of Evolution Strategies. Springer, New York (2001)
12. Holland, J.H.: Genetic Algorithms. Scientific American, 44–50 (July 1992)
13. Clerc, M.: Particle Swarm Optimization. ISTE Publishing Company (2006) ISBN 1905209045
14. Pluhacek, M., Senkerik, R., Zelinka, I.: Impact of various chaotic maps on the performance of chaos enhanced PSO algorithm with inertia weight – an initial study. In: Zelinka, I., Snasel, V., Rössler, O.E., Abraham, A., Corchado, E.S. (eds.) Nostradamus: Mod. Meth. of Prediction, Modeling. AISC, vol. 192, pp. 153–166. Springer, Heidelberg (2013)
15. Pluhacek, M., Senkerik, R., Davendra, D., Zelinka, I.: Designing PID controller for DC motor by means of enhanced PSO algorithm with dissipative chaotic map. In: Snasel, V., Abraham, A., Corchado, E.S. (eds.) SOCO Models in Industrial & Environmental Appl. AISC, vol. 188, pp. 475–483. Springer, Heidelberg (2013)
16. Yang, M., Guan, J., Cai, Z., Wang, L.: Self-adapting differential evolution algorithm with chaos random for global numerical optimization. In: Cai, Z., Hu, C., Kang, Z., Liu, Y. (eds.) ISICA 2010. LNCS, vol. 6382, pp. 112–122. Springer, Heidelberg (2010)
17. Coelho, L., Mariani, V.: Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect. IEEE Transactions On Power Systems 21(2), 989–996 (2006), doi:10.1109/TPWRS.2006.873410
18. Hu, G.-W.: Chaos-differential evolution for multiple sequence alignment. In: 3rd International Symposium on Intelligent Information Technology Application, Nanchang, Peoples R China, vol. 2, pp. 556–558., doi:10.1109/IITA.2009.511
19. Zhao, Q., Ren, J., Zhang, Z., Duan, F.: Immune co-evolution algorithm based on chaotic optimization. In: Workshop on Intelligent Information Technology Application (IITA 2007), Zhang Jiajie, Peoples R China, pp. 149–152 (2007)
20. Peng, C., Sun, H., Guo, J., Liu, G.: Dynamic economic dispatch for wind-thermal power system using a novel bi-population chaotic differential evolution algorithm. International Journal of Electrical Power & Energy Systems 42(1), 119–126 (2012), doi:10.1016/j.ijepes.2012.03.012

21. Zhang, H., Zhou, J., Zhang, Y., Fang, N., Zhang, R.: Short term hydrothermal scheduling using multi-objective differential evolution with three chaotic sequences. International Journal of Electrical Power & Energy Systems 47, 85–99 (2013), doi:10.1016/j.ijepes.2012.10.014
22. Zou, X., Wang, M., Zhou, A., McKay, B.: Evolutionary optimization based on chaotic sequence in dynamic environments. In: 2004 IEEE International Conference on Networking, Sensing and Contro, vol. 2, pp. 1364–1369 (2004)
23. Liua, B., Wanga, L., Jina, Y.-H., Tangb, F., Huanga, D.-X.: Improved particle swarm optimization combined with chaos. Chaos, Solitons & Fractals 25(5), 1261–1271 (2005)
24. Song, Y.: A bi-directional chaos optimization algorithm. In: 2010 Sixth International Conference on Natural Computation (ICNC), August 10-12, vol. 5, pp. 2202–2206 (2010)
25. Gandomi, A., Yun, G., Yang, X., Talatahari, S.: Chaos-enhanced accelerated particle swarm optimization. Communications In Nonlinear Science and Numerical Simulation 18(2), 327–340 (2013), doi:10.1016/j.cnsns.2012.07.017
26. Matousek, R.: HC12: The Principle of CUDA Implementation. In: Matousek (ed.) 16th International Conference on Soft Computing, MENDEL 2010, Brno, pp. 303–308 (2010)
27. Matousek, R., Zampachova, E.: Promising GAHC and HC12 algorithms in global optimization tasks. Journal Optimization Methods & Software 26(3), 405–419 (2011)
28. Matousek, R.: GAHC: Improved Genetic Algorithm. In: Krasnogor, N., Nicosia, G., Pavone, M., Pelta, D. (eds.) Nature Inspired Cooperative Strategies for Optimization (NICSO 2007). SCI, vol. 129, pp. 507–520. Springer, Heidelberg (2008)
29. Zelinka, I., Davendra, D., Senkerik, R., Jasek, R., Oplatkova, Z.: Analytical Programming - a Novel Approach for Evolutionary Synthesis of Symbolic Structures. In: Kita, E. (ed.) Evolutionary Algorithms. InTech (2011) ISBN: 978-953-307-171-8, doi:10.5772/16166, http://www.intechopen.com/books/evolutionary-algorithms/analytical-programming-a-novel-approach-for-evolutionary-synthesis-of-symbolic-structures

New Adaptive Approach for Chaos PSO Algorithm Driven Alternately by Two Different Chaotic Maps – An Initial Study

Michal Pluhacek, Roman Senkerik, Ivan Zelinka, and Donald Davendra

Abstract. In this initial study a novel adaptive approach for chaos driven PSO algorithm is proposed. Two different chaotic maps are used as pseudorandom number generators and switched over during the run of chaos driven PSO algorithm. The new adaptive approach brings promising results that are presented and briefly analyzed.

1 Introduction

The particle swarm optimization algorithm (PSO) is one of the new and promising evolutionary optimization algorithms (EAs) which are a class of soft computing optimization methods that is inspired by nature. In recent years, various EAs have been designed and implemented with promising results in many areas of complex optimization [1–6]. More recently, some studies indicated that using chaotic number generators might improve the quality of results, convergence speed or other performance indicators of EAs [7 - 13]. Several studies have already dealt with the possibilities of integration of chaotic systems into the PSO algorithm and the performance of such algorithms [12–14]. This paper presents a novel adaptive approach for chaos driven PSO algorithm that was developed as a continuation of previous research [13].

The paper is structured as follows: The second section gives the brief definition of the PSO algorithm. Description of used chaotic maps follows in the next sections. Subsequent sections contain the definitions of the adaptive approach and test functions used within this initial research and the experiment setup. Results, brief analysis and conclusion follow afterwards.

Michal Pluhacek · Roman Senkerik
Tomas Bata University in Zlin, Faculty of Applied Informatics, T.G. Masaryka 5555,
760 01 Zlin, Czech Republic
e-mail: {pluhacek,senkerik,zelinka}@fai.utb.cz

Ivan Zelinka · Donald Davendra
Department of Computer Science, Faculty of Electrical Engineering and Computer Science
VŠB-TUO, 17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic
e-mail: donald.davendra@vsb.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 77–87.
DOI: 10.1007/978-3-319-00542-3_9

2 Particle Swarm Optimization Algorithm

The PSO algorithm is inspired by the natural swarm behavior of birds and fish. It was introduced by Eberhart and Kennedy in 1995 [1] as an alternative to other EAs, such as Ant Colony Optimization [2], Genetic Algorithms (GA) [4] or Differential Evolution (DE) [5]. Each particle in the population represents a possible solution of the optimization problem which is defined by its cost function. In each generation, a new location (combination of cost function parameters) of the particle is calculated based on its previous location and velocity vector (velocity vector contains particle velocity for each dimension of the problem).

One of the disadvantages of the original PSO algorithm was poor local search ability. Another problem was the rapid acceleration of particles, which causes them to abandon the defined area of interest. For this reasons, several modifications of the PSO were introduced. The main principles of the PSO algorithm and its modifications are detailed in [1–3, 15, 16]. Within this research, the chaos driven PSO strategy with inertia weight was utilized [15]. This strategy was first introduced in 1998 [15] in order to improve the local search capability of PSO. The selection of inertia weight strategy of PSO was based on numerous previous experiments [9, 10, 13, 14]. Several modifications of inertia weight strategy are well described in [16]. In this study, linear decreasing inertia weight [15, 16] is used. Default values of all PSO parameters were chosen according to the recommendations given in [1–3,15,16].

Inertia weight is designed to influence the velocity of each particle differently over time [15, 16]. In the beginning of the optimization process, the influence of inertia weight factor w is minimal. As the optimization continues, the value of w is decreasing, thus the velocity of each particle is decreasing, since w is always the number less than one and it multiplies the previous velocity of particle in the process of new velocity value calculation. Inertia weight modification PSO strategy has two control parameters w_{start} and w_{end}. A new w for each generation is given by (1), where i stand for current generation number and n for the total number of generations.

$$w = w_{start} - \frac{((w_{start} - w_{end}) \cdot i)}{n} \tag{1}$$

A chaos driven pseudorandom number generator is used in the main PSO formula (Eq. (2)) that determines new ''velocity'' and thus the position of each particle in the next generation (or migration cycle).

$$\begin{aligned} v(t+1) &= w \cdot v(t) + c_1 \cdot Rand \cdot (pBest - x(t)) \\ &+ c_2 \cdot Rand \cdot (gBest - x(t)) \end{aligned} \tag{2}$$

Where:

$v(t + 1)$ - New velocity of a particle.
$v(t)$ - Current velocity of a particle.

c_1, c_2 - Priority factors.
pBest - Best solution found by a particle.
gBest - Best solution found in a population.
x(t) - Current position of a particle.
Rand - Random number, interval (0, 1). Chaos number generator is applied only here.

The new position of a particle is then given by (3), where *x(t + 1)* is the new position:

$$x(t+1) = x(t) + v(t+1) \tag{3}$$

3 Lozi Map

The Lozi map is a simple discrete two-dimensional chaotic map. The Lozi map is depicted in Fig. 1. The map equations are given in Eq. 11 and 12. The parameters used in this work are: a = 1.7 and b = 0.5 as suggested in [7, 17-18].

$$X_{n+1} = 1 - a\left|X_n\right| + bY_n \tag{4}$$

$$Y_{n+1} = X_n \tag{5}$$

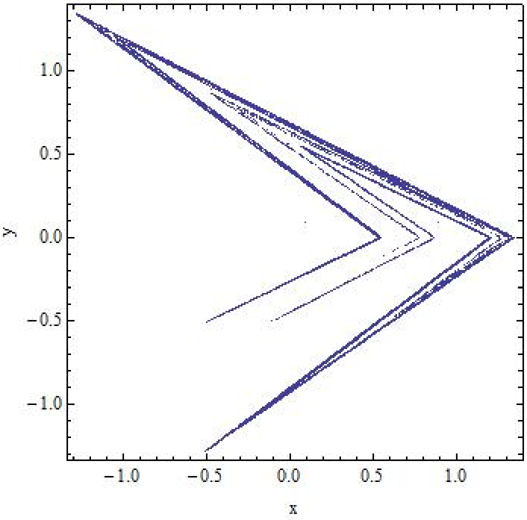


Fig. 1 Lozi map

4 Dissipative Standard Map

The Dissipative Standard map is a two-dimensional chaotic map. The parameters used in this work are b = 0.1 and k = 8.8 as suggested in [17]. The Dissipative standard map is given in Fig. 2. The map equations are given in Eq. 6 and 7.

$$X_{n+1} = X_n + Y_{n+1} (\operatorname{mod} 2\pi) \tag{6}$$

$$Y_{n+1} = bY_n + k \sin X_n (\operatorname{mod} 2\pi) \tag{7}$$

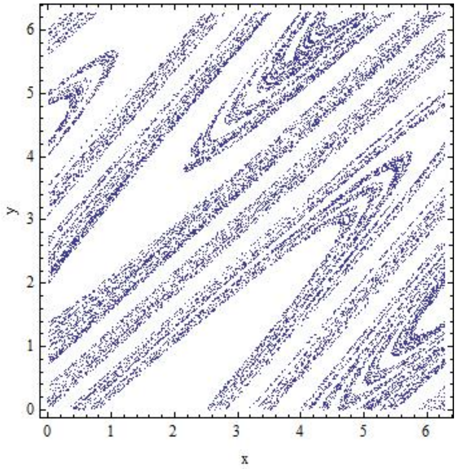


Fig. 2 Dissipative standard map

5 Notation

In total, four versions of PSO algorithm were used. The notation is as follows:

- PSO Weight – PSO algorithm with inertia weight – default C language built-in pseudorandom number generator.
- PSO Lozi – PSO algorithm with inertia weight – chaotic pseudorandom number generator with Lozi map.
- PSO Disi – PSO algorithm with inertia weight – chaotic pseudorandom number generator with Dissipative standard map.
- PSO Chaos – PSO algorithm with inertia weight utilizing two different chaotic pseudorandom number generators with Lozi map and Dissipative standard map (in given order).

As already mentioned in the section 2, the chaotic pseudorandom number generator was applied only for the main formula of PSO (2). For other purposes (generating of initial population etc.) default C language built-in pseudorandom number generator was used within all four described versions of PSO.

The new proposed algorithm (noted PSO Chaos) is identical to PSO Lozi until certain pre-defined automatic "switching" condition is met. Subsequently the pseudorandom number generator is switched over and uses the Dissipative standard map. Thus the PSO-Chaos algorithm is identical to PSO Disi version. Based on the numerous experiments with chaos driven PSO [13], the novelty of this approach is to benefit from both the fast convergence of PSO Lozi and the ability to avoid *gBest* stagnation that is typical feature of PSO Disi.

6 Chaos PSO Algorithm Driven Alternately by Two Different Chaotic Maps – An Adaptive Approach

Within this new proposed approach the optimization process is divided into two phases. The exact transition point from the first phase to the second one is determined by following simple rule:

If the change of *gBest* value between two subsequent generations is less than 0.001 over more than 1% of total number of generations, the first phase ends and the second phase is started.

The new proposed algorithm utilizes Lozi map for the first phase of the optimization process. At the start of the second phase the Lozi map is switched over to Dissipative standard map.

The novelty of the adaptive approach is that, the pseudorandom number generators are switched over automatically without prior knowledge of the optimization problem and without any manual setting of the "switching point" (that could typically be a certain number of generations achieved).

7 Test Functions

Four test functions were used in this study. Two static test function: 1st De Jong´s function, Rastrigin´s function and two shifted test functions: Shifted 1st De Jong´s function and Shifted Rastrigin´s function.

The 1st De Jong´s function is given by Eq. 8

$$f(x)=\sum_{i=1}^{\dim} x_i^2 \tag{8}$$

Function minimum:

Position for E_n: $(x_1,x_2 \ldots x_n) = (0,0,\ldots, 0)$

Value for E_n: $y = 0$

Rastrigin´s function is given by Eq. 9

$$f(x)=10\dim+\sum_{i=1}^{\dim} x_i^2-10\cos(2\pi x_i) \tag{9}$$

Function minimum:

Position for E_n: $(x_1,x_2 \ldots x_n) = (0,0,\ldots,0)$

Value for E_n: $y = 0$

Shifted 1st De Jong´s function is given by Eq. 10

$$f(x)=\sum_{i=1}^{\dim}(x_i-shift_i)^2 \tag{10}$$

Function minimum:

Position for E_n: $(x_1,x_2\ldots x_n) = \boldsymbol{shift}$

Value for E_n: $y = 0$

Shifted Rastrigin´s function is given by Eq. 11

$$f(x) = 10\,\text{dim} + \sum_{i=1}^{\text{dim}} (x_i - shift_i)^2 - 10\cos(2\pi x_i - shift_i) \tag{11}$$

Function minimum:

Position for E_n: $(x_1,x_2\ldots x_n) = \boldsymbol{shift}$

Value for E_n: $y = 0$

Shift$_i$ is a random number from interval <-5.11, 5.11>. Where <-5.11, 5.11> are the low and high bounds for the population individuals. ***Shift*** vector is randomly generated on each start of the optimization process.

8 Experiment Setup

For the experiments, each version of PSO algorithm (as presented in section 5) was tested on the set of four selected test functions.

Control parameters were set up based on the previous numerous experiments and literature [1-3,13] as follows:

Population size: 50
Generations: 5000
w_{start}: 0.9
w_{end}: 0.4
Dimension: 40

From the statistical reasons, all experiments were repeated 30 times.

9 Results and Analysis

In this section, the results for each test function are summarized to a statistical overview (Table 1 – 4). The best results (Cost function values) and the best mean results are highlighted by bold numbers. Also the histories of *gBest* values were tracked for each case study and they are depicted on Fig. 3 – 6.

Table 1 Results for the 1st De Jong´s function

	PSO Chaos	PSO Lozi	PSO Disi	PSO Weight
Mean CF Value:	**0.00073965**	0.375871	0.00188788	0.107528
Std. Dev.:	0.000404357	0.178256	0.00103238	0.0344475
CF Value Median:	0.000629695	0.36995	0.00172405	0.102521
Max. CF Value:	0.00196567	0.873381	0.00507515	0.182904
Min. CF Value:	**0.000224513**	0.109895	0.00055352	0.0393308

Table 2 Results for the shifted 1st De Jong´s function

	PSO Chaos	PSO Lozi	PSO Disi	PSO Weight
Mean CF Value:	**43.6701**	74.5847	44.4867	61.0551
Std. Dev.:	16.6307	18.1815	11.7714	14.486
CF Value Median:	40.29	74.4481	42.2706	60.4348
Max. CF Value:	89.212	119.278	63.1382	95.1964
Min. CF Value:	**20.6627**	38.1495	22.5233	34.9005

Table 3 Results for the Rastrigin´s function

	PSO Chaos	PSO Lozi	PSO Disi	PSO Weight
Mean CF Value:	**19.7609**	79.8255	21.3157	56.7228
Std. Dev.:	2.79069	20.3124	3.88759	16.1155
CF Value Median:	20.0801	76.2509	20.7344	57.9589
Max. CF Value:	25.6031	125.794	29.7158	95.8192
Min. CF Value:	**14.1543**	46.6955	13.0408	31.0069

Table 4 Results for the shifted Rastrigin´s function

	PSO Chaos	PSO Lozi	PSO Disi	PSO Weight
Mean CF Value:	242.987	294.186	**237.117**	259.964
Std. Dev.:	43.4855	44.0743	36.1697	39.9865
CF Value Median:	239.428	292.694	234.962	263.845
Max. CF Value:	336.988	383.449	345.377	353.837
Min. CF Value:	**157.808**	204.044	179.312	195.384

The results presented in Tables 1 - 4 seem to support the claim that using the adaptive automatic approach for switching over the chaotic pseudorandom number generators could lead to improved performance of the chaos driven PSO algorithm. The behavior of newly proposed algorithm (PSO Chaos) seems to correspond with the initial ideas as described in section 5. This could be clearly observed in Figures 3 - 6.

Within this initial study the adaptive rule was based on very simple idea and should be enhanced in order to improve the satisfactory results obtained. Thus the results presented in this initial study should motivate the future research in this promising area.

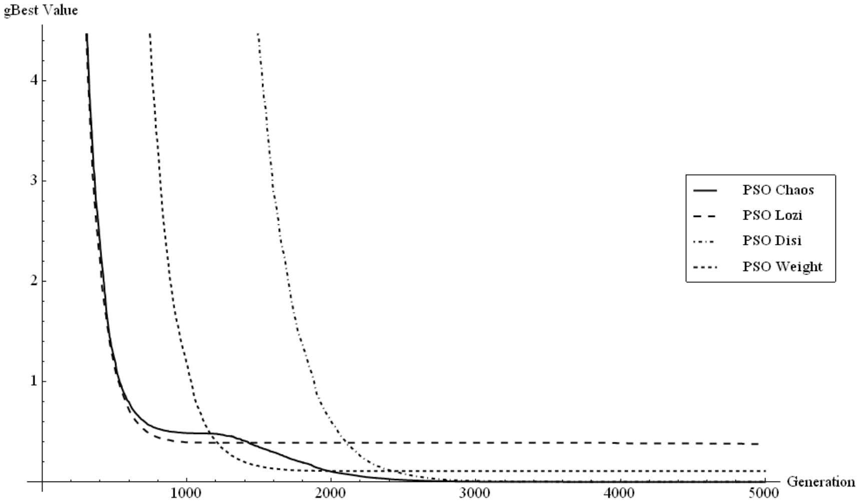


Fig. 3 1st De Jong´s function – Comparison of histories of mean best CF values for 30 runs

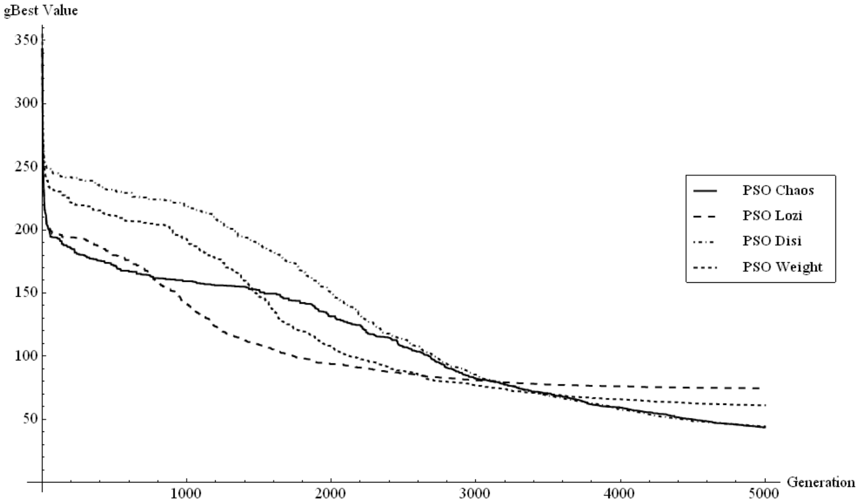


Fig. 4 Shifted 1st De Jong´s function - Comparison of histories of mean best CF values for 30 runs

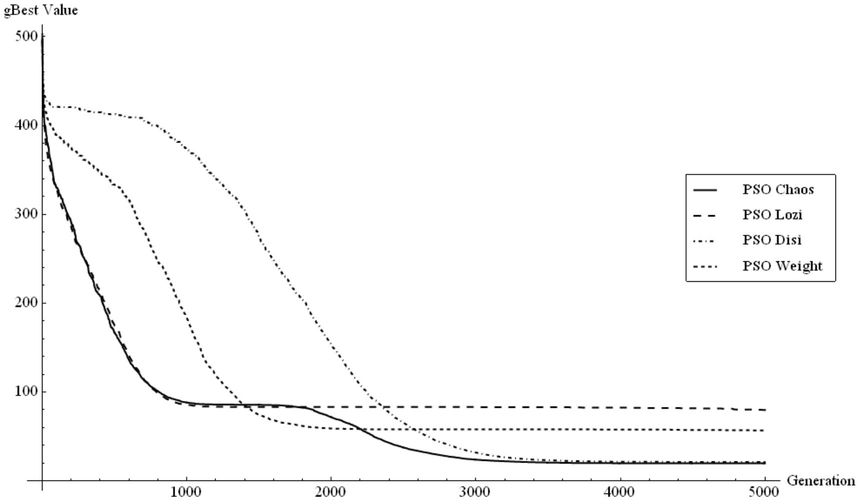


Fig. 5 Rastrigin´s function - Comparison of histories of mean best CF values for 30 runs

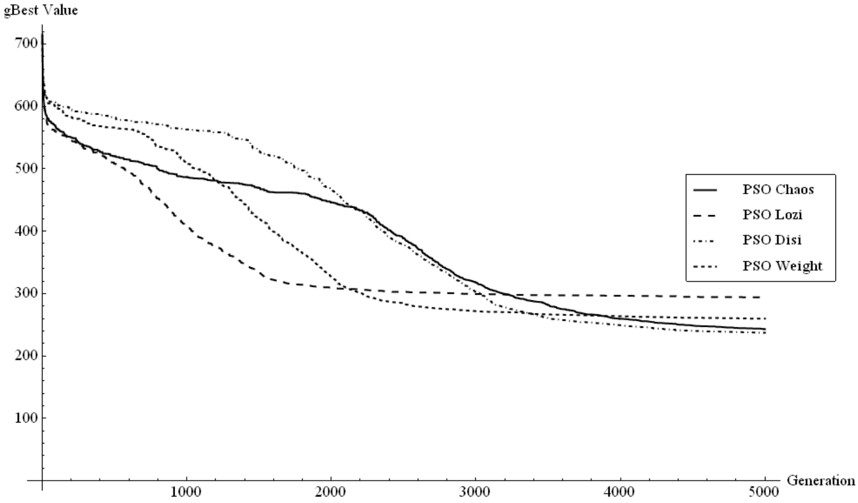


Fig. 6 Shifted Rastrigin´s function - Comparison of histories of mean best CF values for 30 runs

10 Conclusion

The new adaptive approach for chaos driven PSO algorithm was proposed within this initial research study. It was tested on a set of selected four test functions. Presented results seem to support the idea that switching over of chaotic pseudorandom number generators within one run of the PSO algorithm could improve its performance and the optimization process could benefit from the particular positives of PSO algorithm driven by each of these chaotic pseudorandom number generators. Regarding to the adaptive nature of the presented approach there is no need for prior knowledge of the optimization problem and manual setting of the "switching point".

Due to the relatively small range of this initial study more experiments should be conducted with different settings and results compared with other methods. However the primary aim of this paper is to inform about this new and promising approach that can possibly be extended and generalized into the usage in many other metaherustics.

Acknowledgements. This work was supported by Grant Agency of the Czech Republic-GACR P103/13/08195S, by the project Development of human resources in research and development of latest soft computing methods and their application in practice, reg. no. CZ.1.07/2.3.00/20.0072 funded by Operational Program Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic; European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089, by the Technology Agency of the Czech Republic under the Project TE01020197, and by Internal Grant Agency of Tomas Bata University under the project No. IGA/FAI/2013/012.

References

1. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. IV (1948)
2. Dorigo, M.: Ant Colony Optimization and Swarm Intelligence. Springer (2006)
3. Eberhart, R., Kennedy, J.: Swarm Intelligence. The Kaufmann Series in Artificial Intelligence. Morgan Kaufmann, San Francisco (2001)
4. Goldberg, D.E.: Genetic Algorithms in Search Optimization and Machine Learning, p. 41. Addison Wesley (1989) ISBN 0201157675
5. Storn, R., Price, R.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11, 341–359 (1997)
6. Zelinka: SOMA - self organizing migrating algorithm. In: Babu, B.V., Onwubolu, G. (eds.) New Optimization Techniques in Engineering, vol. 33, ch.7. Springer (2004) ISBN: 3-540-20167X
7. Caponetto, R., Fortuna, L., Fazzino, S., Xibilia, M.G.: Chaotic sequences to improve the performance of evolutionary algorithms. IEEE Transactions on Evolutionary Computation 7(3), 289–304 (2003)
8. Davendra, D., Zelinka, I., Senkerik, R.: Chaos driven evolutionary algorithms for the task of PID control. Computers & Mathematics with Applications 60(4), 1088–1104 (2010) ISSN 0898-1221
9. Pluhacek, M., Senkerik, R., Davendra, D., Zelinka, I.: Designing PID Controller For DC Motor System By Means Of Enhanced PSO Algorithm With Discrete Chaotic Lozi Map. In: Proceedings of the 26th European Conference on Modelling and Simulation, ECMS 2012, pp. 405–409 (2012) ISBN 978-0-9564944-4-3
10. Pluhacek, M., Senkerik, R., Davendra, D., Zelinka, I.: PID Controller Design For 4th Order System By Means Of Enhanced PSO algorithm With Lozi Chaotic Map. In: Proceedings of the 18th International Conference on Soft Computing, MENDEL 2012, pp. 35–39 (2012) ISBN 978-80-214-4540-6

11. Araujo, E., Coelho, L.: Particle swarm approaches using Lozi map chaotic sequences to fuzzy modelling of an experimental thermal-vacuum system. Applied Soft Computing 8(4), 1354–1364 (2008)
12. Alatas, B., Akin, E., Ozer, B.A.: Chaos embedded particle swarm optimization algorithms. Chaos, Solitons & Fractals 40(4), 1715–1734 (2009) ISSN 0960-0779
13. Pluhacek, M., Senkerik, R., Davendra, D., Kominkova Oplatkova, Z., Zelinka, I.: On the behavior and performance of chaos driven PSO algorithm with inertia weight. Computers and Mathematics with Applications (Article in press, 2013), doi:10.1016/j.camwa.2013.01.016
14. Pluhacek, M., Budikova, V., Senkerik, R., Oplatkova, Z., Zelinka, I.: On The Performance Of Enhanced PSO Algorithm With Lozi Chaotic Map – An initial Study. In: Proceedings of the 18th International Conference on Soft Computing, MENDEL 2012, pp. 40–45 (2012) ISBN 978-80-214-4540-6
15. Shi, Y.H., Eberhart, R.C.: A modified particle swarm optimizer. In: IEEE International Conference on Evolutionary Computation, Anchorage Alaska, pp. 69–73 (1998)
16. Nickabadi, M.M., Ebadzadeh, R.: A novel particle swarm optimization algorithm with adaptive inertia weight. Applied Soft Computing 11(4), 3658–3670 (2011) ISSN 1568-4946
17. Sprott, J.C.: Chaos and Time-Series Analysis. Oxford University Press (2003)
18. Aziz-Alaoui, M.A., Robert, C., Grebogi, C.: Dynamics of a Hénon–Lozi-type map. Chaos, Solitons & Fractals 12(12), 2323–2341 (2001)

On the Performance of Enhanced PSO Algorithm with Dissipative Chaotic Map in the Task of High Dimensional Optimization Problems

Michal Pluhacek, Roman Senkerik, Ivan Zelinka, and Donald Davendra

Abstract. In this paper, it is proposed the utilization of discrete Dissipative standard map based chaos pseudorandom number generator to enhance the performance of PSO algorithm with linear decreasing inertia weight in the task of high dimensional optimization. The performance is tested on a set of four typical test functions and the results are briefly analyzed and compared against the standard PSO version with inertia weight.

1 Introduction

The particle swarm optimization algorithm (PSO) is one of the new and promising evolutionary optimization algorithms (EAs) which are a class of soft computing optimization methods that is inspired by nature. In recent years, various EAs have been designed and implemented with promising results in many areas of complex optimization [1–6]. More recently, some studies indicated that using chaotic number generators might improve the quality of results, convergence speed or other performance indicators of EAs [7 - 13]. Several studies have already dealt with the possibilities of integration of chaotic systems into the PSO algorithm and the performance of such algorithms [12–14]. The aim of this paper is to investigate on the performance of the chaos driven PSO algorithm that uses Dissipative standard map as a pseudorandom number generator in the task of high dimensional optimization. This research builds up directly on the previous research [13].

Michal Pluhacek · Roman Senkerik · Ivan Zelinka
Tomas Bata University in Zlin, Faculty of Applied Informatics, T.G. Masaryka 5555,
760 01 Zlin, Czech Republic
e-mail: {pluhacek,senkerik,zelinka}@fai.utb.cz

Ivan Zelinka · Donald Davendra
Department of Computer Science, Faculty of Electrical Engineering and Computer Science
VŠB-TUO, 17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic
e-mail: donald.davendra@vsb.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 89 – 99.
DOI: 10.1007/978-3-319-00542-3_10

The paper is structured as follows: The second section gives the brief definition of the PSO algorithm. Description of used chaotic map follows in the next section. Subsequent sections contain the definitions of the notation and test functions used within this initial research and the experiment setup. Results, brief analysis and conclusion follow afterwards.

2 Particle Swarm Optimization Algorithm

The PSO algorithm is inspired by the natural swarm behavior of birds and fish. It was introduced by Eberhart and Kennedy in 1995 [1] as an alternative to other EAs, such as Ant Colony Optimization [2], Genetic Algorithms (GA) [4] or Differential Evolution (DE) [5]. Each particle in the population represents a possible solution of the optimization problem which is defined by its cost function. In each generation, a new location (combination of cost function parameters) of the particle is calculated based on its previous location and velocity vector (velocity vector contains particle velocity for each dimension of the problem).

One of the disadvantages of the original PSO algorithm was poor local search ability. Another problem was the rapid acceleration of particles, which causes them to abandon the defined area of interest. For this reasons, several modifications of the PSO were introduced. The main principles of the PSO algorithm and its modifications are detailed in [1–3, 15, 16]. Within this research, the chaos driven PSO strategy with inertia weight was utilized [15]. This strategy was first introduced in 1998 [15] in order to improve the local search capability of PSO. The selection of inertia weight strategy of PSO was based on numerous previous experiments [9, 10, 13, 14]. Several modifications of inertia weight strategy are well described in [16]. In this study, linear decreasing inertia weight [15, 16] is used. Default values of all PSO parameters were chosen according to the recommendations given in [1–3,15,16].

Inertia weight is designed to influence the velocity of each particle differently over time [15, 16]. In the beginning of the optimization process, the influence of inertia weight factor w is minimal. As the optimization continues, the value of w is decreasing, thus the velocity of each particle is decreasing, since w is always the number less than one and it multiplies the previous velocity of particle in the process of new velocity value calculation. Inertia weight modification PSO strategy has two control parameters w_{start} and w_{end}. A new w for each generation is given by Eq. 1, where i stand for current generation number and n for the total number of generations.

$$w = w_{start} - \frac{((w_{start} - w_{end}) \cdot i)}{n} \tag{1}$$

A chaos driven pseudorandom number generator is used in the main PSO formula (Eq. 2) that determines new ‘‘velocity’’ and thus the position of each particle in the next generation (or migration cycle).

$$v(t+1) = w \cdot v(t) + c_1 \cdot Rand \cdot (pBest - x(t)) \\ + c_2 \cdot Rand \cdot (gBest - x(t)) \tag{2}$$

Where:

v(t + 1) - New velocity of a particle.
v(t) - Current velocity of a particle.
c_1, c_2 - Priority factors.
pBest - Best solution found by a particle.
gBest - Best solution found in a population.
x(t) - Current position of a particle.
Rand - Random number, interval (0, 1). Chaos number generator is applied only here.

The new position of a particle is then given by Eq. 3, where *x(t + 1)* is the new position:

$$x(t+1) = x(t) + v(t+1) \tag{3}$$

3 Dissipative Standard Map

The Dissipative Standard map is a two-dimensional chaotic map. The parameters used in this work are $b = 0.1$ and $k = 8.8$ as suggested in [7,8, 17]. The Dissipative standard map is given in Fig. 1. The map equations are given in Eq. 4 and 5.

$$X_{n+1} = X_n + Y_{n+1} (\text{mod}\, 2\pi) \tag{4}$$

$$Y_{n+1} = bY_n + k \sin X_n (\text{mod}\, 2\pi) \tag{5}$$

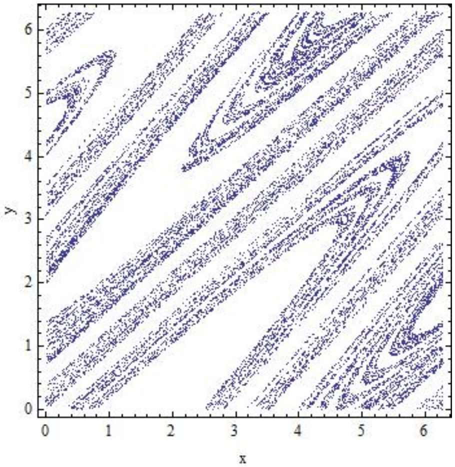


Fig. 1 Dissipative standard map

4 Notations

Two versions of PSO algorithm were used for results comparison in this study. The notation is as follows:

- PSO Weight – PSO algorithm with inertia weight – default C language built-in pseudorandom number generator.
- PSO Disi – PSO algorithm with inertia weight – chaotic pseudorandom number generator with Dissipative standard map.

As already mentioned in the section 2, the chaotic pseudorandom number generator was applied only for the main formula of PSO (Eq. 2). For other purposes representing generating of initial population etc., the default C language built-in pseudorandom number generator was used within both described versions of PSO.

5 Test Functions

Four test functions were used in this study: 1st De Jong´s function, 2nd De Jong´s function, Rastrigin´s function and Schwefel´s function

The 1st De Jong´s function is given by Eq. 6

$$f(x)=\sum_{i=1}^{\dim} x_i^2 \tag{6}$$

Function minimum:

Position for E_n: $(x_1,x_2\ldots x_n) = (0,0,\ldots, 0)$

Value for E_n: $y = 0$

The 2nd De Jong´s function is given by Eq. 7

$$f(x)=\sum_{i=1}^{\dim-1} 100(x_i^2-x_{i+1})^2+(1-x_i)^2 \tag{7}$$

Function minimum:

Position for E_n: $(x_1,x_2\ldots x_n) = (1,1,\ldots,1)$

Value for E_n: $y = 0$

Rastrigin´s function is given by Eq. 8

$$f(x)=10\dim+\sum_{i=1}^{\dim} x_i^2-10\cos(2\pi x_i) \tag{8}$$

Function minimum:

Position for E_n: $(x_1,x_2\ldots x_n) = (0,0,\ldots,0)$

Value for E_n: $y = 0$

Schwefel´s function is given by Eq. 9

$$f(x)=\sum_{i=1}^{\dim} -x_i\sin(\sqrt{|x|}) \tag{9}$$

Function minimum:

Position for E_n: $(x_1,x_2\ldots x_n) = (420.969, 420.969,\ldots, 420.969)$

Value for E_n: $y = -418.983 \cdot \text{dimension}$

6 Experiment Setup

For the experiments, each version of PSO algorithm (as presented in section 4) was tested on the set of four selected test functions.

Control parameters were set up based on the previous numerous experiments and literature [1-3,13] as follows:

Population size: 200
Generations: 1000
w_{start}: 0.9
w_{end}: 0.4
Dimension: 250, 500, 750, 1000

From the statistical reasons, all experiments were repeated 50 times.

7 Results

In this section, the results for each test function are summarized to a statistical overview (Table 1 – 8). The best results (Cost function values) and the best mean results are highlighted by bold numbers. Also the histories of *gBest* values were tracked. The histories for the case of dimension = 1000 are depicted in Fig. 2 – 5.

Table 1 Results for the 1st De Jong´s function

Dimension:	250		500	
PSO Version:	PSO Weight	PSO Disi	PSO Weight	PSO Disi
Mean CF Value:	39.0259	**7.77092**	146.802	**32.5696**
Std. Dev.:	5.12331	1.00339	14.259	3.34175
CF Value Median:	39.4684	7.68393	145.331	32.0691
Max. CF Value:	48.4521	9.88144	180.687	42.1374
Min. CF Value:	26.0462	**5.89301**	115.8	**25.6589**

Table 2 Results for the 1st De Jong´s function

Dimension:	750		1000	
PSO Version:	PSO Weight	PSO Disi	PSO Weight	PSO Disi
Mean CF Value:	282.214	**62.6671**	421.35	**96.6133**
Std. Dev.:	25.3314	6.40367	40.7967	10.2843
CF Value Median:	280.313	62.6731	419.156	96.5296
Max. CF Value:	349.26	80.3937	520.948	121.711
Min. CF Value:	216.792	**46.9578**	340.627	**78.6706**

Table 3 Results for the 2nd De Jong´s function

Dimension:	250		500	
PSO Version:	PSO Weight	PSO Disi	PSO Weight	PSO Disi
Mean CF Value:	8063.92	**1663.45**	37666.1	**5972.86**
Std. Dev.:	1709.16	183.448	7131.63	1059.25
CF Value Median:	8140.04	1679	37122.2	5938.4
Max. CF Value:	12957.5	2098.96	49430.3	9628.83
Min. CF Value:	4947.98	**1247.27**	22860.2	**4155.74**

Table 4 Results for the 2nd De Jong´s function

Dimension:	750		1000	
PSO Version:	PSO Weight	PSO Disi	PSO Weight	PSO Disi
Mean CF Value:	82306.3	**11510.5**	133109	**18737.2**
Std. Dev.:	14131.7	2228.31	24884	2736.98
CF Value Median:	80806.8	10967.4	128420	19036.8
Max. CF Value:	122079	21046	204027	25690.4
Min. CF Value:	56176.8	**8596.26**	91841.4	**13039**

Table 5 Results for the Rastrigin´s function

Dimension:	250		500	
PSO Version:	PSO Weight	PSO Disi	PSO Weight	PSO Disi
Mean CF Value:	1549.5	**1144.66**	3834.95	**3232.45**
Std. Dev.:	82.3611	73.6264	128.073	164.864
CF Value Median:	1555.93	1149.73	3826.75	3202.81
Max. CF Value:	1754.44	1326.2	4180.54	3715.04
Min. CF Value:	1367.45	**985.083**	3499.8	**2828.99**

Table 6 Results for the Rastrigin´s function

Dimension:	750		1000	
PSO Version:	PSO Weight	PSO Disi	PSO Weight	PSO Disi
Mean CF Value:	6203.91	**5432.04**	8673.68	**7875.76**
Std. Dev.:	166.342	190.661	185.721	178.29
CF Value Median:	6189.49	5419.88	8660.48	7870.81
Max. CF Value:	6611.81	5853.13	9040.82	8280.64
Min. CF Value:	5899.95	**5086.82**	8273.31	**7542.85**

Table 7 Results for the Schwefel´s function

Dimension:	250		500	
PSO Version:	PSO Weight	PSO Disi	PSO Weight	PSO Disi
Mean CF Value:	**-20866.1**	-18457.4	**-29893.3**	-26603.2
Std. Dev.:	1739.44	1745.02	2739.61	3304.43
CF Value Median:	-20837.8	-18302.2	-30047.3	-26888.2
Max. CF Value:	-17331.5	-15045.7	-20550.1	-19523.5
Min. CF Value:	**-24401.6**	-22991.5	**-35694.8**	-33803.4

Table 8 Results for the Schwefel´s function

Dimension:	750		1000	
PSO Version:	PSO Weight	PSO Disi	PSO Weight	PSO Disi
Mean CF Value:	**-36454.2**	-31499	**-41737.4**	-37417.5
Std. Dev.:	3027.57	3620.46	3654.49	3880.61
CF Value Median:	-36636	-31550.9	-41886.6	-36948.5
Max. CF Value:	-28276.4	-23223.1	-31367.4	-29006.7
Min. CF Value:	**-41492**	-39301.7	**-48610.3**	-47084.1

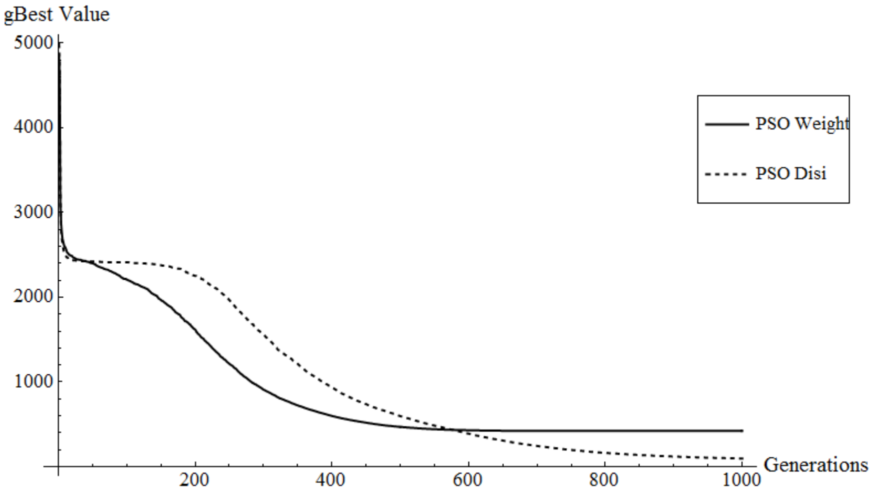


Fig. 2 1st De Jong´s function – Comparison of histories of mean best CF values for 50 runs

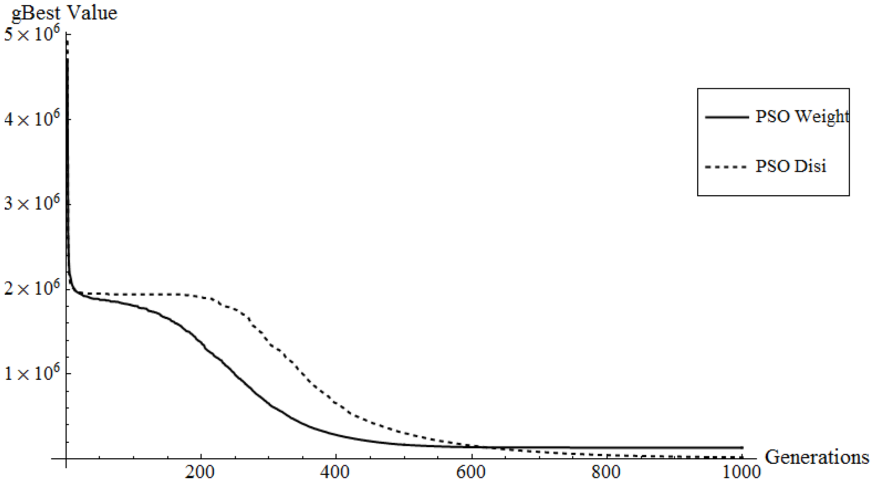


Fig. 3 2nd De Jong´s function **-** Comparison of histories of mean best CF values for 50 runs

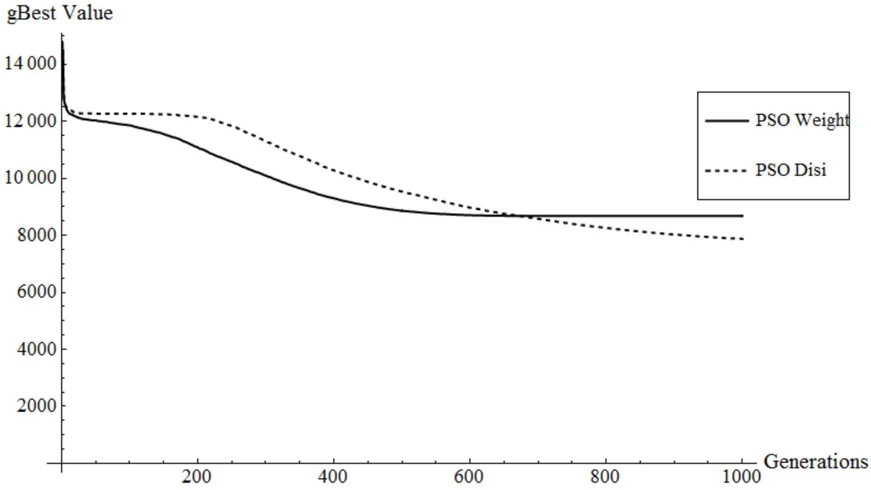


Fig. 4. Rastrigin´s function - Comparison of histories of mean best CF values for 50 runs

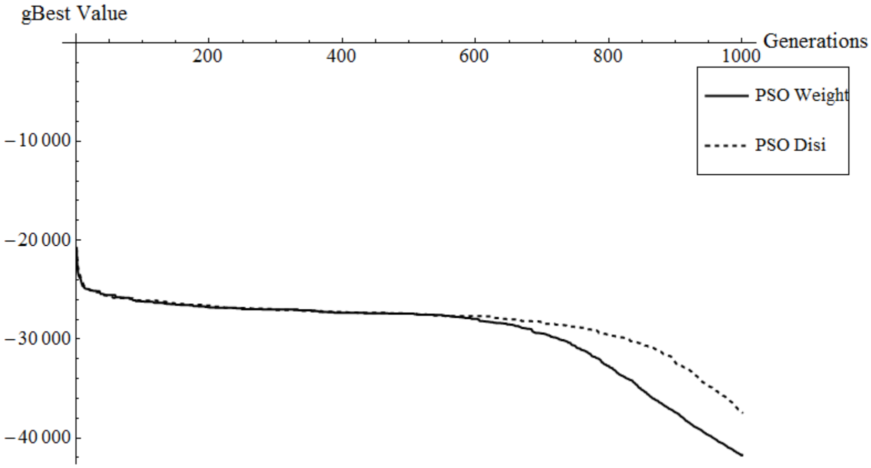


Fig. 5 Schwefel´s function - Comparison of histories of mean best CF values for 50 runs

8 Brief Results Analysis

The results presented in Tables 1 – 2 and Fig. 2 strongly support the claim that using Dissipative standard map as a pseudorandom number generator could significantly improve the performance of PSO algorithm in high dimensional optimization task on such a simple test function, which is the 1st De Jong´s function.

Presented results for the 2nd De Jong´s function (See Tables 3 - 4 and Fig. 3) truly confirm the above mentioned claim.

In the case study of the Rastrigin´s test function (See Tables 5 - 6 and Fig. 4), the performance improvement of PSO algorithm is not so significant as in the previous two case studies. Nevertheless it can be stated that the PSO Disi algorithm managed to obtain better results for all four high dimension settings.

The only exception from the prevailing results trend can be found in the case study of the Schwefel´s test function (presented in Tables 7 – 8 and Fig. 5). With the limited number of generation set to 1000, the PSO Disi did not manage to find better solution than the standard PSO Weight version, where the chaos based pseudorandom number generator was not utilized. However based on the previous successful experiences with PSO algorithm driven by Dissipative standard map based pseudorandom number generator [13], it could be assumed that the performance of PSO Disi algorithm in higher dimensional optimization problems task will be improved together with the increasing of number of generations, thus it will become superior to the performance of PSO Weight algorithm.

9 Conclusion

In this paper, the PSO algorithm enhanced with discrete Dissipative chaotic map was presented and investigated over its performance and ability to deal with high dimensional optimization problems. Obtained promising results presented in tables and figures seem to give weight to the claims that using Dissipative standard map based pseudorandom number generator could significantly improve the performance of PSO algorithm in the task of high dimensional optimization problems. More experiments should be undertaken to prove or disprove these claims. The mostly superior change (with one exception) in the behavior of PSO algorithm driven by Dissipative standard chaotic map can be clearly observed in Figures 2 - 5. This research should encourage future research activity in this promising area of chaos driven metaheuristics.

Acknowledgements. This work was supported by Grant Agency of the Czech Republic-GACR P103/13/08195S, by the project Development of human resources in research and development of latest soft computing methods and their application in practice, reg. no. CZ.1.07/2.3.00/20.0072 funded by Operational Program Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic; European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089, by the Technology Agency of the Czech Republic under the Project TE01020197, and by Internal Grant Agency of Tomas Bata University under the project No. IGA/FAI/2013/012.

References

1. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. IV, pp. 1942–1948 (1995)
2. Dorigo, M.: Ant Colony Optimization and Swarm Intelligence. Springer (2006)
3. Eberhart, R., Kennedy, J.: Swarm Intelligence. The Kaufmann Series in Artificial Intelligence. Morgan Kaufmann (2001)

4. Goldberg, D.E.: Genetic Algorithms in Search Optimization and Machine Learning, p. 41. Addison Wesley (1989) ISBN 0201157675
5. Storn, R., Price, R.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11, 341–359 (1997)
6. Zelinka: SOMA - self organizing migrating algorithm. In: Babu, B.V., Onwubolu, G. (eds.) New Optimization Techniques in Engineering, vol. 33, ch.7. Springer (2004) ISBN: 3-540-20167X
7. Caponetto, R., Fortuna, L., Fazzino, S., Xibilia, M.G.: Chaotic sequences to improve the performance of evolutionary algorithms. IEEE Transactions on Evolutionary Computation 7(3), 289–304 (2003)
8. Davendra, D., Zelinka, I., Senkerik, R.: Chaos driven evolutionary algorithms for the task of PID control. Computers & Mathematics with Applications 60(4), 1088–1104 (2010) ISSN 0898-1221
9. Pluhacek, M., Senkerik, R., Davendra, D., Zelinka, I.: Designing PID Controller For DC Motor System By Means Of Enhanced PSO Algorithm With Discrete Chaotic Lozi Map. In: Proceedings of the 26th European Conference on Modelling and Simulation, ECMS 2012, pp. 405–409 (2012) ISBN 978-0-9564944-4-3
10. Pluhacek, M., Senkerik, R., Davendra, D., Zelinka, I.: PID Controller Design For 4th Order System By Means Of Enhanced PSO algorithm With Lozi Chaotic Map. In: Proceedings of the 18th International Conference on Soft Computing, MENDEL 2012, pp. 35–39 (2012) ISBN 978-80-214-4540-6
11. Araujo, E., Coelho, L.: Particle swarm approaches using Lozi map chaotic sequences to fuzzy modelling of an experimental thermal-vacuum system. Applied Soft Computing 8(4), 1354–1364 (2008)
12. Alatas, B., Akin, E., Ozer, B.A.: Chaos embedded particle swarm optimization algorithms. Chaos, Solitons & Fractals 40(4), 1715–1734 (2009) ISSN 0960-0779
13. Pluhacek, M., Senkerik, R., Davendra, D., Kominkova Oplatkova, Z., Zelinka, I.: On the behavior and performance of chaos driven PSO algorithm with inertia weight. Computers and Mathematics with Applications (Article in press, 2013), doi:10.1016/j.camwa.2013.01.016
14. Pluhacek, M., Budikova, V., Senkerik, R., Oplatkova, Z., Zelinka, I.: On The Performance Of Enhanced PSO Algorithm With Lozi Chaotic Map – An initial Study. In: Proceedings of the 18th International Conference on Soft Computing, MENDEL 2012, pp. 40–45 (2012) ISBN 978-80-214-4540-6
15. [14] Shi, Y.H., Eberhart, R.C.: A modified particle swarm optimizer. In: IEEE International Conference on Evolutionary Computation, Anchorage Alaska, pp. 69–73 (1998)
16. Nickabadi, M.M., Ebadzadeh, R.: A novel particle swarm optimization algorithm with adaptive inertia weight. Applied Soft Computing 11(4), 3658–3670 (2011) ISSN 1568-4946
17. Sprott, J.C.: Chaos and Time-Series Analysis. Oxford University Press (2003)

Utilization of Analytic Programming for Evolutionary Synthesis of the Robust Controller for Set of Chaotic Systems

Roman Senkerik, Zuzana Kominkova Oplatkova, Ivan Zelinka, and Michal Pluhacek

Abstract. In this paper, it is presented a utilization of tool for symbolic regression, which is analytic programming, for the purpose of the synthesis of a new robust feedback control law. This universal synthesized robust chaotic controller secures the fully stabilization of selected set of three discrete chaotic systems. The paper consists of the descriptions of analytic programming as well as selected chaotic systems, used heuristic and cost function design. For experimentation, Self-Organizing Migrating Algorithm (SOMA) and Differential evolution (DE) were used.

1 Introduction

During the recent years, usage of new intelligent systems in engineering, technology, modeling, computing and simulations has attracted the attention of researchers worldwide. The most current methods are mostly based on soft computing, which is a discipline tightly bound to computers, representing a set of methods of special algorithms, belonging to the artificial intelligence paradigm. The most popular of these methods are neural networks, evolutionary algorithms, fuzzy logic and tools for symbolic regression like genetic programming. Currently, evolutionary algorithms are known as a powerful set of tools for almost any difficult and complex optimization problem.

Roman Senkerik · Zuzana Oplatkova · Michal Pluhacek
Tomas Bata University in Zlin, Faculty of Applied Informatics, T.G. Masaryka 5555,
760 01 Zlin, Czech Republic
e-mail: {senkerik,oplatkova,pluhacek}@fai.utb.cz

Ivan Zelinka
VŠB-Technical University of Ostrava, Faculty of Electrical Engineering and Computer Science, Department of Computer Science, 17. listopadu 15,
708 33 Ostrava-Poruba, Czech Republic
e-mail: ivan.zelinka@vsb.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 101 – 110.
DOI: 10.1007/978-3-319-00542-3_11

The interest about the interconnection between evolutionary techniques and control of chaotic systems is spread daily. First steps were done in [1] representing the utilization of differential evolution algorithm for the synchronization and control of chaotic systems. The papers [2], [3] were concerned to tune several parameters inside the original control technique for discrete chaotic systems. The evolutionary tuned control technique was based on Pyragas method: Extended delay feedback control – ETDAS [4]. Another example of interconnection between deterministic chaos and evolutionary algorithms represents the research focused on the embedding of chaotic dynamics into the evolutionary algorithms [5] - [7].

This paper shows a possibility how to generate the whole robust control law by means of analytic programming (AP) (not only to optimize several parameters) for the purpose of stabilization of a set of chaotic systems. The synthesis of control is inspired by the Pyragas's delayed feedback control technique [8], [9].

AP is a superstructure of EAs and is used for synthesis of analytic solution according to the required behaviour. Control law from the proposed system can be viewed as a symbolic structure, which can be synthesized according to the requirements for the stabilization of the chaotic system.

Firstly, AP is explained, and then a problem design is proposed. The next sections are focused on the description of used softcomputing tools and the design of cost function. Results and conclusion follow afterwards.

2 Motivation

This work is focused on the expansion of AP application for synthesis of a whole robust control law instead of parameters tuning for existing and commonly used control technique to stabilize desired Unstable Periodic Orbits (UPO) of set of selected discrete chaotic systems.

This work represents an extension of previous research [10], [11], where the new control laws were evolutionary synthesized only individually for the particular chaotic systems.

In general, this research is concerned to stabilize set of chaotic systems at p-1 UPO, which is a stable state, utilizing the only one universal robust synthesized control law.

3 Selected Discrete Chaotic Systems

This section contains the brief description of selected and used discrete chaotic systems within this research.

3.1 Logistic Equation

The first described example of discrete chaotic systems is the one-dimensional Logistic equation in form (1):

$$x_{n+1} = rx_n\left(1 - x_n\right). \tag{1}$$

The Logistic equation (logistic map) is a one-dimensional discrete-time example of how complex chaotic behaviour can arise from very simple non-linear dynamical equation [12]. This chaotic system was introduced and popularized by the biologist Robert May [13]. The example of its chaotic behaviour can be clearly seen from bifurcation diagram (Fig. 1).

3.2 Hénon Map

The second chosen example of chaotic system was the two dimensional Hénon map in form (2):

$$\begin{aligned} x_{n+1} &= a - x_n^2 + by_n \\ y_{n+1} &= x_n \end{aligned} \tag{2}$$

This is a model invented with a mathematical motivation to investigate chaos. The Hénon map is a discrete-time dynamical system, which was introduced as a simplified model of the Poincaré map for the Lorenz system. It is one of the most studied examples of dynamical systems that exhibit chaotic behavior. The map depends on two parameters, *a* and *b*, which for the canonical Hénon map have values of *a* = 1.4 and *b* = 0.3. For these canonical values the Hénon map is chaotic [12]. The example of this chaotic behavior can be clearly seen from bifurcation diagram (Fig. 1), which was created by plotting of a variable *x* as a function of the one control parameter for the fixed second parameter.

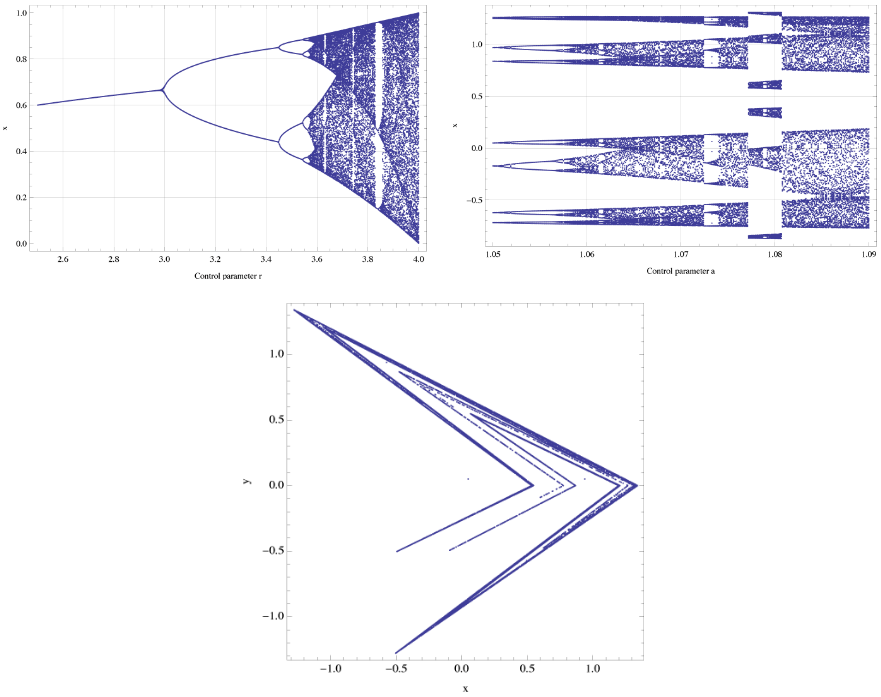


Fig. 1 Bifurcation diagrams of Logistic equation (upper left), Hénon map (upper right) and *x*-*y* plot of the Lozi map (below)

3.3 Lozi Map

The last example of discrete chaotic systems is the Lozi map, which represents the simple discrete two-dimensional chaotic map. The *x*, *y* plot of the Lozi map is depicted in Fig. 1. The map equations are given in (3). The parameters are: a = 1.7 and b = 0.5 as suggested in [14].

$$\begin{aligned} X_{n+1} &= 1 - a\left|X_n\right| + bY_n \\ Y_{n+1} &= X_n \end{aligned} \tag{3}$$

4 Original Chaos Control Method

This work is focused on explanation of application of AP for synthesis of a whole control law instead of demanding tuning of any original method control law to stabilize desired Unstable Periodic Orbits (UPO). In this research desired UPO is only p-1 (the fixed point, which represents the stable state). Original Time-Delay-Auto-Synchronization (TDAS) delayed feedback control method was used in this research as an inspiration for synthesizing a new feedback control law by means of evolutionary techniques and for preparation of sets of basic functions and operators for AP.

The original control method – TDAS has form (4) and its discrete form is given in (5).

$$F(t) = K\left[x(t-\tau) - x(t)\right] \tag{4}$$

$$F_n = K\left(x_{n-m} - x_n\right) \tag{5}$$

Where: K is adjustable constant, F is the perturbation, τ_d is a time delay; and m is the period of m-periodic orbit to be stabilized. The perturbation F_n in equation (5) may have arbitrarily large value, which can cause diverging of the system. Therefore, F_n should have a value between $-F_{\max}$, $F_{\max}$. In this work a suitable $F_{\max}$ value was taken from the previous research.

5 Used Soft-Computing Tools

This section gives the brief overview and the description of used soft-computing tools. This research utilized the symbolic regression tool, which is analytic programming and two evolutionary algorithms: Self-Organizing Migrating Algorithm [15]; and Differential Evolution [16].

Future simulations expect a usage of soft computing GAHC algorithm (modification of HC12) [17] and a CUDA implementation of HC12 algorithm [18].

5.1 Analytic Programming

Basic principles of the AP were developed in 2001. Until that time only genetic programming (GP) and grammatical evolution (GE) had existed. GP uses genetic algorithms while AP can be used with any evolutionary algorithm, independently on individual representation. AP represents synthesis of analytical solution by means of evolutionary algorithms. Various applications of AP are described in [19] - [22].

AP exists in 3 versions – basic without constant estimation, $\mathrm{AP_{nf}}$ – estimation by means of nonlinear fitting package in Mathematica environment and $\mathrm{AP_{meta}}$ – constant estimation by means of another evolutionary algorithms; meta means meta-evolution.

5.2 Self-Organizing Migrating Algorithm (SOMA)

Self-Organizing Migrating Algorithm is a stochastic optimization algorithm that is modeled on the basis of social behavior of cooperating individuals [15]. It was chosen because it has been proven that the algorithm has the ability to converge towards the global optimum [15] and due to the successful applications together with AP [23], [24].

5.3 Differential Evolution

DE is a population-based optimization method that works on real-number-coded individuals [25] - [27]. DE is quite robust, fast, and effective, with global optimization ability. It does not require the objective function to be differentiable, and it works well even with noisy and time-dependent objective functions. Description of used DERand1Bin strategy is presented in (6). Please refer to [16], [27] for the description of all other strategies.

$$u_{i,G+1} = x_{r1,G} + F \cdot \left(x_{r2,G} - x_{r3,G}\right) \tag{6}$$

6 Cost Function Design

The proposal of the basic cost function (CF) is in general based on the simplest CF, which could be used problem-free only for the stabilization of p-1 orbit. The idea was to minimize the area created by the difference between the required state and the real system output on the whole simulation interval – τ_i. This CF design is very convenient for the evolutionary searching process due to the relatively favorable CF surface. Nevertheless, this simple approach has one big disadvantage, which is the including of initial chaotic transient behavior of not stabilized system into the cost function value. As a result of this, the very tiny change of control method setting for extremely sensitive chaotic system causing very small change of CF value, can be suppressed by the above-mentioned including of initial chaotic transient behavior

But another universal cost function had to be used for stabilizing of extremely sensitive chaotic system and for having the possibility of adding penalization rules. It was synthesized from the simple CF and other terms were added.

This CF is in general based on searching for desired stabilized periodic orbit and thereafter calculation of the difference between desired and found actual periodic orbit on the short time interval - τ_s (40 iterations for higher order UPO) from the point, where the first minimal value of difference between desired and actual system output is found (i.e. floating window for minimization – see Fig. 2.).

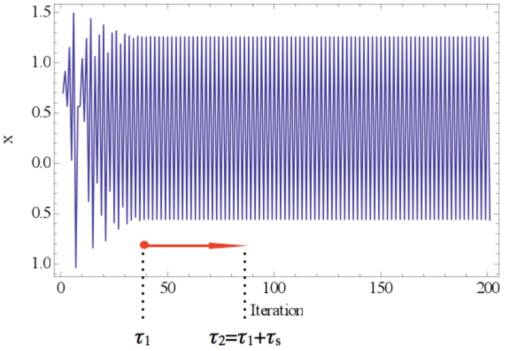


Fig. 2 “Floating window” for minimization

Such a design of universal CF should secure the successful stabilization of either p-1 orbit (stable state) or higher periodic orbits anywise phase shifted. Furthermore, due to CF values converging towards zero, this CF also allows the using of decision rules, avoiding very time demanding simulations. This rule stops EA immediately, when the first individual with good parameter structure is reached, thus the value of CF is lower then the acceptable (CF_{acc}) one. Based on the numerous experiments, typically $CF_{acc} = 0.001$ at time interval $\tau_s = 20$ iterations, thus the difference between desired and actual output has the value of 0.0005 per iteration – i.e. successful stabilization for the used control technique. The CF_{Basic} has the form (7):

$$CF_{Basic} = pen_1 + \sum_{t=\tau 1}^{\tau 2} \left| TS_t - AS_t \right|, \tag{7}$$

where:

TS - target state, AS - actual state

τ_1 - the first min value of difference between TS and AS

τ_2 – the end of optimization interval ($\tau_1 + \tau_s$)

$pen_1 = 0$ if $\tau_i - \tau_2 \geq \tau_s$; $pen_1 = 10*(\tau_i - \tau_2)$ if $\tau_i - \tau_2 < \tau_s$ (i.e. late stabilization).

Finally, the CF used within the evolutionary synthesis of a new robust control law is given in (8). It was evaluated as a sum of the three partial cost functions values each for one of the selected chaotic system.

$$CF_{Final} = CF_{Basic\_Logistic} + CF_{Basic\_Hénon} + CF_{Basic\_Lozi} \tag{8}$$

7 Results

Analytic Programming requires some EA for its run. In this paper, AP_{meta} version was used. Meta-evolutionary approach means usage of one main evolutionary algorithm for AP process and the second algorithm for coefficient estimation, thus to find optimal values of constants in the evolutionary synthesized control law.

SOMA algorithm was used for the main AP process and DE was used in the second evolutionary process. Settings of EA parameters for both processes given in Table 1 and Table 2 were based on performed numerous experiments with chaotic systems and simulations with AP_{meta}.

Table 1 SOMA settings for AP

SOMA Parameter	Value
PathLength	3
Step	0.11
PRT	0.1
PopSize	50
Migrations	4
Max. CF Evaluations (CFE)	5345

Table 2 DE settings for meta-evolution

DE Parameter	Value
PopSize	40
F	0.8
CR	0.8
Generations	150
Max. CF Evaluations (CFE)	6000

The data set for AP required only constants, operators like plus, minus, power and output values x_n and x_{n-1} .

Basic set of elementary functions for AP:

GFS2arg= +, -, /, *, ^
GFS0arg= $data_{n-1}$ to $data_n$, K

Total number of cost function evaluations for AP was 5345, for the second EA it was 6000, together 32.07 millions per each simulation. See Table 3 for simple final CF values statistic.

Table 3 Cost Function value and simple statistics

Statistical parameter	Value
Min	$4.7739•10^{-15}$
Max	$4.9491•10^{-7}$
Average	$5.4298•10^{-8}$
Median	$2.0880•10^{-10}$
Std. Deviation	$1.2619•10^{-7}$

Following description of selected experiment results contains illustrative examples of direct output from AP – synthesized control law without coefficients estimated (9); further the notation with simplification after estimation by means of second algorithm DE (10), Table 4 with corresponding CF statistics and the average error value between actual and required system output, and finally Fig. 3 with simulation results.

$$F_n = \frac{2x_n - 2x_{n-1}}{x_{n-1} + x_n - K_1 + K_2} \tag{9}$$

$$F_n = \frac{2x_n - 2x_{n-1}}{x_{n-1} + x_n + 1.94611} \tag{10}$$

Table 4 Experiment results attributes

Results attributes	Value
Final CF value	4.7739•10^{-15}
CF$_{\text{Logistic}}$ value	6.6613•10^{-16}
CF$_{\text{Hénon}}$ value	2.9976•10^{-15}
CF$_{\text{Lozi}}$ value	1.1102•10^{-15}
Average error per iteration	1.19•10^{-16}

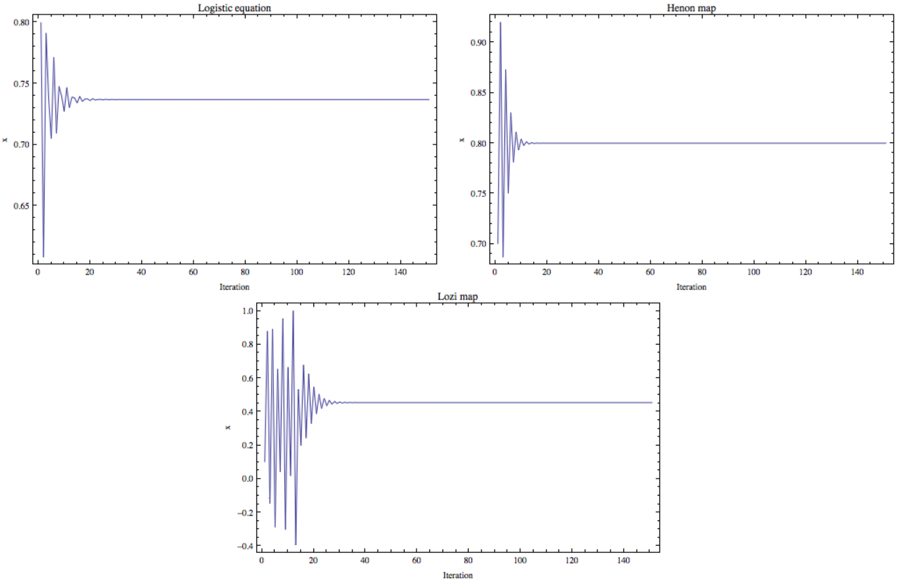


Fig. 3 Simulation results for a new synthesized robust feedback control law: Logistic equation (upper left), Hénon map (upper right) and Lozi map (below)

8 Conclusions

This paper deals with a synthesis of a new universal robust control law by means of AP for stabilization of selected discrete chaotic systems at fixed point. One-dimensional Logistic equation, Hénon map and Lozi map as the examples of two-dimensional discrete chaotic systems were used in this research.

Obtained results reinforce the argument that AP is able to solve this kind of difficult problems and to produce a new robust synthesized control law in a symbolic way securing desired behaviour and precise stabilization of the selected set of chaotic systems.

The future research will include the development of better cost functions, testing of different AP data sets, and performing of numerous simulations to obtain more results and produce better statistics, thus to confirm the robustness of this approach.

Acknowledgments. This work was supported by Grant Agency of the Czech Republic GACR P103/13/08195S; by the project Development of human resources in research and development of latest soft computing methods and their application in practice, reg. no. CZ.1.07/2.3.00/20.0072 funded by Operational Program Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic; by European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089 and by Internal Grant Agency of Tomas Bata University under the project No. IGA/FAI/2013/012.

References

[1] Liu, B., Wang, L., Jin, Y.H., Huang, D.X., Tang, F.: Control and synchronization of chaotic systems by differential evolution algorithm. Chaos, Solitons& Fractals 34(2), 412–419 (2007) ISSN 0960-0779

[2] Zelinka, I., Senkerik, R., Navratil, E.: Investigation on evolutionary optimization of chaos control. Chaos, Solitons& Fractals 40(1), 111–129 (2009)

[3] Senkerik, R., Zelinka, I., Davendra, D., Oplatkova, Z.: Utilization of SOMA and differential evolution for robust stabilization of chaotic Logistic equation. Computers & Mathematics with Applications 60(4), 1026–1037 (2010)

[4] Pyragas, K.: Control of chaos via extended delay feedback. Physics Letters A 206, 323–330 (1995)

[5] Aydin, I., Karakose, M., Akin, E.: Chaotic-based hybrid negative selection algorithm and its applications in fault and anomaly detection. Expert Systems with Applications 37(7), 5285–5294 (2010)

[6] Davendra, D., Zelinka, I., Senkerik, R.: Chaos driven evolutionary algorithms for the task of PID control. Computers & Mathematics with Applications 60(4), 1088–1104 (2010) ISSN 0898-1221

[7] Pluhacek, M., Senkerik, R., Davendra, D., Kominkova Oplatkova, Z., Zelinka, I.: On the behavior and performance of chaos driven PSO algorithm with inertia weight. Computers & Mathematics with Applications (article in press, 2013) ISSN 0898-1221, doi:10.1016/j.camwa.2013.01.016

[8] Just, W.: Principles of Time Delayed Feedback Control. In: Schuster, H.G. (ed.) Handbook of Chaos Control. Wiley-Vch (1999)

[9] Pyragas, K.: Continuous control of chaos by self-controlling feedback. Physics Letters A 170, 421–428 (1992)

[10] Senkerik, R., Oplatkova, Z., Zelinka, I., Davendra, D.: Synthesis of feedback controller for three selected chaotic systems by means of evolutionary techniques: Analytic programming. Mathematical and Computer Modelling 57(1-2), 895–7177 (2013)

[11] Kominkova Oplatkova, Z., Senkerik, R., Zelinka, I., Pluhacek, M.: Analytic programming in the task of evolutionary synthesis of a controller for high order oscillations stabilization of discrete chaotic systems. Computers & Mathematics with Applications (2013), ISSN 0898-1221, doi:10.1016/j.camwa.2013.02.008

[12] Hilborn, R.C.: Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers. Oxford University Press (2000) ISBN: 0-19-850723-2

[13] May, R.M.: Stability and Complexity in Model Ecosystems. Princeton University Press (2001) ISBN: 0-691-08861-6

[14] Sprott, J.C.: Chaos and Time-SeriesAnalysis. Oxford University Press (2003)

[15] Zelinka, I.: SOMA – Self Organizing Migrating Algorithm. In: Babu, B.V., Onwubolu, G. (eds.) New Optimization Techniques in Engineering. Springer (2004) ISBN 3-540-20167X

[16] Price, K., Storn, R.M., Lampinen, J.A.: Differential Evolution: A Practical Approach to Global Optimization. Springer (2005)

[17] Matousek, R., Zampachova, E.: Promising GAHC and HC12 algorithms in globaloptimizationtasks. Optimization Methods& Software 26(3), 405–419 (2011) ISSN 1055-6788

[18] Matousek, R.: HC12: ThePrincipleof CUDA Implementation. In: Proceedingsof 16th International Conference On Soft Computing Mendel 2010, pp. 303–308 (2010) ISBN 978-80-214-4120-0

[19] Zelinka, I., Guanrong, C., Celikovsky, S.: Chaos Synthesis by Means of Evolutionary algorithms. International Journal of Bifurcation and Chaos 18(4), 911–942 (2008)

[20] Oplatkova, Z., Zelinka, I.: Investigation on Evolutionary Synthesis of Movement Commands. Modelling and Simulation in Engineering 2009, Article ID 845080, 12 pages (2009) ISSN: 1687-559

[21] Zelinka, I., Davendra, D., Senkerik, R., Jasek, R., Oplatkova, Z.: Analytical Programming - a Novel Approach for Evolutionary Synthesis of Symbolic Structures. In: Kita, E. (ed.) Evolutionary Algorithms. InTech (2011)

[22] Chramcov, B., Varacha, P.: UsageoftheEvolutionaryDesignedNeural Network for Heat DemandForecast. In: ProceedingsofNostradamus 2012: ModernMethodsofPrediction, Modeling and Analysisof Nonlinear Systems, pp. 103–122 (2013) ISBN 978-3-642-33226-5

[23] Varacha, P., Jasek, R.: ANN Synthesis for an Agglomeration Heating Power Consumption Approximation. In: Recent Researches in Automatic Control, pp. 239–244. WSEAS Press, Montreux, ISBN 978-1-61804-004-6

[24] Varacha, P., Zelinka, I.: Distributed Self-Organizing Migrating Algorithm Application and Evolutionary Scanning. In: Proceedings of the 22nd European Conference on Modelling and Simulation, ECMS 2008, pp. 201–206 (2008) ISBN 0-9553018-5-8

[25] Lampinen, J., Zelinka, I.: New Ideas in Optimization – Mechanical Engineering Design Optimization by Differential Evolution, vol. 1, 20p. McGraw-Hill, London (1999) ISBN 007-709506-5

[26] Price, K.: An Introduction to Differential Evolution. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, pp. 79–108. McGraw-Hill, London (1999)

[27] Price, K., Storn, R.: Differential evolution homepage (2001), `http://www.icsi.berkeley.edu/~storn/code.html` (accessed March 01, 2013)

Chaos Powered Selected Evolutionary Algorithms

Lenka Skanderova, Ivan Zelinka, and Petr Šaloun

Abstract. It is well known that the evolution algorithms use pseudo-random numbers generators for example to generate random individuals in the space of possible solutions, crossing etc. In this paper we are dealing with the effect of different pseudo-random numbers generators on the course of evolution and the speed of their convergence to the global minimum. From evolution algorithms the differential evolution and self organizing migrating algorithm have been chosen because they have different strategies. As the random generators Mersenne Twister and chaotic system - logistic map have been used.

1 Introduction

Evolution algorithms are based on the Darwin's and Mendel's principles. There is the population of individuals, which is improving in the time. The individuals are crossing and mutated and the best survive, while worse die. The population is developing during the generation cycles, we call them *Generations* or *Migrations* according to the used algorithm [1]. From the view of pseudo-random numbers generators, we need to generate individuals in the space of possible solutions. The individual consists of parameters (each parameter has its low and high bound) and these parameters are usually real numbers. In the first generation cycle the individuals are chosen randomly - their parameters are chosen randomly in their low and high borders. Next the pseudo-random numbers generators plays the essential role in the process of crossing.

In this paper we connect together three basic research areas – evolution algorithms, chaos and pseudo-random numbers generators. With connection

Lenka Skanderova · Ivan Zelinka · Petr Šaloun
Department of Computer Science, VSB-Technical university Ostrava,
17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic
e-mail: {lenka.skanderova,ivan.zelinka,petr.saloun}@vsb.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 111–124.
DOI: 10.1007/978-3-319-00542-3_12

of evolutionary algorithms (especially differential evolution and SOMA) and chaos [2] - [4] have been written. Artificial bee colony algorithm is connected with chaos in [5]. New adaptive differential evolution technique based on logistic map for optimal distribution placement and sizing is presented in [7] and in [6]the differential evolution with chaos theory for self adaptation of differential evolution's parameters is combined. In [8] chaotic Logistic equation is mentioned in connection with SOMA and differential evolution. In recent years, some new pseudo-random number generators were described, e.g. [11] and [12]. The pseudo-random numbers generators based on chaos are presented in [9], [10]. In [13] Sandpile model - the complex system operating at a critical state between chaos and order - is proposed and author state that cellular automata can be used as a generator of pseudo-random numbers.

This paper is divided into the sections, where in the first one we describe the used evolution algorithms - DE and SOMA, next we are dealing with used pseudo-random numbers generators, motivation and design experiments. In section Results, the experiment's results are shown and in Conclusions we sumarize achieved results.

2 Evolution Algorithms

2.1 Differential Evolution (DE)

Differential evolution works at the principle of improving population during the generation's cycles, as it is mentioned above. The population consists of individuals, each individual has its own parameters and fitness value – the value of the cost function. The fitness says how good this individual is in the population. In generation cycle for each individual three different individuals from the population are chosen randomly and the noise vector is generated, see Eq.(1.

$$v_i^{G+1} = x_{r1}^G + F(x_{r2}^G - x_{r3}^G), r_1 \neq r_2 \neq r_3 \tag{1}$$

where v_i^{G+1} is the i-th noise vector in the next generation, $x_{\mathbf{r1}}$ is the first randomly chosen individual, $x_{\mathbf{r2}}^G$ is the second randomly chosen individual and $x_{\mathbf{r3}}^G$ is the third randomly chosen individual. F is the mutation constant [14].

When the noise vector is made, the trial individual creation can start. To the trial individual the parameters from the noise vector or from the actual individual are chosen according to Eq.(2).

$$u_{i,G+1}^j = \begin{cases} v_{i,G+1}^j & if\ r(j) \leq CR \\ x_{i,G}^j & otherwise \end{cases} \tag{2}$$

where $u_{i,G+1}$ is the j-th parameter of the trial vector, $r(j)$ is the random number from the interval $[0, 1]$ [14] and CR is the crossover probability.

The selection of a new individual is described by Eq.(3).

$$x_{i,G+1} = \begin{cases} u_{i,G+1} & if\ f(u_{i,G+1}) \prec f(x_{i,G}) \\ x_{i,G} & otherwise \end{cases} \tag{3}$$

where $f(u_{i,G+1})$ is the fitness of the trial vector and $f(x_{i,G})$ denote fitness of the actual individual. The sign $\prec$ is used because we search global minimum [14].

2.2 Self Organized Migrating Algorithm (SOMA)

The strategy of SOMA differs from DE. In DE new offspring is creating during the evolution. In SOMA there is no offspring. The individuals migrate in the space of possible solutions, they just change their positions. In this paper version *AllToOne* of SOMA is used, where all individuals migrate to the one individual, we call it *Leader*. Except this strategy, *AllToAll*, *AllToOne Random*, *AllToAll Adaptive* and *AllToOne Adaptive* exist.

The begin of the algorithm is the same like in DE – the random individuals are generated in population. Following principle is different: Each individual migrates to the *Leader* in steps (the length of all step is united), while the sum of steps do not reach or even cross the parameter denoted as *PathLength*, see Eq.(4). In SOMA the step is denoted by the parameter *Step* and its value should be odd, usually it is the value 0.11. For each step, where individual reaches a new position, new fitness is computed. In the end of migration cycle the best reached position is chosen.

Mutation is replaced by perturbation in SOMA, see Eq.4.

$$\mathbf{r} = \mathbf{r_0} + \mathbf{m}t\mathbf{PRTvector} \tag{4}$$

where $\mathbf{r}$ is a new candidate solution, $\mathbf{r_0}$ denotes actual individual, m is a difference between *Leader* and start position of individual and t is the parameter *Step*, $t \in [0, PathLength]$ [15].

The direction of the individual's migration is given by the parameter PRT, ussually with the value 0.1. According to the PRT the $PRTvector$ is generated by this way: for each parameter of $PRTvector$ a random number from interval [0,1] is generated. If this number is smaller than PRT, the parameter of $PRTvector$ will have value 1 else it will have value 0, see Eq.(5).

$$if\ rnd_j \prec PRT\ then\ PRTvector_j = 1\ else\ 0 \tag{5}$$

where rnd_j means random number from the interval [0,1], $PRTvector_j$ denoted the j-th parameter of the perturbation vector. If the fitness of the best found position od the individual is better than its actual fitness, the individual will migrate to the best position. Otherwise individual stays at its old position.

3 Pseudo-random Numbers Generator

3.1 Mersenne Twister (MT)

Mersenne Twister has been proposed by M. Matsumoto and T. Nishimura in the year 1997. Its period is $2^{19937} - 1$ and it has 632-dimensional equidistribution property. It is the variant of previously proposed generators, TGFSR. For more information see [16].

MT has been used for example in the Monte Carlo Localization Algorithm in [17]. It has been also used as a comparative tool in the developement of new pseudo-random numbers generators, see [19]. In [18] authors state that MT cannot be used efficiently without substantial changes as a random number generator for massively parallel simulations on GPU. In connection with evolution algorithm MT has been used in [20], where autors are specialized in genetic algorithms and simulated annealing, [21] in parallel evolutionary algorithm for RNA holding and [22] in Particle Swarm Optimization.

3.2 Logistic Map

Each definition of chaos describes some kind of unpredictability in the evolution of the system. At this idea theory of Li and Yorke is based, this theory says that in the logistic map in interval [0,1] it is possible to find two close trajectories, which are moving away from each other with increasing time. In other words small change in starting conditions may cause very different results. The system can behave equally if and only if the starting conditions are absolutely same [33]. Good example is the butterfly effect.

In connection with chaos control [24] and [27] have been written. In 2013 [25] and [26] have been written about chaos and evolution algorithms connection.

Logistic map is a one – dimensional quadratic map defined by Eq.6.

$$x_{n+1} = ax_n(1 - x_n) \tag{6}$$

where a is an external parameter and x_n value moves in interval [0,1] [28].

Kuznetsov N.V. and Leonov G.A. in [34] say about Lyapunov exponent: "*In 1930 O. Perron found the effects of Lyapunov exponent sign inversion. It has been shown that the negativeness of the largest Lyapunov exponent of the first approximation system does not always result in the stability of zero solution of the original system. Small neighborhood of zero solution, the solutions of the original system with positive Lyapunov exponent (Lyapunov characteristic exponent) can exist. A. M. Lyapunov has introduced the notion of regular linear system and showed that for regular linearizations the stability is defined by the negativeness of Lyapunov exponents of linearized system - that was the first sufficient condition of asymptotic stability by the first approximation for nonstationary linearizations.*"

We can say that Lyapunov exponent is the basic tool for dynamic system description. If the Lyapunov exponent is negative, the dynamic system is not sensitive to beginning conditions. On the other hand if the Lyapunov exponent is positive, the system will be sensitive to basic condition. The chaotic system must have at least one positive exponent [36].

The behavior of the logistic map is described in [28] where authors investigate behavior of logistic map for $x \geq x_\infty$, they say that Lyapunov exponent λ (it characterizes the rate of separation of infinitesimally close trajectories) of the logistic map at x_∞ is zero. Lyapunov exponent λ becomes mostly possitive for $x > x_\infty$ and therefore authors say that chaos starts at the end of the bifurcation region, see Fig.1. Next authors state that: "*the detailed behavior of the iterates of the logistic map appears rather complicated in this region, it shows regularities which are again dictated by doubling operator and therefor universal. For $x_\infty \prec r$, periodic and chaotic regions are densely interwoven, and one finds a sensitive dependence on the parameter value*".

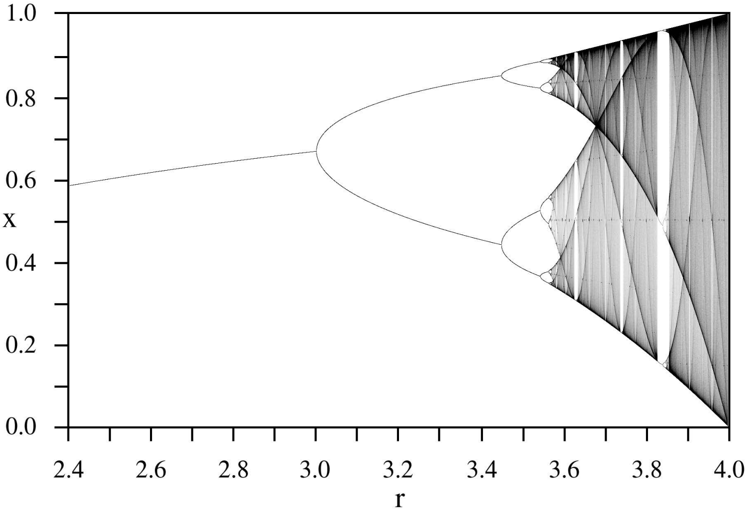


Fig. 1 Logistic map

In [29] the logistic map has been used for example to generate cycle time in series of signals, in cryptography - Baptista's cryptosystem, [30], and image encryption [31]. From the view of connection between evolution algorithms logistic map is mentioned for example in [32].

4 Motivation

The main motivation was to compare pseudo-random numbers generators from the view of influence to developing of the population in evolutionary algorithms. Mersenne Twister has been chosen because it has a big period

$(2^{19937} - 1)$ and as it was mentioned above it is very often connected with evolution algorithms as a confirmed random number generator.

5 Experiment Design

For experiments, SOMA and DE have been chosen because these algorithms have different strategies. As it was mentioned above, DE's population is still developing during the generation cycles, while in SOMA the individuals are migrating in the space of possible solutions and no offspring is generated. Each experiment in the experiment group has been repeated one hundred times and exact settings of both algorithms are mentioned in Tabs. 1 and 2 where NP means the number of individuals in one population, D dimension (how many parameters will be contained in one individual), $Migrations$ and $Generations$ denote the number of evolution cycles.

For experiments HP Pavilion dv7-6050 with processor Intel Core i7 with frequency 2 GHz, 4 GB RAM and graphic card AMD Radeon HD 6770M and Microsoft Visual Studio 2010 have been used. The experiments have beed processed by Mathematica 8.

Table 1 DE setting

NP	1000
D	20
$Generations$	500
F	0.8
CR	0.5

Table 2 SOMA setting

NP	1000
D	20
$Migrations$	500
PRT	0.11
$PathLength$	3
$Step$	0.11

As the trial functions 1^{st} de Jong's function, Schwefel's function and Ranna's function have been chosen. Schwefel's function has many local minimums, it is very jagged and we know the global minimum - $f(x) = -418.9829D$. 1^{st} de Jong's function has just one minimum - global minimum - $f(x) = 0$ and this function is not jagged. And in Ranna's function there the global minimum has not been found. And it is not mentioned in any literature.

Table 3 Setting of logistic map

Experiment's group	Value of parameter a
1^{st} group	3.58
2^{nd} group	[3.8280, 3.8285]
3^{rd} group	3.855
4^{th} group	[3.8567, 3.8570]
5^{th} group	4

In logistic map of chaos as the beginning value of x_n 0.02 has been set. This value has been chosen randomly. Each algorithm's experiments have been divided into 5 groups according setting of parameter a, see Tab. 3. 2^{nd} and 4^{th} groups are special, because the parameter a has been changing for each x_n by the step 0.0001. Otherwise the parameter a has been constant.

6 Results

The Figs. 2 and 3 show comparing of results of MT and Chaos random number generators, where in chaos $a = 4$ and Ranna's function has been used as a testing function. It is known that if $a = 4$ logistic equation will generate numbers from all interval [0,1], see 1. This fact is important for evolution developing. As it is obvious, evolutions where chaos pseudo-random number generator has been used, convergate faster than evolutions, where MT has been used. On the other hand when $a = 3.58$ and 1^{st} de Jong's and Schwefel's functions have been testing function, some experiments from the collection, where chaos has been used convergate much slower than MT and the total results have been much worse than MT.

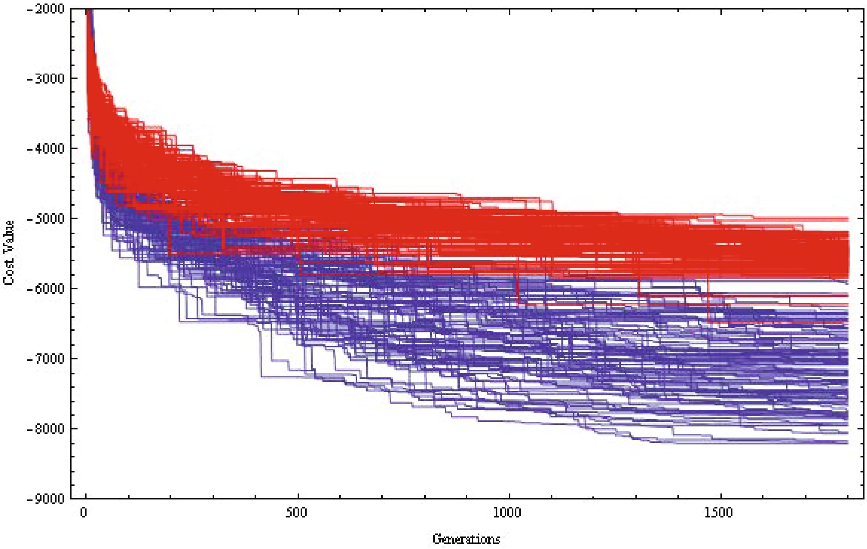


Fig. 2 Comparing MT and Chaos. Differential evolution, Ranna's function, $a = 4$. Blue represents Chaos, red represents MT.

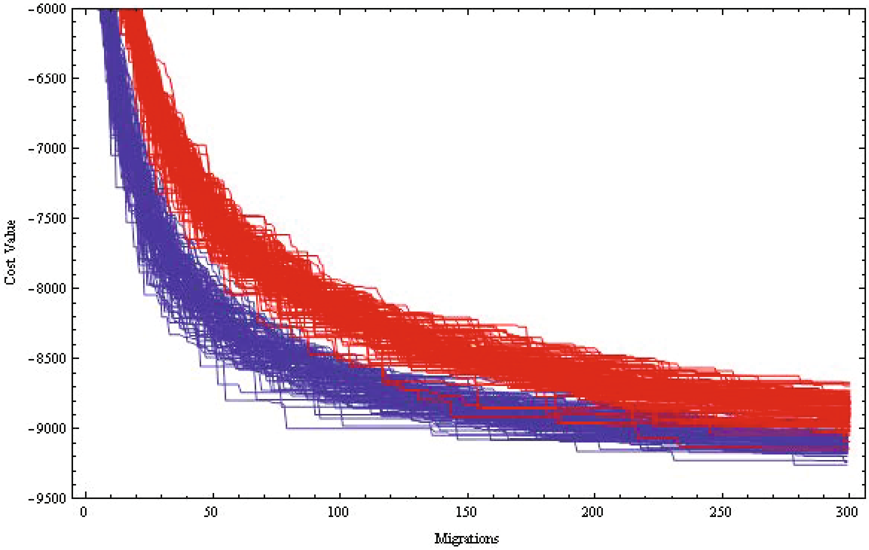


Fig. 3 Comparing MT and Chaos. SOMA, Ranna's function, $a = 4$. Blue represents Chaos, red represents MT.

Table 4 Minimum, maximu and average fitness value of DE experiments for 1^{st} de Jong's, Ranna's and Schwefel's function with settings mentioned in Tab. 1.

	Function	Min. fitness value	Max. fitness value	Average fitness value
Chaos $a = 3.58$	1^{st} de Jong's	0.000	0.000	0.000
	Ranna's	-7669.411	-5426.377	-6656.126
	Schwefel's	-8379.658	-7539.422	-8160.376
Chaos $a = 3.828$	1^{st} de Jong's	11.674	15220.172	4206.105
	Ranna's	-8081.707	-5125.884	-6894.176
	Schwefel's	-7763.678	-6167.574	-7107.649
Chaos $a = 3.855$	1^{st} de Jong's	0.003	0.074	0.023
	Ranna's	-8585.980	-6383.648	-7730.503
	Schwefel's	-8379.657	-7712.970	-8231.763
Chaos $a = 3.8567$	1^{st} de Jong's	0.004	0.121	0.026
	Ranna's	-8494.416	-6136.087	-7687.413
	Schwefel's	-8379.657	-7692.009	-8221.569
Chaos $a = 4$	1^{st} de Jong's	0.008	0.144	0.039
	Ranna's	-8732.198	-6175.018	-7860.793
	Schwefel's	-8379.651	-7882.383	-8234.940
MT	1^{st} de Jong's	0.000	0.003	0.000
	Ranna's	-8013.129	-6213.274	-7052.287
	Schwefel's	-8379.656	-7144.871	-7976.479

Table 5 Minimum, maximu and average fitness value of SOMA experiments for 1^{st} de Jong's, Ranna's and Schwefel's function with settings mentioned in Tab. 2.

	Function	Min. fitness value	Max. fitness value	Average fitness value
	1^{st} de Jong's	0.000	0.000	0.000
Chaos $a = 3.58$	Ranna's	-8588.349	-6946.222	-7716.781
	Schwefel's	-8379.658	-6543.776	-7277.547
	1^{st} de Jong's	0.000	0.000	0.000
Chaos $a = 3.828$	Ranna's	-9180.587	-8415.158	-8857.963
	Schwefel's	-8379.658	-8379.658	-8379.658
	1^{st} de Jong's	0.000	0.000	0.000
Chaos $a = 3.855$	Ranna's	-9110.511	-8435.843	-8830.062
	Schwefel's	-8379.658	-8379.658	-8379.658
	1^{st} de Jong's	0.000	0.000	0.000
Chaos $a = 3.8567$	Ranna's	-9264.553	-8535.640	-8887.881
	Schwefel's	-8379.658	-8379.657	-8379.658
	1^{st} de Jong's	0.000	0.000	0.000
Chaos $a = 4$	Ranna's	-9258.219	-8947.286	-9090.472
	Schwefel's	-8379.658	-8379.658	-8379.658
	1^{st} de Jong's	0.000	0.000	0.000
MT	Ranna's	-9139.584	-8667.654	-8866.831
	Schwefel's	-8379.658	8379.658	8379.658

7 Conclusion

From the experiments we can make some conclusions:

- It depends on the parameter a in logistic equation. When $a = 4$ the evolution convergence has been faster with using chaos pseudo-random numbers generator than MT. When a has been from interval [3.828, 3.8285], $a = 3.855$ and a has been from interval [3.8567, 3.857] chaos random number generator and MT results have been comparable except 1^{st} de Jong's function, which converged faster with MT using. If $a = 3.58$ the results have been much worse in chaos than in MT. It is logical, because in this area in the bifurcation diagram we can see deterministic windows (white areas), see 1. That means many numbers from interval [0,1] have not been contained in the computation. This area is weighted by a big periodicity, that influence the results negatively. Much better results with using chaos have been reached when $a = 4$, see Figs. 2 and 3. It is the area of chaos in bifurcation diagram and random numbers are generated from all interval [0,1].
- In chaos pseudo-random numbers generator some experiments differ from total average by their convergence trajectories, while in MT there was no experiment, which would be much different than others from the same group.
- In Tabs. 4 and 5 minimum, maximum and average cost function values of the best individuals in evolution are shown for DE and SOMA, where MT and Chaos - logistic map as the pseudo-random numbers generators

have been used. As it was mentioned above 1^{st} de Jong's, Ranna's and Schwefel's function have ben used as test functions. Minimums in Tabs. 4 and 5 are the best values of the cost function, maximums are the worst values of the cost function, which have been reached during the evolution process.
- There is a big difference between SOMA and DE in 1^{st} de Jong's function's searched minimum, SOMA has got a smaller value than DE, while in the other functions the minimum values are comparable, and it does not matter if MT or Chaos - logistic map has been used.
- In Figs. 2 and 3 it is shown that if $a = 4$ in logistic equation and the Ranna's function has been choosen, evolution converged faster than when MT has been used as a pseudo-random numbers generator.

In the future the experiments will be extended by next kinds of SOMA - *AllToAll*, *AllToOne Random*, *AllToAll Adaptive* and *AllToOne Adaptive* as well as some other kinds of DE. As testing functions next testing functions (e.g. Rastrigin's, Ackley's, Michalewicz's) will be tried. Further research will be focused on more extensive and intensive testing of our ideas proposed here. Our aim is to try algorithms like scatter search [46], evolutionary strategies [47], genetic algorithms [48], [52] or particle swarm [49]. Also novel algorithms will be tested for its performance under our proposed approach in [50], [51] and alternative methods of symbolic regression [53].

Wider class of different algorithms, test functions and deterministic processes will be selected for future experiments to prove and specify the domain of validity of our ideas proposed here.

Acknowledgements. The following two grants are acknowledged for the financial support provided for this research: Grant Agency of the Czech Republic - GACR P103/13/08195S, by the Development of human resources in research and development of latest soft computing methods and their application in practice project, reg. no. CZ.1.07/2.3.00/20.0072 funded by Operational Programme Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic and by grant SV 4603351.

References

1. Zelinka, I., et al.: Evolutionary Algorithms and Chaotic Systems. SCI. Springer (2010) ISBN-10: 3642107060, ISBN-13: 978-364210706
2. Zelinka, I.: On evolutionary synthesis of chaotic systems. In: Zelinka, I., Snasel, V., Rössler, O.E., Abraham, A., Corchado, E.S. (eds.) Nostradamus: Mod. Meth. of Prediction, Modeling. AISC, vol. 192, pp. 29–34. Springer, Heidelberg (2013)
3. Brandejsky, T., Zelinka, I.: Specific behaviour of GPA-ES evolutionary system observed in deterministic chaos regression. In: Zelinka, I., Snasel, V., Rössler, O.E., Abraham, A., Corchado, E.S. (eds.) Nostradamus: Mod. Meth. of Prediction, Modeling. AISC, vol. 192, pp. 73–81. Springer, Heidelberg (2013)

4. Pluhacek, M., Senkerik, R., Zelinka, I.: Impact of various chaotic maps on the performance of chaos enhanced PSO algorithm with inertia weight – an initial study. In: Zelinka, I., Snasel, V., Rössler, O.E., Abraham, A., Corchado, E.S. (eds.) Nostradamus: Mod. Meth. of Prediction, Modeling. AISC, vol. 192, pp. 153–166. Springer, Heidelberg (2013)
5. Tien, J.P., Li, T.H.S.: Hybrid Taguchi-chaos of multilevel immune and the artificial bee colony algorithm for parameter identification of chaotic systems. Computer & Mathematics with Applications 64, 1108–1119 (2012)
6. Manal, K.K., et al.: Emission Constrained Economic Dispatch Using Logistic Map Adaptive Differential Evolution. In: Proceedings of the International Conference on Information Systems Design and Intelligent Applications 2012, vol. 132, pp. 387–394 (2012)
7. Mandal, K.K., Bhattacharya, B., Tudu, B., Chakraborty, N.: Logistic Map Adaptive Differential Evolution for Optimal Capacitor Placement and Sizing. In: Panigrahi, B.K., Suganthan, P.N., Das, S., Satapathy, S.C. (eds.) SEMCCO 2011, Part I. LNCS, vol. 7076, pp. 68–76. Springer, Heidelberg (2011)
8. Senkerik, R., et al.: Utilization of SOMA and differential evolution for robust stabilization of chaotic Logistic equation. 3rd Global Conference on Power Control Optimization. Computers & Mathematics with Applications 60, 1026–1037 (2010)
9. Hu, H.P., et al.: Pseudorandom sequence generator based on the Chen chaotic system. Computer Physicics Communications 184, 765–768 (2013)
10. Wang, X.Y., Qin, X.: A new pseudo-random number generator based on CML and chaotic iteration. Nonlinear Dynamics 70, 1589–1592 (2012)
11. Song, H.L.: New Pseudorandom Number Generator Artin-Schreier Tower for p=5. China Communications 9, 60–67 (2012)
12. Marquardt, P., et al.: Pseudorandom number generators based on random covers for finite groups. Designs Codes and Cryptography 64, 209–220 (2012)
13. Karimi, H., et al.: On the combination of self-organized systems to generate pseudo-random numbers. Information Science 221, 371–388 (2013)
14. Zhou, Y., Li, X., Gao, L.: A differential evolution algorithm with intersect mutation operator. Applied Soft Computing 13, 390–401 (2013)
15. Nolle, L., Zelinka, I., Hopgood, A.A., Goodyear, A.: Comparison of an self-organizing migration algorithm with simulated annealing and differential evolution for automated waveform tuning
16. Matsumoto, M., Nishimura, T.: Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator. ACM Transactions on Modeling and Computer Simulation 8, 3–30 (1998)
17. Bonato, V., et al.: A Mersenne Twister Hardware Implementation for the Monte Carlo Localization Algorithm. Journal of Signal Processing Systemsfor Signal, Image, and Video Technology (formerly the Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology) (2012)
18. Manssen, M., et al.: Random number generators for massively parallel simulations on GPU. The European Physical Journal Special Topics, EDP Sciences, 53–71 (2012)
19. Leiserson, et al.: Deterministic Parallel Random-Number Generation for Dynamic-Multithreading Platforms. In: 17th ACM SIGPLAN symposium on Principles and Practice of Parallel Programming, pp. 193–204. ACM, New York (2012)

20. Maucher, M., Schning, U., Kestler, H.A.: Search heuristics and the influence of non-perfect randomness: examining Genetic Algorithms and Simulated Annealing. Springer (2011)
21. Wiese, K.C., et al.: P-RnaPredict - A parallel evolutionary algorithm for RNA folding: Effects of pseudorandom number quality. IEEE Transactions on Nanobioscience 4, 219–227 (2005)
22. Igarashi, J., Sonoh, S., Koga, T.: Particle Swarm Optimization with SIMD-Oriented Fast Mersenne Twister on the Cell Broadband Engine. In: Köppen, M., Kasabov, N., Coghill, G. (eds.) ICONIP 2008, Part II. LNCS, vol. 5507, pp. 1065–1071. Springer, Heidelberg (2009)
23. http://msdn.microsoft.com/en-us/library/system.random.aspx
24. Hegazi, A.S., et al.: On chaos control and synchronization of the commensurate fractional order Liu system. Communications in Nonlinear Science and Numerical Simulation 18, 1193–1202 (2013)
25. Senkerik, R.: On the Evolutionary Optimization of Chaos Control - A Brief Survey. Nostradamus: Modern Methods of Prediction, Modeling and Analysis of Nonlinear Systems 192, 35–48 (2013)
26. Senkerik, R., Davendra, D., Zelinka, I., Oplatkova, Z., Pluhacek, M.: Optimization of the batch reactor by means of chaos driven differential evolution. In: Snasel, V., Abraham, A., Corchado, E.S. (eds.) SOCO Models in Industrial & Environmental Appl. AISC, vol. 188, pp. 93–102. Springer, Heidelberg (2013)
27. Chen, D.Y., et al.: Control and synchronization of chaos in an induction motor system. International Journal of Innovative Computing Information and Control 8, 7237–7248 (2012)
28. Schuster, H.G., Just, W.: Deterministic Chaos An Introduction. Wiley-VCH Verlag GmbH & Co., KGaA, Weinheim (2005)
29. Nagatani, T., Sugiyama, N.: Vehicular traffic flow through a series of signals with cycle time generated by a logistic map. Physica A: Statistical Mechanics and its Applications 392, 851–856 (2013)
30. Hussain, I., et al.: An efficient approach for the construction of LFT S-boxes using chaotic logistic map. Nonlinear Dynamics 71, 133–140 (2013)
31. Akhshani, A., et al.: An image encryption scheme based on quantum logistic map. Communications in Nonlinear Science and Numerical Simulation 17, 4653–4661 (2012)
32. He, Y.Y., et al.: A fuzzy clustering iterative model using chaotic differential evolution algorithm for evaluating flood disaster. Expert Systems with Applications 38, 10060–10065 (2011)
33. Wu, X., Zhu, P.: Chaos in a class of non-autonomous discrete systems. Applied Mathematics Letters 26, 431–436 (2013)
34. Kuznetsov, N.V., Leonov, G.A.: On stability by the first approximation for discrete systems. In: Proceedings of International Conference on Physics and Control, PhysCon 2005, vol. 2005, pp. 596–599 (2005)
35. Leonov, G.A., Kuznetsov, N.V.: Time-Varying Linearization and the Perron effects. International Journal of Bifurcation and Chaos 17, 1079–1107 (2007)
36. Kaclek, J., Mca, I.: Nelinern analza a predikce stovho provozu Elektrorevue (2009) ISSN 1213 – 1539

37. Pluhacek, M., et al.: On the Behaviour and Performance of Chaos Driven PSO Algorithm with Inertia Weight. Computers & Mathematics with Applications (2012) (accepted for publication) ISSN 0898-1221
38. Pluhacek, M., Budikova, V., Senkerik, R., Oplatkova, Z., Zelinka, I.: Extended initial study on the performance of enhanced PSO algorithm with lozi chaotic map. In: Zelinka, I., Snasel, V., Rössler, O.E., Abraham, A., Corchado, E.S. (eds.) Nostradamus: Mod. Meth. of Prediction, Modeling. AISC, vol. 192, pp. 167–177. Springer, Heidelberg (2013)
39. Pluhacek, M., Senkerik, R., Zelinka, I.: Impact of various chaotic maps on the performance of chaos enhanced PSO algorithm with inertia weight – an initial study. In: Zelinka, I., Snasel, V., Rössler, O.E., Abraham, A., Corchado, E.S. (eds.) Nostradamus: Mod. Meth. of Prediction, Modeling. AISC, vol. 192, pp. 153–166. Springer, Heidelberg (2013)
40. Pluhacek, M., Senkerik, R., Davendra, D., Zelinka, I.: Designing PID controller for DC motor by means of enhanced PSO algorithm with dissipative chaotic map. In: Snasel, V., Abraham, A., Corchado, E.S. (eds.) SOCO Models in Industrial & Environmental Appl. AISC, vol. 188, pp. 475–483. Springer, Heidelberg (2013)
41. Pluhacek, M., et al.: PID Controller Design For 4th Order system By Means Of Enhanced PSO algorithm With Lozi Chaotic Map. In: Proceedings of 18th International Conference on Soft Computing, MENDEL 2012, pp. 35–39 (2012) ISBN 978-80-214-4540-6
42. Pluhacek, M., et al.: On The Performance Of Enhanced PSO algorithm With Lozi Chaotic Map –An Initial Study. In: Proceedings of 18th International Conference on Soft Computing, MENDEL 2012, pp. 40–45 (2012) ISBN 978-80-214-4540-6
43. Pluhacek, M., et al.: Designing PID Controller For DC Motor System By Means Of Enhanced PSO Algorithm With Discrete Chaotic Lozi Map. In: Proceedings of 26th European Conference on Modelling and Simulation, ECMS 2012, pp. 405–409 (2012) ISBN 978-0-9564944-4-3
44. Pluhacek, M., et al.: Designing PID Controller for 4th Order System By Means of Enhanced PSO Algorithm with Discrete Chaotic Dissipative Standard Map. In: Proceedings of 24th European Modeling & Simulation Symposium, EMSS 2012, pp. 396–401 (2012) ISBN 978-88-97999-09-6
45. Pluhacek, M., et al.: On the Performance of Enhanced PSO Algorithm with Lozi Chaotic Map. In: Application of Modern Methods of Prediction, Modeling and Analysis of Nonlinear Systems. SCI, vol. 1, p. 18. Springer (November 2012) (accepted for publication) ISSN: 1860-949X
46. Glover, F., Laguna, M., Mart, R.: Scatter Search. In: Ghosh, A., Tsutsui, S. (eds.) Advances in Evolutionary Computation: Theory and Applications, pp. 519–537. Springer, New York (2003)
47. Beyer, H.-G.: Theory of Evolution Strategies. Springer, New York (2001)
48. Holland, J.H.: Genetic Algorithms. Scientific American, 44–50 (July 1992)
49. Clerc, M.: Particle Swarm Optimization. ISTE Publishing Company (2006) ISBN 1905209045
50. Matousek, R.: HC12: The Principle of CUDA Implementation. In: Matousek (ed.) 16th International Conference on Soft Computing, MENDEL 2010, Brno, pp. 303–308 (2010)

51. Matousek, R., Zampachova, E.: Promising GAHC and HC12 algorithms in global optimization tasks. Journal Optimization Methods & Software 26(3), 405–419 (2011)
52. Matousek, R.: GAHC: Improved Genetic Algorithm. In: Krasnogor, N., Nicosia, G., Pavone, M., Pelta, D. (eds.) Nature Inspired Cooperative Strategies for Optimization (NICSO 2007). SCI, vol. 129, pp. 507–520. Springer, Heidelberg (2008)
53. Zelinka, I., Davendra, D., Senkerik, R., Jasek, R., Oplatkova, Z.: Analytical Programming - a Novel Approach for Evolutionary Synthesis of Symbolic Structures. In: Kita, E. (ed.) Evolutionary Algorithms. InTech (2011) ISBN: 978-953-307-171-8, http://www.intechopen.com/books/evolutionary-algorithms/analytical-programming-a-novel-approach-for-evolutionary-synthesis-of-symbolic-structures , doi:10.5772/16166

Case Study of Evolutionary Process Visualization Using Complex Networks

Patrik Dubec, Jan Plucar, and Lukáš Rapant

Abstract. This paper presents a case study of visualization of evolutionary process using complex network. Our previous research focused on application of evolutionary algorithms on finding global minimum of energetic function obtained in Force-directed graph drawing algorithm. This research has been combined with novel method for visualization of Differential Evolution (DE) and Self-Organizing Migration Algorithm (SOMA) process. We have developed and run our own algorithms, visualized and analyzed evolutionary complex networks obtained from their process. This paper presents improvements to the evolutionary network visualization by observing changes of some of the complex network properties during evolution. We also propose further improvements to the evolutionary network visualization.

Keywords: Force-based layout algorithm, Differential evolution, Soma, Complex network, Complex network analysis.

1 Introduction

Evolutionary algorithms (EA) have been widely used for solving or optimizing problems from wide range of fields. Example given, we can find their usage in solving nonlinear global optimization problems [1], in chemical engineering - optimization of a copolymerization reactor [2] and in the field of power and energy systems for loss minimization of such systems [3]. In a of force-directed graph layout algorithms field, we have tested bio-inspired approach in a process of finding global minimum of an energetic function, which is the key function for correct node placement in a graph. Force-directed algorithms calculate the layout of a graph using only information contained within the structure of the graph itself. Graphs drawn with these algorithms tend to be easily readable, simple, intuitive

Patrik Dubec, · Jan Plucar · Lukáš Rapant
VSB-Technical University of Ostrava, 17. listopadu 15, 70833 Ostrava-Poruba,
Czech Republic
e-mail: {patrik.dubec,jan.plucar,lukas.rapant}@vsb.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 125–135.
DOI: 10.1007/978-3-319-00542-3_13

and exhibit symmetries. However, there are also two distinctive disadvantages: poor local minima, high running time.

In this paper we would like to connect two areas of research: bio-inspired algorithms and complex networks. Huge amounts of data are generated by using bio-inspired algorithms. This data can be represented as a complex network, which is a natural alternative for representing and modeling the structure and non-linear dynamics of all discrete complex systems. We would like to use complex networks as a tool to describe and analyze behavior of bio-inspired algorithms, namely DE and SOMA. In our future research we are planning to use results from this analysis for both EA parameter tuning and control.

2 State of the Art

As mentioned above, EA have been widely used to solve problems from wide range of fields. However, parameters for such EA have to be modified in order to enhance the performance of EA for specific area of interest. Setting these parameters is a scientific discipline that is commonly divided into two approaches. First is parameter tuning and second is parameter control. Parameter tuning is easier task, since parameters are not changed during the EA run. When tuning parameters, scientists mostly rely on suggested values of parameters and well known conventions and often use approach of testing several different parameters settings [4, 11]. On the other hand, parameter control is a process of changing parameters during EA run, which requires well designed parameters control strategies [7, 8]. More information about parameters tuning and control could be found in [5, 6, 9].

3 Complex Networks

From a mathematical point of view, we can represent complex network using a graph. This graph is an ordered pair $G = (V, E)$ composed of a set of vertices and/or edges, where V represents set of vertices $V = \{1, \ldots, n\}$ and E represents set of edges $E = \{ e_1, \ldots, e_m \}$. Pair of vertices is linked together using edges. These vertices are called adjacent (neighbors). In complex networks, vertices represent objects of a real world and edges represent relations between them.

Our complex networks have two important properties. They are directed graphs and multigraphs. In a directed graph, the edges are taken as ordered pairs, i.e., each edge is directed from the first to the second vertex of the pair. A multigraph is a graph in which more than one edge is allowed between a pair of vertices and edges are also allowed to connect a vertex to itself.

In the analysis of our complex networks we are using many properties that describe the graph. These are degree distribution, graph diameter, graph density, graph modularity, number of connected components, clustering coefficient and average path length. More information about these properties can be found in [23].

4 Evolutionary Algorithms

A typical feature of evolutionary algorithms is that they are based on working with populations of individuals. We can represent the population as a matrix M × N where columns represent individuals. Each individual represents a solution to the current problem. In other words, it is set of cost function argument values. Combining these individuals, we are able to obtain optimal solution.

The main activity of evolutionary algorithms is the cyclic creation of new populations which are better than the previous ones. Better population is a population whose individuals have better fitness. If we are looking for global minimum, like in our case, we can schematically describe it with following formula:

$$\forall\, indA, indB \in \mathbb{R}, f_{cost}(indA) < f_{cost}(indB) \Rightarrow F(indA) > F(indB)\,, \tag{1}$$

where $indA, indB$ are population members, f_{cost} is cost function and F is function computing fitness (it is transformed cost function in our case). The goal is to find a vector $X = (x_1, x_2, x_3, \dots, x_n)$ which minimizes $f_{cost}(X)$ function with regard to function and arguments restrictions.

To find a global minimum of f_{cost} we use two representatives of evolutionary algorithms. These are Differential Evolution (DE) and Self-Organizing Migration Algorithm (SOMA).

For further information about evolutionary algorithms and their usage please refer to [18, 19].

5 Visualization of DE and SOMA

Methods proposed in [24] have been used for visualization of evolutionary process of DE and SOMA. We will describe them briefly here. We try to describe running of both algorithms by an oriented multigraph that exhibits properties of complex network. In both algorithms each member of the population is considered as a vertex of an evolutionary graph. In case of DE, we add edges oriented from vertices representing parents to vertex of their child every time the child is accepted as better solution than original specimen. This would produce three edges every time a child is accepted. In case of SOMA we add edges oriented from all other members of population to the leader every time the leader is chosen. This would produce *n-1* edges every time the leader is chosen providing we have *n* members of population. After enough generations or migrations are done, graph produced by this approach should have properties of complex network.

6 Experiment Description and Results

Our experiment composes of two parts. The first part is generation of data by application of DE and SOMA on a problem. The second part is visualization and analysis of graph produced by evolutionary processes. This approach has been applied to our previous research.

We have tried to improve Force-directed graph drawing algorithm [1, 2] by application of DE and SOMA for finding the minimal energy of the graph instead of the classical approach. Mathematically we have optimized following function:

$$\sum_{i=1}^{i\le|V(G)|}\left(\sum_{j=1 i\neq j A(i,j)=0}^{j\le|V(G)|}\frac{1}{\sqrt{\sum_{k=1}^{k\le 2}\left(x_{ik}-x_{jk}\right)^2}}+\sum_{j=1 A(i,j)\neq 0}^{j\le|V(G)|} ln\left(\sqrt{\sum_{k=1}^{k\le 2}\left(x_{ik}-x_{jk}\right)^2}\right)\right) \tag{6.1}$$

where *A* is adjacency matrix of the graph, *V(G)* is set of all vertices of the graph and *x* is vector with Cartesian coordinates of all vertices.

We have run both algorithms on visualization of a random graph with 50 vertices (not evolutionary graph) with parameters set as described in Tables 1 and 2.

Table 1 Parameter settings for differential evolution

Number of generations	300
Population size	1000
Crossover threshold	0.95
Mutation constant	0.8

Table 2 Parameter settings for Soma

Number of migrations	300
Population size	100
Path length	3
Step	0.2
Perturbation	0.5

Both algorithms were implemented in Matlab programming environment and tested on a computer with Intel Core i5 2540M with frequency of 2.6 GHz and with 8 GB of RAM memory. Resulting complex network has been visualized and analyzed in both Gephi and Matlab software in order to compare analysis and get more precise results.

Results from DE algorithm are shown in Table 3 and Figures 1, 3, 4 and 5. Table 3 shows various parameters of the evolutionary network. Diameter of the network, which is 5, is notable. It means that the resultant network follows small world hypothesis [23]. Figure 1 shows the evolutionary network itself. Figures 3, 4 and 5 show degree distribution during the evolution. It clearly shows that evolution converges towards Gaussian distribution. It implies that the evolutionary network created by DE algorithm is not Scale-free network [23].

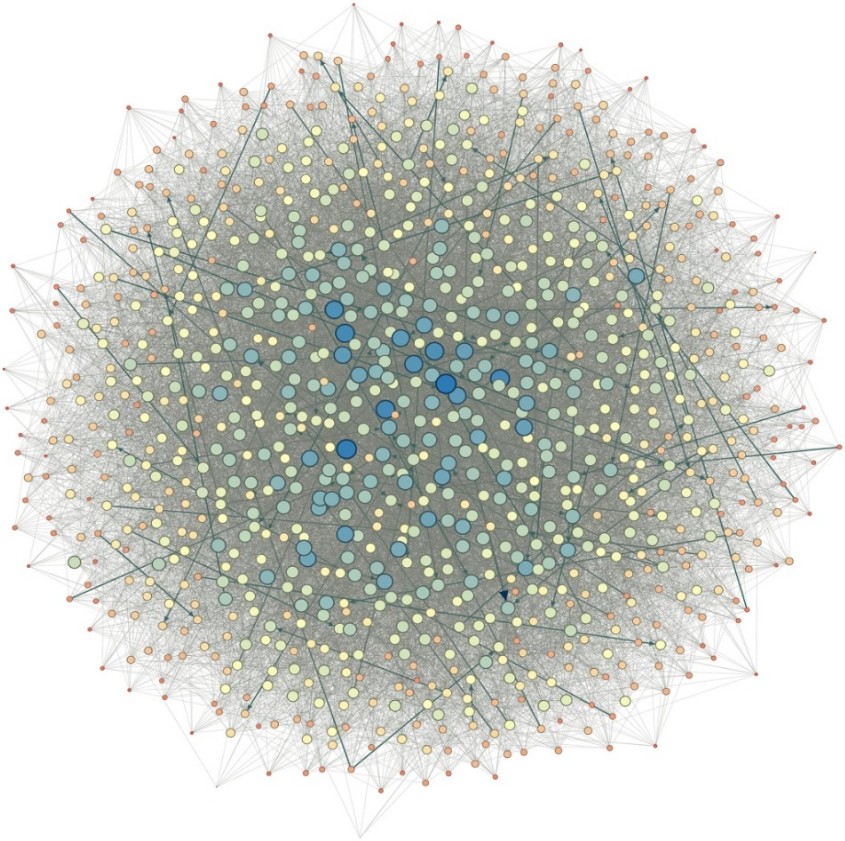


Fig. 1 DE, evolutionary network

Table 3 Properties of DE evolutionary network

Statistic	Value
Average Degree	14,792
Average Weighted Degree	14,889
Network Diameter	5
Average Path length	2,873
Number of shortest paths	980019
Graph Density	0,015
Modularity	0,161
Average Clustering Coefficient	0,016

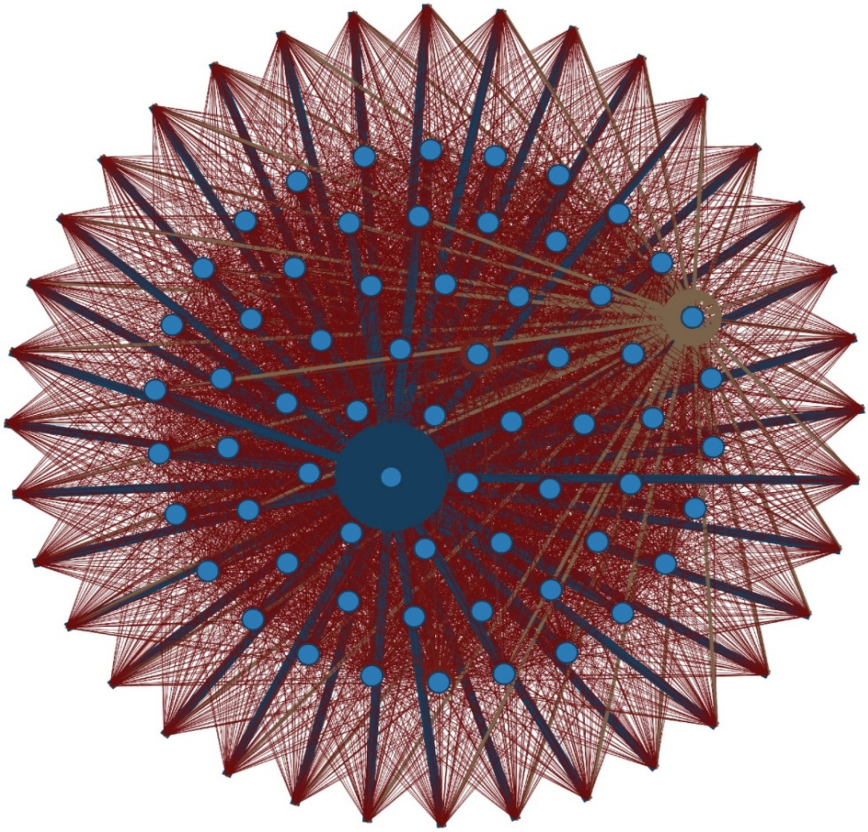


Fig. 2 SOMA evolutionary network

Table 4 Properties of SOMA evolutionary network

Statistic	Value
Average Degree	63,36
Average Weighted Degree	297
Network Diameter	1
Average Path length	1
Number of shortest paths	6336
Graph Density	0,640
Modularity	0,169
Average Clustering Coefficient	0,767

Table 4 and Figures 2, 6, 7 and 8 represent results of SOMA algorithm. Table 4 shows parameters of resultant network. Diameter is very small, therefore it is Small-world network. Also, relatively high clustering coefficient means that it is closer to real-world complex networks than in case of DE. Figure 2 shows the evolutionary network which is much more symmetric than in case of DE. Figures 6, 7

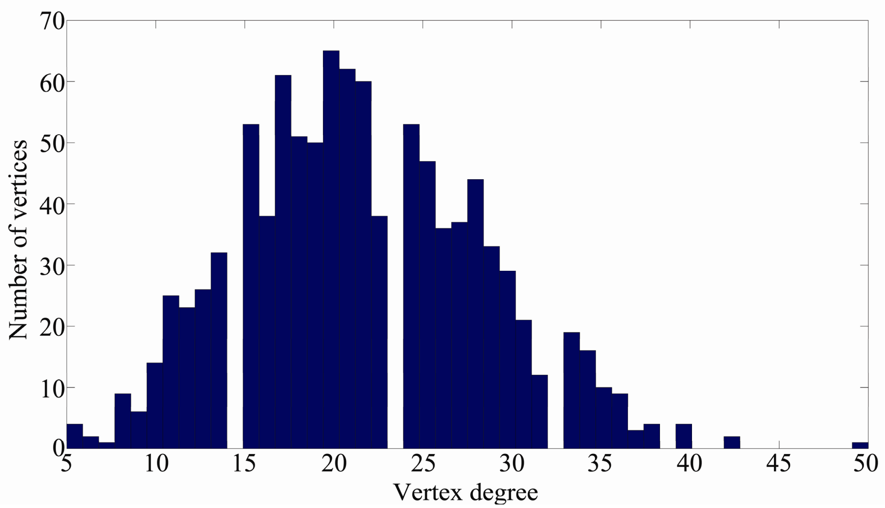


Fig. 3 Degree distribution of DE evolutionary network after 100 migrations

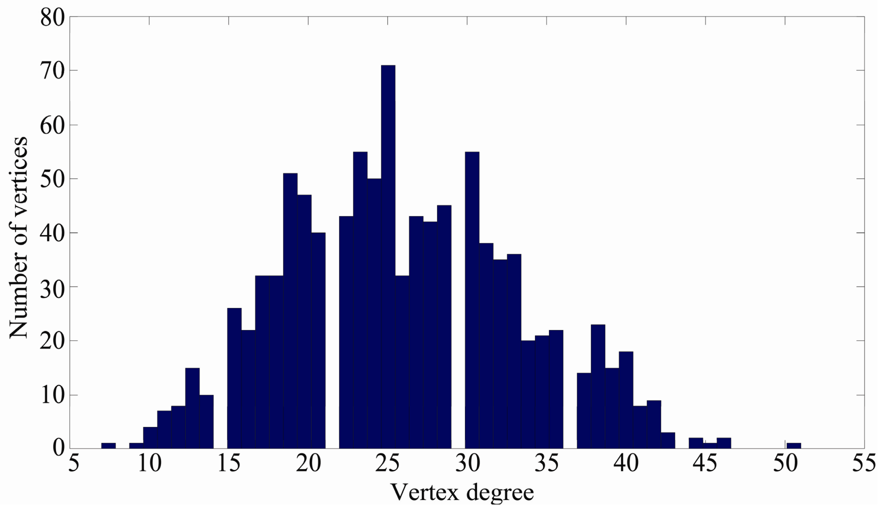


Fig. 4 Degree distribution of DE evolutionary network after 200 migrations

and 8 feature degree distribution during the process of SOMA algorithm. Its degree distribution follows power law with its heavy tail so it belongs to the scale-free network class. This property also does not change during the evolutionary process.

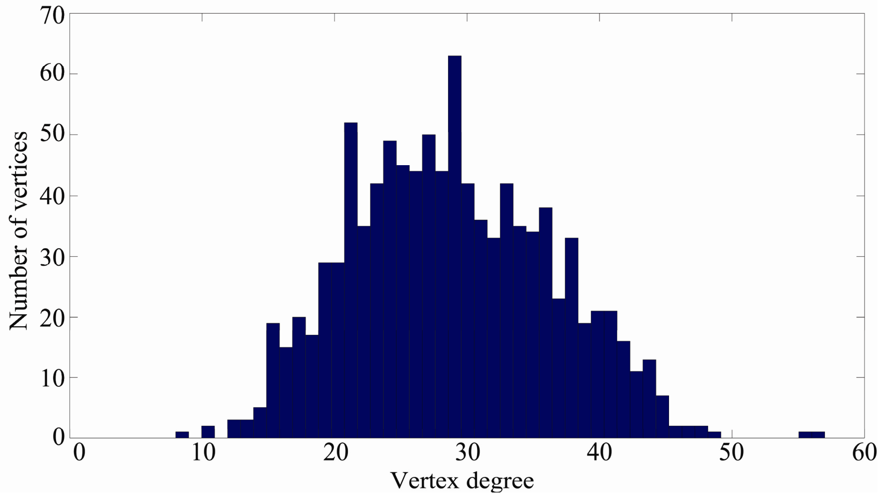


Fig. 5 Degree distribution of DE evolutionary network after 300 migrations

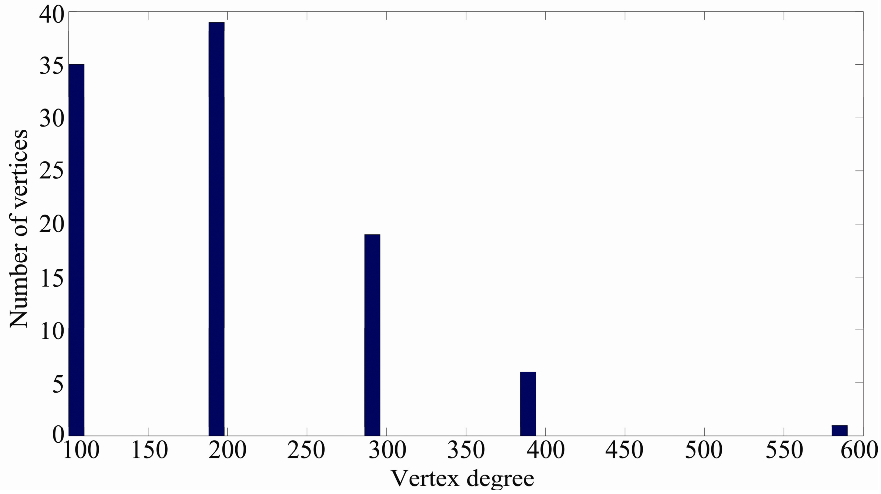


Fig. 6 Degree distribution of SOMA evolutionary network after 100 migrations

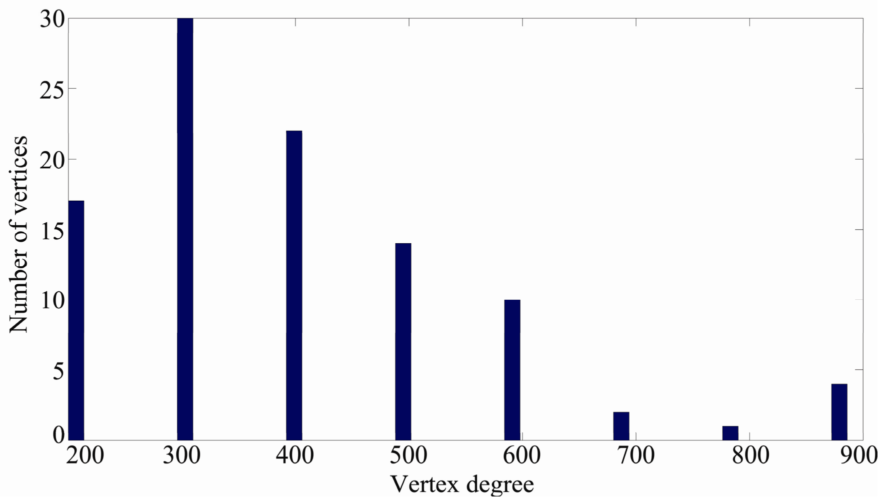


Fig. 7 Degree distribution of SOMA evolutionary network after 200 migrations

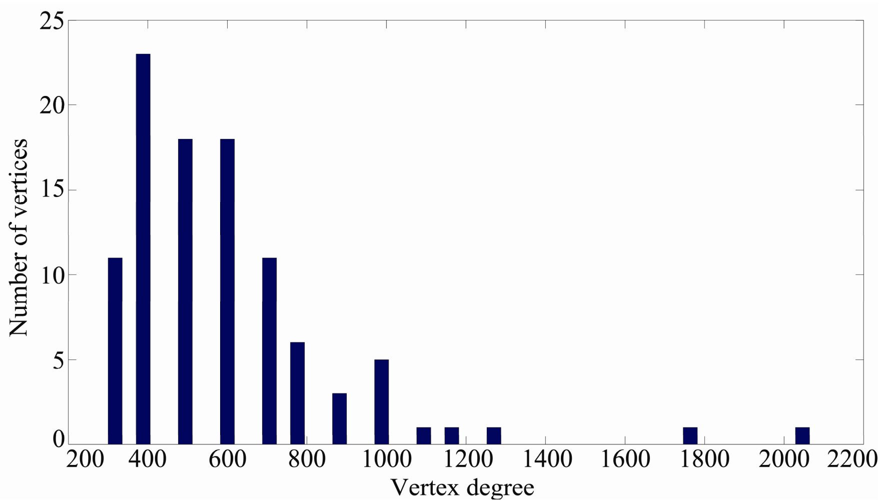


Fig. 8 Degree distribution of SOMA evolutionary network after 300 migrations

7 Conclusion

In this article we have presented a case study about the use of complex networks for visualization of evolutionary process. This approach to evolutionary visualization was tested on our previous research problem of Force-directed graph drawing algorithm. We have optimized energetic function contained in Force-directed algorithm by DE and SOMA algorithms and created an evolutionary complex networks. We have described properties of these networks and showed that they exhibit some typical properties of complex networks, like Small-world property and in case of SOMA high clustering coeficient. Several phenomena are to be studied further such as why DE converges towards Gaussian degree distribution and SOMA towards degree distribution following power law. Aim of this article was to present feasibility of this approach to a real problem and that properties of such network can be described and further analyzed. In our future research, we are planning to use this analysis to understand evolutionary process better and to enhance it by means of external control based on these insights and transition of complex network to CML system (Coupled map lattices, see [25]).

Acknowledgements. The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 218086.

This work was supported by the European Regional Development Fund in the IT4 Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070).

References

1. Azad, M. A.K., Fernandes, M.G.P.: Modified constrained differential evolution for solving nonlinear global optimization problems. In: Madani, K., Correia, A.D., Rosa, A., Filipe, J. (eds.) Computational Intelligence. SCI, vol. 465, pp. 85–100. Springer, Heidelberg (2013)
2. Anand, P., Venkateswarlu, C., BhagvanthRao, M.: Multistage dynamic optimiza-tion of a copolymerization reactor using differential evolution. Asia-Pacific Journal of Chemical Engineering 14(1) (2013)
3. Sivasubramani, S., Swarup, K.S.: Multiagent–based differential evolution algorithm for loss minimisation in power systems. International Journal of Power and Energy Conversion, 34–46 (2012)
4. Lobo, F., Lima, C., Michalewicz, Z. (eds.): Parameter Setting in Evolutionary Algorithms. SCI, vol. 54. Springer (2007)
5. Nannen, V., Eiben, A.: Relevance estimation and value calibration of evolutionary algorithm parameters. In: Proceedings of the Joint International Conference for Artificial Intelligence (IJCAI), pp. 975–980 (2006)
6. Hutter, F., Hoos, H.H., Stützle, T.: Automatic algorithm configuration based on local search. In: Proceedings of the 22nd Conference on Artifical Intelligence, pp. 1152–1157 (2007)
7. Eiben, A.E., Marchiori, E., Valkó, V.A.: Evolutionary algorithms with on-the-fly population size adjustment. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 41–50. Springer, Heidelberg (2004)
8. Nannen, V., Eiben, A.E.: Relevance Estimation and Value Calibration of Evolutionary Algorithm Parameters. In: Veloso, M.M. (ed.) IJCAI, pp. 1034–1039 (2007)
9. Wang, Y.: Differential Evolution with Composite Trial Vector Generation Strategies and Control Parameters. IEEE Transactions on Evolutionary Algorithms, 55–66 (February 2011)

10. Eiben, A.E.: Parameter tuning for configuring and analyzing evolutionary algorithms. Swarm and Evolutionary Computation 1, 19–31 (2011)
11. Eades, P.: A heuristic for graph drawing. Congressus Numerantium 42, 149–160 (1984)
12. Fruchterman, T., Reingold, E.: Graph drawing by force-directed placement. Softw. – Pract. Exp. 21(11), 1129–1164 (1991)
13. Kamada, T., Kawai, S.: An algorithm for drawing general undirected graphs. Inform. Process. Lett. 31, 7–15 (1989)
14. Gajer, P., Goodrich, M.T., Kobourov, S.G.: A multi-dimensional ap-proach to force-directed layouts of large graphs. Computational Geometry 29, 3–18 (2004)
15. Crawford, C.: A Multilevel Force-directed Graph Drawing Algorithm Using Multilevel Global Force Approximation. In: 2012 16th International Conference on Information Visualisation, IV (2012)
16. Finkel, B., Tamassia, R.: Curvilinear graph drawing using the force-directed method. In: Pach, J. (ed.) GD 2004. LNCS, vol. 3383, pp. 448–453. Springer, Heidelberg (2005)
17. Chernobelskiy, R., Cunningham, K.I., Goodrich, M.T., Kobourov, S.G., Trott, L.: Force-Directed Lombardi-Style Graph Drawing. In: Speckmann, B. (ed.) GD 2011. LNCS, vol. 7034, pp. 320–331. Springer, Heidelberg (2011)
18. Zelinka, I.: SOMA – Self Organizing Migrating Algorithm. In: Babu, B.V., Onwubolu, G. (eds.) New Optimization Techniques in Engineering, pp. 167–218. Springer, New York (2004) ISBN 3-540-20167X
19. Price, K., Storn, R.: Differential Evolution – A simple evolutionary strategy for fast optimization. In: Dr. Dobb's Journal, 264th edn., pp. 18–24, 78 (1997)
20. Price, K.: Differential evolution: a fast and simple numerical optimizer. In: Proc. 1996 Biennial Conference of the North American Fuzzy Information Processing Society, pp. 524–527. IEEE Press, New York (1996)
21. Price, K.: Genetic Annealing. Dr. Dobb's Journal, 127–132 (October 1994)
22. Price, K.: An Introduction to Differential Evolution. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, pp. 79–108. McGraw-Hill, London (1999)
23. Newman, M.E.J.: The Structure and Function of Complex networks. SIAM Rev. 45(2), 67–256 (2006)
24. Zelinka, I., Davendra, D.D., Chadli, M., Senkerik, R., Dao, T.T., Skanderova, L.: Evolutionary dynamics as the structure of complex networks. In: Zelinka, I., Snasel, V., Abraham, A. (eds.) Handbook of Optimization. ISRL, vol. 38, pp. 215–243. Springer, Heidelberg (2013)
25. Schuster, H.: Handbook of Chaos Control. Wiley-VCH (1999)

Stabilization of Chaotic Logistic Equation Using HC12 and Grammatical Evolution

Radomil Matousek and Petr Minar

Abstract. The paper deals with stabilization of simple deterministic discrete chaotic system. By means of proper utilization of meta-heuristic optimization tool, the HC12 algorithm stands alone and together with a symbolic regression tool, which is Grammatical Evolution (GE), and can synthesise a new control law. Given soft-computing tools appear as powerful optimization tool for an optimal control parameters tuning and general control law design too. The well known one dimensional discrete Logistic equation was used as a model of deterministic chaotic system. Satisfactory results obtained by both heuristics and propose objective function are also compared with previous research of other authors.

The chaotic system stabilization is based on time-delay auto-synchronization (TDAS, ETDAS) and proper combination with own synthesized control law. This synthesized chaotic controller is based on one or two compensator. The primary compensator generates the perturbation sequence using TDAS/ETDAS method, second one is own design using method of GE. The original design of the objective function takes inspiration from standard control theory. All tests are performed using Matlab/Simulink environment.

Keywords: Logistic equation, Deterministic chaos, Delayed feedback control, TDAS, ETDAS, Metaheuristic optimization, HC12, Grammatical evolution.

1 Introduction

Chaos theory currently penetrates to natural and engineering sciences. This theory during the last 20 years from its beginning has changed most of the things we have perceived as definitive, right, and normal, under influence of our traditions, customs, and history. According to chaos theory the systems which we know have no

Radomil Matousek · Petr Minar
Brno University of Technology, Technicka 2, 61669 Brno, Czech Republic
e-mail: {matousek,matousek}@fme.vutbr.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 137–146.
DOI: 10.1007/978-3-319-00542-3_14

energy transfers with environment (so called isolated systems) tend to evolve towards chaos. Therefore, the entropy increases. Practical common systems do exchange energy, matter or information with their environment (so called open systems). These systems evolve from simple to complex, from structures less organized to structures more organized, i.e. the so called auto-organization emerges. Certain level of complexity of given system is a condition of certain inner irreversibility of processes inside the system which invokes inconsistency and instability of system. It is a paradox this instability can be a source of new organization and new order. For this situation the term *deterministic chaos* has been defined . Deterministic chaos is typical for open non-linear dynamic macro systems for which due to initial condition or some important system's parameters (control parameters) come to unpredictability of its behaviour. Generally, if the system is markedly diverting from its steady state it will reach a boundary of its stability. This is characterized by so called *bifurcation point*. Primary division is primary bifurcation, it is generally characterized by the period of boundary cycle. Further from the steady state the number of oscillation frequencies increases and the cascades of bifurcations occur. System is getting to chaos state.

An example of a simple chaotic system is so called *Logistic map*. The map has been popularized in a seminal 1976 paper by the biologist Robert May [12], in part as a discrete-time demographic model analogous to the continuous logistic equation first created by Pierre Franois Verhulst (1838). Mathematically, the discrete logistic map is given by the recursive equation

$$x_{k+1} = rx_k(1 - x_k), \tag{1}$$

where $x_k \in [0,1]$, real meaning of k index is not important in our case, we take it as a number of iterations $k \in \mathbf{N}$, and r represents control parameter of the discrete logistic equation. As one can see in bifurcation diagram Fig. 1, the r parameter determines the behaviour of the system.

The model description of the complex non-linear system is the first step. Second and no less important questions are the system stability and the process control

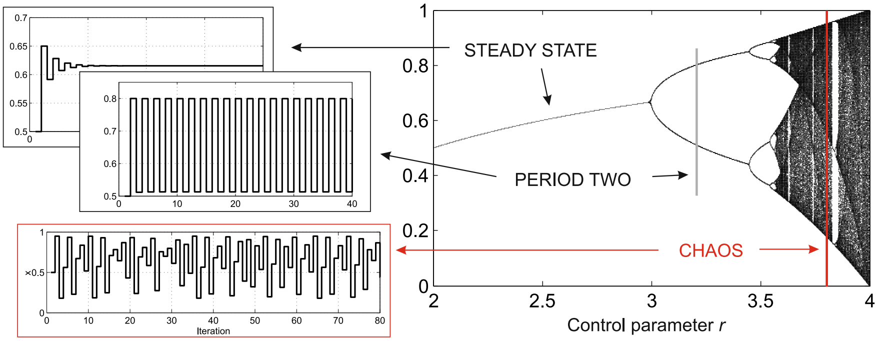


Fig. 1 Bifurcation of the logistic function over changes in maximum rate of change

which is thr main part of the paper. Question of stabilization or control design is more interesting for non-linear and chaotic systems. The control of chaos means the stabilization using small system perturbations of unstable periodic orbits. The perturbation has to be tiny to avoid significant modification of the system's natural dynamics.

Several approaches have been devised for chaos control but most are based on two basic approaches: the OGY method (Ott, Grebogi and Yorke), and Pyragas delayed "feedback" control. Both methods require previous determination of the unstable periodic orbits of the chaotic system before the controlling algorithm can be designed. In the OGY method [13, 14], small, wisely chosen, swift kicks are applied to the system once per cycle, to maintain it near the desired unstable periodic orbit. In the Pyragas method [16] an appropriate continuous controlling signal is injected into the system with practically zero intensity as the system evolves close to the desired periodic orbit but increases when it drifts away from the desired orbit. Late Pyragas' approaches [17] used delayed feedback control technique. In this paper, Pyragas' delayed feedback control technique ETDAS (Extended Time Delay Auto Synchronization) and the simple version for one orbit state TDAS will be used.

It is need to remark there are really many researcher's approaches in chaos control area. For example, numerous methods based on adaptive control theory [18], centre manifold targeting [21], reducing the targeting problem to the problem of finding optimal paths through a graph [1], and also modern soft computing methods as neural networks control [5] or evolutionary approaches are used. Proper overview of chaos control papers can be found in [2] and [3].

As we have denoted above, in the paper Pyragas method TDAS and ETDAS will be used. The big advantage of this methods is good definition of control parameters in case of global optimization techniques. Considering this fact the metaheuristic optimisation techniques is a proper choice. Very promising results was obtained using SOMA (Self-Organizing Migrating Algorithm) [26], DE (Differential Evolution)[7] and AP (Analitic Programming) [26] methods.

In Zelinka et al. [26] the SOMA optimization algorithms and TDAS/ETDAS principle were used for orbit stabilisation in several cases of discrete chaotic systems. In Senkerik et al. [23, 28] two evolutionary algorithms SOMA and DE for optimization of TDAS/ETDAS parameters are used. This paper also presents proper objective functions which are utilized up to $p-4$ orbits stabilisation [27]. In the papers [24, 25, 6] one can see new level of chaos control synthesis based on design of complete control low using AP.

2 Optimization Algorithms

In this section we briefly describe optimization methods used for design of optimal parametrization of control model of given chaotic system. As we have denoted, the principle of control will be based on perturbations generated by TDAS model in case of stabilization to $p-1$ orbits and ETDAS model in case of stabilization to higher order orbits.

Let us note the following. In both cases a discrete variant of given methods which are originally designed as continuous and were introduced by Pyragas [16] is used. Considering the terminology of automation control it is a compensator in essence.

The given method of stabilization is established as time-delayed feedback control or time-delay autosynchronization. In practical point of view, the method constructs a control force from the difference of the present state of a given system to its delayed value, i.e., $s(t)-s(t-\tau)$ or $s(k)-s(k-1)$ in our discrete version. Also from a mathematical perspective the solution corresponds to the class of delay difference equations.

The Logistic equation Eq. 1 is stabilised by optimal tuning of compensator F_n by TDAS model in case of $p-1$ orbit, see Eq. 2, and by ETDAS model in case of $p-2$ orbit, see Eq. 3. Orbits higher than $p-2$ are not discussed in this paper.

$$\begin{aligned} x_{k+1} &= rx_k(1-x_k)+F_k \\ F_k &= K(x_{k-1}-x_k) \end{aligned} \tag{2}$$

$$\begin{aligned} x_{k+1} &= rx_k(1-x_k)+F_k \\ F_k &= K\left[(1-R)S_{k-2}-x_k\right] \\ S_k &= x_k+RS_{k-2} \end{aligned} \tag{3}$$

where K and R are adjustable parameters, F_k is the perturbation series, S_k is given by a delay equation utilizing previous system states and iteration index $k \in \mathbb{N}$. As one can see the variable x is defined in $0 < x \leq 1$. The perturbation action has possibility to cause diverging of system outside this interval. Due to this fact, the saturation function is used and the values of F_k are limited by parameter F_{max}, i.e. $F_k \in \left[-F_{max}, F_{max}\right]$. The limitation is the next adjustable parameter in case of optimal system stabilisation.

The goal of our optimization task is to stabilize the system (1) in the fixed point $p-1$ and also in $p-2$ periodic orbit. The models given by (2) or (3) were performed by the developed objective function

$$IKAE = k\sum_{k=1}^{kMax} |x_{TS}-x_k|, \tag{4}$$

where TS index denotes a *target state* (set point in automatic control terminology), x_k means an *actual state* given by model (2) or (3) and *kMax* means a maximum number of iteration steps of computer simulation. The objective function (4) includes penalisation of target state error (i.e. actual state value deviation from fixed point in our case) and penalisation in relation of the iteration step. Of course, there are many variants of (4) as in (5), where a sense of robustness by parameters r,s is evident.

$$IKAE^{rs} = k^r\sum_{k=1}^{kMax} \left(|x_{TS}-x_k|\right)^s \tag{5}$$

The Eq. (4) was used in case of one fixed point stabilisation. The *IKAE* criterion has to be split up `even` and `odd` part by orbit for two fixed points target state TS_1 and TS_2 (6). The same principle can be used in case of higher periodic orbit.

$$IKAE_2 = IKAE_2^{odd} + IKAE_2^{even}, \text{where} \begin{cases} IKAE_2^{odd} = k \sum\limits_{k=1,3,5,\dots}^{kMax} |x_{TS_1} - x_k| \\ IKAE_2^{even} = k \sum\limits_{k=2,4,6,\dots}^{kMax} |x_{TS_2} - x_k| \end{cases} \quad (6)$$

The minimization problems given by Eqs. (4) and (6) are solved. The logarithmic scale of *IKAE* criteria is used for a practical implementation. A logarithmic scale reduces wide-ranging quantities, i.e. $log_{10}(IKAE + \varepsilon)$ in our case, where due to theoretical possibility $IKAE = 0$ the ε is a constant close to zero.

This research utilized two evolutionary algorithms HC12 and GE. There are our original variants of meta-heuristic optimisation processes:

- HC12 optimization of F_k series (i.e. $K, F_{max},$ or R model parameter tuning).
- HC12 optimization of F_k series and next design of new M_k perturbation series by GE as parallel compensator.
- HC12 optimization of F_k series and next design of new M_k perturbation series by GE as parallel compensator and finally the re-tuning of F_k series.

2.1 HC12 Algorithm

The HC12 algorithm is a simple optimization meta-heuristic which was developed by Matousek (1996). This method has been implemented in many optimization fields [11]. The principle of the HC12 algorithm is the parallel manipulation with binary vectors and its advantage is the excellent scalability [10]. The algorithm HC12 has quadratic complexity. It generates neighbourhood using bit masks. Those masks have Hamming distances one and two from the origin binary string which represents a problem solution. The HC12 mathematical description can be found for example in [8, 9]. The presented results which are described in Sect. 3 use 12 bit parameter encoding, i.e. approximately 4 digits floating point precision for given domain of definition $K, F_{max},$ or R.

2.2 Grammatical Evolution

Grammatical evolution (GE) is one of the newest approaches in evolution algorithms. GE was designed by Conor Ryan, JJ Collins and Michael O'Neill in 1998 [15]. Practically, from optimization point of view, it has similar options as well known GP or less known AP. The GE can realized automatic generating programs or in our case a control law. In the GE the programs (or models) are generated in languages described by Backus-Naur form (BNF) of context-free grammars. The GE paradigm is a principle of coding/decoding of known GA [4] using BNF, i.e. a genotype is interpreted in the same way as in GA and decoded a phenotype according the rules of GE. A choice of terminals and non-terminals of grammar and rewriting rules understandingly changes options of result model construction. In additional, the resulting models have been generated in Matlab/Simulink environment. The `memory` block interpreted a delay of sequence's members. According the presented aim the following grammar has been chosen:

Listing 1 The grammar for automatic generating control low (M_k series)

```
N = <init> | <node> | <1input> | <2input> | <const>
T = memory | + | - | * | / | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
    x(n)
S = <init>

<init> = <node> <1input> | <node> <node> <2input>
<node> = <init> | x(n) | <const>
<1input> = menory, gain(-1)
<2input> = + | - | * | /
<const> = 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

2.3 Differential Evolution

Differential Evolution (DE) is a very known and simple population based evolution meta-heuristics suitable for searching of global extreme in non-linear and multidimensional space. Differential Evolution grew out of Ken Price's attempts to solve the Chebychev Polynomial Fitting Problem that had been posed to him by Rainer Storn during 1994-1996. There are many variants of DE algorithm [7], which differ mainly in computing weighted difference vector. The algorithm DE is still very popular also for comparison of performance of newer optimization meta heuristics like in our case. The advantage of DE is in simplicity of implementation and low demanding computations for generating new populations. The next big advantage of the algorithm is a capability to work with continuous parameters. The disadvantage of DE is strong dependence on initial settings of the method F [coefficient of mutation], CR [coefficient of crossover] and NP [population size].

3 Experimental Results and Conclusion

The following results present two basic version of our experiments listed above. The first version is the model (2), (3) parameter tuning, the second one is a design of own control sequences M_k which together with F_k sequences will stabilised given discrete chaotic system (1). An automatic model design for M_k sequence generating is based on GE using grammar by Listing 1.

The objective of the optimisation was to estimate the optimal parameter values to system stabilisation in target state(s):

- $p-1$ orbit of Eq. (1) for $r = 3.8$ has the fixed point $x_F = 0.7368$.
- $p-2$ orbit of Eq. (1) for $r = 3.8$ has the fixed points $x_{F_1} = 0.3737, x_{F_2} = 0.8894$.

The results for $p-1$ and $p-2$ orbits stabilisation are displayed in Table 1 and Table 2. The domain of definitions of all estimated parameters were the following: $K \in [-2,2]$, $F_{max} \in [0.01, 0.5]$, $R \in [0,1]$. The united perturbations principle was used in case of HC12+GE optimization, i.e. $x_{k+1} = rx_k(1-x_k) + F_k + M_k$. This procedure means the optimal F_k parameter tuning with subsequent design of M_k sequence. The HC12+GE+ was equal as previous one, but after GE design the F_k parameter tuning was made again. An example of M_k sequences generators are displayed in Table 3 for $p-1$ and $p-2$ orbits stabilisation. Simulation time, i.e. number of steps per simulation in our discrete version, was $kMax = 200$ steps.

Table 1 The best solutions for $p-1$ orbit (TDAS control, $IKAE$ objective function, $x_0 = 0.5$)

Optimization	Model	Perturbance	K	F_{max}	$IKAE$	CF_1
HC12	TDAS	F_k	-0.41642	0.213099	0.9975	4.62e-5
DE	TDAS	F_k	-0.43691	0.212527	1.0122	1.59e-8
HC12+GE	TDAS+BNF	F_k, M_k	-0.41642	0.213099	0.9229	4.44e-15
HC12+GE+	TDAS+BNF	F_k, M_k	-0.22483	0.484848	0.7515	1.11e-16

Table 2 The best solutions for $p-2$ orbits (ETDAS control, $IKAE_2$ objective function, $x_0 = 0.5$)

Optimization	Model	Perturbance	K	F_{max}	R	$IKAE$	CF_2
HC12	ETDAS	F_k	0.45943	0.42473	0.29716	0.9904	1.64e-5
DE	ETDAS	F_k	0.45225	0.20344	0.27652	0.8172	9.21e-5
HC12+GE	ETDAS+BNF	F_k, M_k	0.45943	0.42473	0.29716	0.7269	1.90e-5
HC12+GE+	ETDAS+BNF	F_k, M_k	0.51417	0.14369	0.34408	0.2393	1.23e-6

Table 3 An example of M_k sequence generators for $p-1$ orbit (HC12+GE optimization)

Orbit	Model	Equation	$IKAE$
p-1	[1]	$(x_{k-1} + x_k)x_{k-1} + 2x_k + x_{k-1}$	0.9229
	[2]	$(x_{k-1} + x_k)x_{k-1} + 4x_k - x_{k-1}$	0.9387
	[3]	$(x_{k-1} + x_k)x_{k-1} + (x_k + x_{k-1})$	0.9815
	[4]	$(x_{k-1} + x_k)x_{k-1} + x_k - 1/(6(x_k - x_{k-1})) + x_k + 4$	0.9857

The best presented results in the tables are based on one hundred independent restarts of HC12 algorithm in each instance. There are also next comparative objective function values based on different construction principle (CF_1 and CF_2) proposed in [28, 24, 6]. The following figures show the influence of given optimisation procedure listed in Table 1 and Table 2.

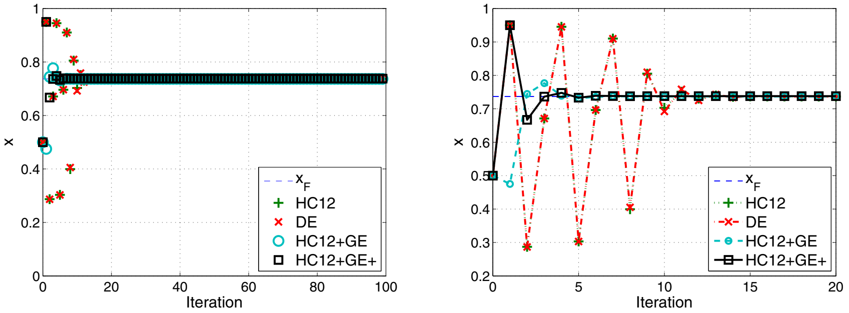


Fig. 2 Examples of simulation results - stabilisation of chaotic systems for $p-1$ orbit (one fixed point) by different control laws and different optimisation procedures

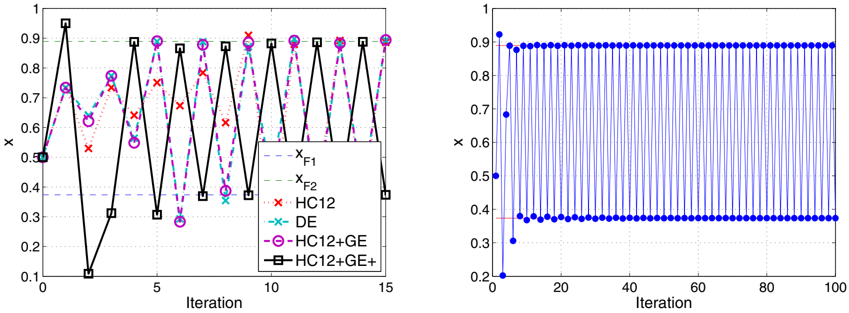


Fig. 3 Examples of simulation results - stabilisation of chaotic systems for $p-1$ orbit (one fixed point) by different control laws and different optimisation procedures

The research deals with a synthesis of the control laws by means of HC12 and GE for fixed point(s) stabilisation of Logistic equation. One can see that combining the stabilisation methods based on F_k and M_k sequences provides the best results in both examples. The basic results obtained by HC12 and DE are similar and the next comparison is also the question of the numeric precision. The presented results reinforce the argument that HC12 and GE are able to solve this kind of difficult non-linear optimization problem. Also simple design objective function $IKAE$ can be used and generalised in case of chaos control - stabilisation.

An important question and also future work, will be using HPC CUDA implementation of HC12 [10] where we expect significantly better results. A GE improvement of the result will be expected in case of newly designed BNF grammar base.

Acknowledgements. This work was supported by grants IGA No. FSI-S-11-31: Applied Artificial Intelligence and TACR No. TA02021449: Intelligent Alarm System for Nuclear Power Plant Technology. The authors also thank to friends in arm namely Roman, Ivan and Zuzka for convenient discussion about chaos control.

References

1. Bollt, E.J., Kostelich, E.J.: Optimal targeting of chaos. Physics Letters A 245, 399–496 (1998)
2. Chen, G.: Control and Synchronization of Chaotic Systems (bibliography). ECE Dept, Univ of Houston, `http://www.ee.cityu.edu.hk/~gchen/chaos-bio.html`(cited March 13, 2013)
3. Fradkov, A.L.: Chaos Control Bibliography (1997-2000). Russian Systems and Control Archive (RUSYCON), `http://www.rusycon.ru/chaos-control.html` (cited March 13, 2013)
4. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman Publishing Co.,Inc, Boston (1989)
5. Iplikci, S., Denizhan, Y.: An improved neural network based targeting method for chaotic dynamics. Chaos, Solutions & Fractals 17(2-3), 523–529 (2003)
6. Kominkova-Oplatkova, Z., Senkerik, R., Zelinka, I., Pluhacek, M.: Analytic programming in the task of evolutionary synthesis of a controller for high order oscillations stabilization of discrete chaotic systems. Computers & Mathematics with Applications (March 5, 2013)
7. Lampinen, J., Zelinka, I.: Mechanical engineering design optimization by differential evolution. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, pp. 127–146. McGraw-Hill (1999) 007-709506-5
8. Matousek, R.: GAHC: Improved GA with HC mutation. In: WCECS 2007, San Francisco, pp. 915–920 (2007) ISBN: 978-988-98671-6-4
9. Matousek, R.: GAHC: Hybrid Genetic Algorithm. In: Ao, S.-I., Rieger, B., Chen, S.-S. (eds.) Advances in Computational Algorithms and Data Analysis. LNEE, vol. 14, pp. 549–562. Springer, Heidelberg (2009)
10. Matousek, R.: HC12: The Principle of CUDA Implementation. In: 16th International Conference on Soft Computing, MENDEL 2010, Brno, pp. 303–308 (2010)
11. Matousek, R., Zampachova, E.: Promising GAHC and HC12 algorithms in global optimization tasks. Journal Optimization Methods & Software 26(3), 405–419 (2011)
12. May, R.M.: Simple mathematical models with very complicated dynamics. Nature 261(5560), 459–467 (1976), doi:10.1038/s261459a0
13. Ott, E., Grebogi, C., Yorke, J.A.: Controlling chaos. Phys. Rev. Lett. 64, 1196–1199 (1990)
14. Ott, E., Grebogi, C., Yorke, J.A.: Controlling chaotic dynamical systems. In: Campbell, D.K. (ed.) Chaos, Amer. Inst. of Phys., New York, pp. 153–172 (1990)

15. O'Neill, M., Ryan, C.: Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language. Kluwer Academic Publishers (2003)
16. Pyragas, K.: Continuous control of chaos by self-controlling feedback. Phys. Lett. A. 170, 421–428 (1992)
17. Pyragas, K.: Control of chaos via extended delay feedback. Phys. Lett. A 206, 323–330 (1995)
18. Ramaswamy, R., Sinha, R., Gupte, N.: Targeting chaos through adaptive control. Phys. Rev. Lett. 57(3), 2507–2510 (1998)
19. Richter, H., Reinschke, K.J.: Optimization of local control of chaos by an evolutionary algorithm. Physica D 144, 309–334 (2000)
20. Richter, H., Reinschke, K.J.: Local control of chaotic systems: a Lyapunov approach. Int. J. Bifurc. Chaos 8, 1565–1573 (1998)
21. Starrett, J.: Time-optimal chaos control by center manifold targeting. Phys. Rev. Lett. 66(4), 6206–6211 (2002)
22. Senkerik, R., Zelinka, I., Oplatkova, Z.: Optimal control of evolutionary synthesized chaotic system. In: Matousek, R. (ed.) 15th International Conference on Soft Computing, MENDEL 2009, pp. 220–227 (2009) ISSN: 1803-3814, ISBN: 978-80-214-3884-2
23. Senkerik, R., Zelinka, I., Davendra, D., Oplatkova, Z.: Utilization of SOMA and differential evolution for robust stabilization of chaotic Logistic equation. Computers & Mathematics with Applications 60(4), 1026–1037 (2010)
24. Senkerik, R., Oplatkova, Z., Zelinka, I., Davendra, D.: Evolutionary chaos controller synthesis for stabilizing chaotic Henon maps. Complex Systems, 0891-2513 20(3), 205–214 (2012)
25. Senkerik, R., Oplatkova, Z., Zelinka, I., Davendra, D.: Synthesis of feedback controller for three selected chaotic systems by means of evolutionary techniques: analytic programming. Mathematical and Computer Modelling 57(1-2), 57–67 (2013)
26. Zelinka, I.: SOMA–self organizing migrating algorithm. In: Babu, B.V., Onwubolu, G. (eds.) New Optimization Techniques in Engineering, vol. 33, ch. 7. Springer (2004)
27. Zelinka, I., Guanrong, C., Celikovsky, S.: Chaos synthesis by means of evolutionary algorithms. International Journal of Bifurcation and Chaos 18(4), 911–942 (2008)
28. Zelinka, I., Senkerik, R., Navratil, E.: Investigation on evolutionary optimitazion of chaos control. Chaos, Solitons & Fractals 40, 111–129 (2009)

Hypervolume-Driven Analytical Programming for Solar-Powered Irrigation System Optimization

T. Ganesan, I. Elamvazuthi, Ku Zilati Ku Shaari, and P. Vasant

Abstract. In the field of alternative energy and sustainability, optimization type problems are regularly encountered. In this paper, the Hypervolume-driven Analytical Programming (Hyp-AP) approaches were developed. This method was then applied to the multiobjective (MO) design optimization of a real-world photovoltaic (PV)-based solar powered irrigation system. This problem was multivariate, nonlinear and multiobjective. The Hyp-AP method was used to construct the approximate Pareto frontier as well as to identify the best solution option. Some comparative analysis was performed on the proposed method and the approach used in previous work.

Keywords: Solar power, photovoltaic (PV), irrigation system, multiobjective (MO) Optimization, analytical programming (AP), hypervolume indicator (HVI).

1 Introduction

Currently, optimization problems are frequently encountered by engineers and scientists working in systems involving alternative energy and sustainable technologies (Elamvazuthi *et al.* 2011; Ganesan *et al.* 2012). Standard irrigation systems are usually powered by diesel generators or other fossil-fuel based power sources. Hence, the idea of the utilization of solar energy to supply power to

T. Ganesan · Ku Zilati Ku Shaari
Department of Chemical Engineering
University Technology Petronas, 31750 Tronoh, Perak, Malaysia

I. Elamvazuthi
Department of Electrical & Electronic Engineering
University Technology Petronas, 31750 Tronoh, Perak, Malaysia

P. Vasant
Department of Fundamental & Applied Sciences,
University Technology Petronas, 31750 Tronoh, Perak, Malaysia
e-mail: tim.ganesan@gmail.com

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 147–154.
DOI: 10.1007/978-3-319-00542-3_15

irrigation pumps is economically and ecologically more attractive (Helikson *et al.*, 1991; Wong & Sumathy, 2001). The design and sizing of solar powered irrigation system enters into the optimization domain. This is due to the fact that optimal design and sizing should be carried out such that issues like; cost saving, emissions, system efficiency as well power output are ensured or improved.

In multi-objective optimization, one approach that has been effective in measuring the quality of the solution set that constructs the Pareto-frontier (in cases where the Pareto frontier is unknown) is the Hypervolume Indicator (HVI) (Zitzler & Thiele, 1998). The HVI is the only indicator which is strictly Pareto-compliant that can be used to measure the quality of solution sets (degree of dominance) in MO optimization problems (Beume *et al.*, 2007; Zitzler & Thiele, 1999).

The main approach presented in this work involves an evolutionary symbolic regression approach called Analytical Programming (AP) (Zelinka, 2002; Zelinka & Oplatkova, 2003). This work aims to solve and obtain a set of Pareto-efficient solution options for the MO optimization of the solar powered irrigation system. The solar powered irrigation system problem was developed and rigorously validated in Chen *et al*, (1995). The solution method proposed in this work is the hypervolume-driven symbolic regression approach (Hyp-AP). This method uses the HVI to drive the AP approach in search for dominant solutions in the objective space.

This paper is organized as follows. In section 2 of this paper, the problem formulation is presented. In section 3 the computational technique is discussed and this is followed by Section 4 which discusses the computational results. Finally, this paper ends with some concluding remarks and recommendations for future works.

2 Problem Formulation

In the solar powered irrigation system, the design parameters of the irrigation system influences the system's performance, efficiency and cost. In Chen *et al.* 1995 [a], objectives were set as the pump load/power output, f_1 (kW), overall efficiency, f_2 (%) and the fiscal savings, f_3 (USD). The design variables were; the maximum pressure, x_a (MPa), maximum temperature, x_b (K), maximum solar collector temperature, x_c (K), the fluid flowrate, x_d (kg/s), ambient temperature, Z_a (K) and the level of insolation, Z_b (K). The objective functions and the design constraints are shown as follows:

$$\begin{aligned} f_1 = -(&24.947 + 16.011x_d + 1.306x_b + 0.820x_b x_d - 0.785Z_a \\ &- 0.497x_d Z_a + 0.228x_a x_b + 0.212x_a - 0.15x_b^2 + 0.13x_a x_d \\ &- 0.11x_a^2 - 0.034x_b Z_a + 0.002x_a Z_a)10^{-3.24} \end{aligned} \tag{1}$$

$$\begin{aligned} f_2 = -&43.4783(0.18507 + 0.01041x_a + 0.0038Z_b - 0.00366Z_a \\ &- 0.0035x_c - 0.00157x_b) \end{aligned} \tag{2}$$

$$\begin{aligned} f_3 = &-(174695.73 + 112114.69x_d + 9133.8x_b + 5733.05x_b x_d \\ &- 5487.76Z_a - 3478.84x_d Z_a + 1586.48x_a x_b + 1486.84x_a \\ &- 1067.42x_b{}^2 + 916.26x_a x_d - 768.9x_a^2 - 242.88x_b Z_a \\ &+ 152.4x_a Z_a)10^{-3.23} \end{aligned} \tag{3}$$

$$0.3 \le x_a \le 3 \ ;450 \le x_b \le 520 \ ;520 \le x_c \le 800 \ ;0.01 \le x_d \le 0.2 \ ;293 \le Z_a \le 303 \ ;800 \le Z_b \le 1000 \tag{4}$$

The Hyp-AP algorithm used in this work was programmed using the C++ programming language on a personal computer (PC) with an Intel dual core processor running at 2 GHz.

3 Proposed Technique

The AP algorithm is a form of generalized approach aimed towards the development of a universal method for symbolic regression. There are two known computing methods capable of symbolic regression and hence synthesizing symbolic structures The first is known as genetic programming (GP) (Koza, 1990) and the second is grammatical evolution (GE) approach (Ryan *et al.*, 1998). The AP approach was first introduced in Zelinka & Oplatkova, 2003. The core concept of this technique involves the generation of symbolic structures by evolutionary means (Zelinka *et al.*, 2005). Each instruction array organizes the instruction sequence in the symbolic structure. On the other hand, each symbolic structure represents a computational expression (see Fig. 1).

In this work, the hypervolume indicator (HVI) was embedded into the AP algorithm. The HVI measures the quality (degree of dominance) of the solutions in any given MO setting. Therefore, the Hyp-AP utilizes the HVI to direct the algorithm towards higher dominance at successive iterations. To drive the algorithm effectively towards dominance, the HVI was coupled with the mutation operator. This mechanism works such that, if at iteration, *i* the degree of dominance is lower than the previous iteration, *i-1*, then at iteration *i+1*, the mutation operator would increase its degree of mutation so that the successive population would have a more diverse solution pool. However, if the degree of dominance at *i* is higher than the previous iteration, *i-1*, then at iteration *i+1*, the mutation operator would decrease its degree of mutation so that algorithm may converge into the most dominant solution. The original AP was modified by introducing an evolutionary mutation operator for this application. This mutation operator acts as a population diversifier

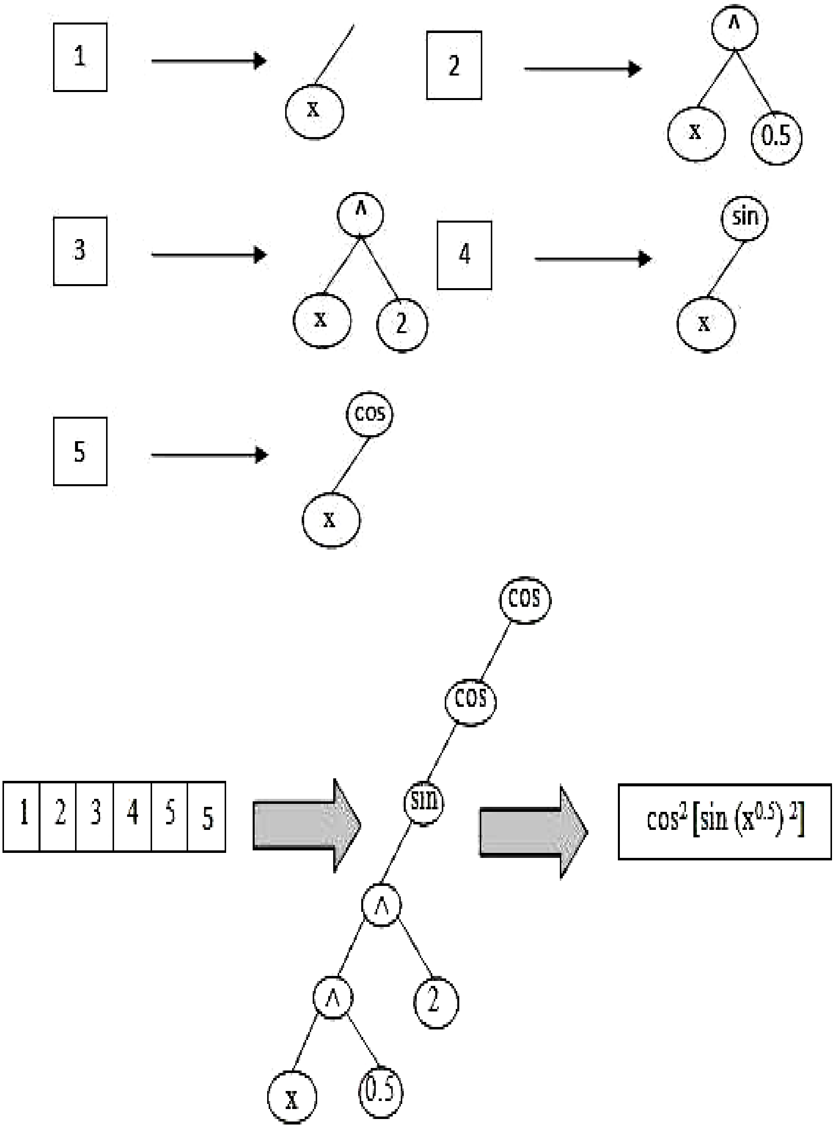


Fig. 1 Symbolic structure representing a computational expression

during the iterations. Thus, the AP is driven by the HVI as it searches for highly dominant solutions in a self-organized way. Fig. 2 depict the workflow of the Hyp-AP approach:

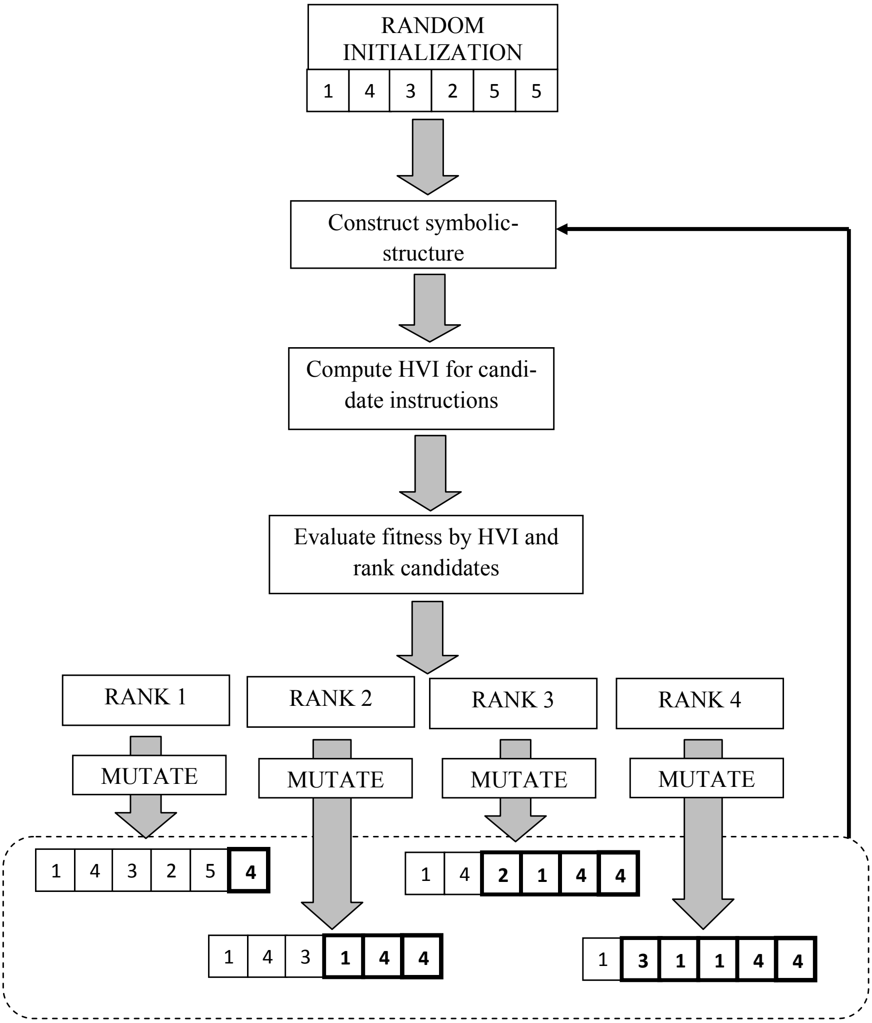


Fig. 2 Hyp-AP workflow

The parameters in the Hyp-AP done in this work were initialized where; the initial population (*P*) was 4, the individual size (*N*) was 6 bits, the HVI reference point (*H*) was (0, 0, 141253.8) and the bit-flip type mutation operator was used.

4 Numerical Results

The approximate Pareto frontiers obtained using the Hyp-AP is shown in Fig. 3:

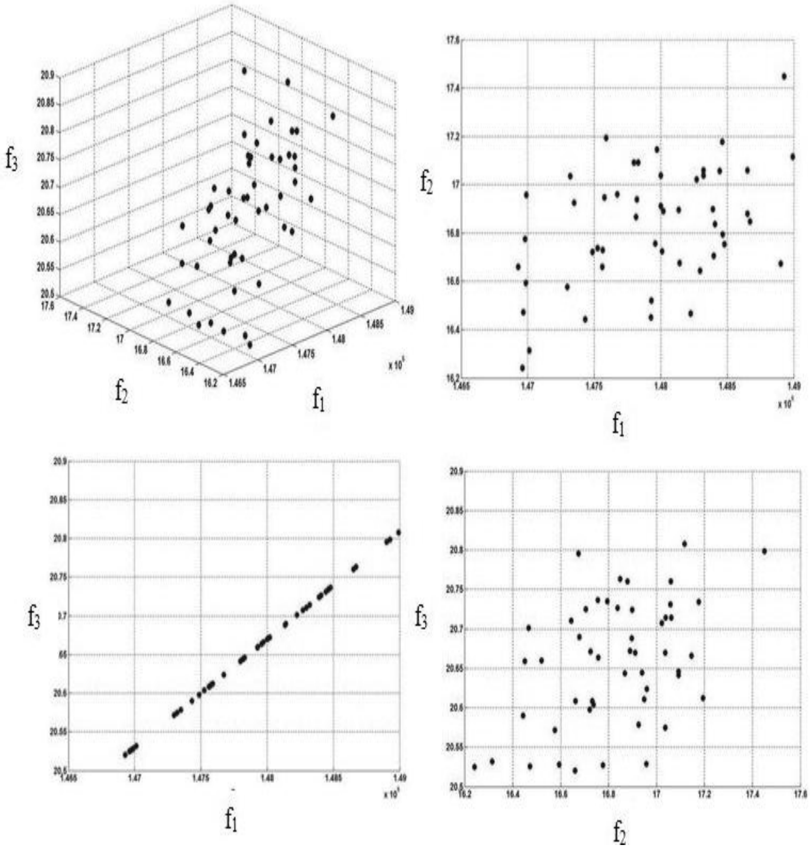


Fig. 3 Approximate Pareto frontiers obtained using the Hyp-AP algorithm

In Fig. 3 it can be observed that the f_1 versus f_3 scatter plot forms an almost straight line although there are some variation in the gradient and the solution distribution on the line. Hence, the solutions in three-dimensional objective space produced by the Hyp-AP are also distributed on a plane. In Chen *et al.* (1995), the Decision Support in the Design of Engineering Systems (DSIDES) software developed in (Mistree *et al.*, 1993) was implemented to the solar irrigation problem. The comparison of the individual best solutions obtained by the Hyp-AP against the solutions obtained in Chen *et al.* (1995) is given in Table 1.

In Fig. 3 the solution distributions obtained using both algorithms form a plane in three-dimensional objective space. This may be attributed to the feature of the objective space (characteristic of the problem) itself, where dominant/optimal solutions mostly exist on the plane. This sort of distribution may not come to existence if these algorithms were to be implemented on other problems that do not share similar features of the objective space.

Table 1 The best individual solutions obtained by the w-AP, Hyp-AP and DSIDES method

	Hyp-AP	DSIDES (Chen et al. (1995)
f_1	20.7987	20.003
f_2	17.4495	19.45
f_3	148927	141143
x_a	0.620651	3
x_b	456.5	450
x_c	524.661	550
x_d	0.038835	0.0258
Z_a	302.707	0.02577
Z_b	807.545	0.02577
HVI	2784827	1822616

One of the advantages of the Hyp-AP method is that it obtains highly dominant solutions in MO scenarios. Besides, the Hyp-AP method also obtains optimal/dominant solutions that construct the Pareto frontier without compromising any of the objectives. For this solar irrigation application problem, the best individual solutions obtained by the Hyp-AP method were more dominant from the DSIDES approach in Chen *et al.*, (1995) by 52.79% respectively. This is because the AP component is a metaheuristic algorithm that successively improves the solution quality during execution while the DSIDES (Chen *et al.*, 1995) algorithm is nonlinear programming (NLP) based classical approach. In addition to that, the HVI aids the AP towards high-quality solutions as it searches the multi-objective space systematically. Besides, since the HVI mechanism is highly suited for multi-dimensional search spaces, this makes searching for optimal/dominant solutions in cases such as this more effective and computationally effective.

In this work, both algorithms Hyp-AP methods produced feasible solutions where no constraints of the model were broken. Besides, the Hyp-AP method performed stable calculations during the program executions. A new local optimum was discovered in this work by the implementation of the Hyp-AP method (see Table 1).

5 Conclusions and Recommendations

A new local maximum and an efficient construction of the Pareto frontier were achieved using the Hyp-AP approach. The HVI was used to gauge the individual solutions produced by the Hyp-AP and the DSIDES (Chen *et al.*, 1995) algorithms. In future works, other meta-heuristic algorithms such as GP (Koza, 1992), GA (Holland, 1992) and other hybrid algorithms should be embedded with the HVI. These approaches should then be tested with other industrial application problems.

References

Elamvazuthi, I., Ganesan, T., Vasant, P.: A Comparative Study of HNN and Hybrid HNN-PSO Techniques in the Optimization of Distributed Generation (DG) Power Systems. In: International Conference on Advance Computer Science and Information System (ICACSI 2011), pp. 195–199 (2011)

Ganesan, T., Elamvazuthi, I., Shaari, K.Z.K., Vasant, P.: Swarm intelligence and gravitational search algorithm for multi-objective optimization of synthesis gas production. J. of Appl. Energy 103, 368–374 (2013)

Helikson, H.J., Haman, D.Z., Baird, C.D.: Pumping water for irrigation using solar enegy, Fact Sheet (EES-63). University of Florida

Wong, Y.W., Sumathy, K.: Thermodynamic analysis and optimization of a solar thermal water pump. Appl. Thermal Eng. 21, 613–627 (2001)

Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms - A comparative case study. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 292–301. Springer, Heidelberg (1998)

Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation 3(4), 257–271 (1999)

Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research 181(3), 1653–1669 (2007)

Zelinka, I.: Analytic programming by Means of Soma Algorithm. In: Proceeding of the 8th International Conference on Soft Computing, Mendel 2002, Brno, Czech Republic, pp. 93–101 (2002)

Zelinka, I., Oplatkova, Z.: Analytic programming – Comparative Study. In: Second International Conference on Computational Intelligence, Robotics, and Autonomous Systems CIRAS 2003, Singapore (2003)

Chen, W., Kwok-Leung, T., Allen, J.K., Mistree, F.: Integration of the Response Surface Methodology with the compromise decision support problem in developing a general robust design procedure. In: Azarm, S., et al. (eds.) Advances in Design Automation, vol. 82-2, ASME, New York (1995)

Koza, J.R.: Genetic Programming: On the Programming of Computers by means of Natural Selection. MIT Press, USA (1992)

Ryan, C., Collins, J.J., Neill, M.O.: Grammatical evolution: Evolving programs for an arbitrary language. In: Banzhaf, W., Poli, R., Schoenauer, M., Fogarty, T.C. (eds.) EuroGP 1998. LNCS, vol. 1391, p. 83. Springer, Heidelberg (1998)

Koza, J.R.: Genetic Programming: On the Programming of Computers by means of Natural Selection. MIT Press, USA (1992)

Mistree, F., Hughes, O.F., Bras, B.A.: The Compromise Decision Support Problem and the Adaptive Linear Programming Algorithm. In: Structural Optimization: Status and Promise, pp. 247–286. AIAA, Washington, D.C (1993)

Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology. In: Control and Artificial Intelligence. MIT Press, USA (1992)

Forecasting of Time Series with Fuzzy Logic

Petr Dostál

Abstract. There are different methods which can be used for the support of forecasting. Nowadays the new theories of soft computing are used for these purposes. There could be mentioned fuzzy logic, neural networks and some other methods. The aim of the paper is focused on the use of fuzzy logic for forecasting purposes. The advantage of the use of fuzzy logic is in processing imprecision, uncertainty, vagueness, semi-truth, or approximated and nonlinear data. The applications on the stock market have specific features in comparison with others. The processes are focused on private corporate attempts at money making; therefore, the details of applications, successful or not, are not published very often. The fuzzy logic helps in decentralization of decision-making processes to be standardized, reproduced, and documented, that is an important factor in the business field. It was proved by the tests in the practice that the presented case studies had their justness to be used as a support for a decision making on the stock market.

1 Introduction

There are various forecasting methods used in economics, and finance: The classical ones and the methods using soft computing. There could be mentioned fuzzy logic, neural networks, and some others methods. The forecasting processes are very complicated because they include political, social, psychological, economic, financial, and other phenomena. Many variables are difficult to measure; they are characterized by imprecision, uncertainty, vagueness, semi-truth, approximation, nonlinearity, etc. The advantage of the use of fuzzy logic is in processing these data. The fuzzy logic could be used under these conditions. The forecasting of the operation in the stock market are quite often reduced to the decision whether to operate in the stock market or not and/or forecasting the trends of time series (increasing, constant, decreasing trend; sell, hold, buy position). Therefore, the aim of the paper and researched field are focused on the use of fuzzy logic for forecasting purposes.

Petr Dostál
Brno University of Technology, Faculty of Business and Management,
Institute of Informatics, Kolejní 2906/4, 612 00 Brno, Czech Republic

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 155–161.
DOI: 10.1007/978-3-319-00542-3_16

The program FuzzyTech® is used. The chapter is focussed only on applications. The fuzzy logic theory is described in many books such as [1], [2], [3], [4], [5], [6], [7], [8], [9].

2 The Case Study 1

Let us mention an example of the use of fuzzy logic for prediction of the time series. At first it is necessary to say that the time series is a sequence of values that are dependent on time *t*. The value at time $t = 1$ is denoted x_1, at time $t = 2$ is denoted x_2, and so on, and the value in time $t = \mathrm{N}$ is denoted x_N, when *N* signifies the number of values in the time series. The time series can be expressed as a vector of values $x = (x_1, x_2, \ldots, x_\mathrm{N})$. For needs of prediction we specify that the value x_N will be the last known value of the time series and it will correspond to the present. The value $\bullet x_{\mathrm{N}+1}$ will be the first future value predicted, the value $\bullet x_{\mathrm{N}+2}$ will be the second value predicted, etc. (The symbol • is be used to denote predicted values.) The interval between measurement is very often constant, then $\Delta = t_2 - t_1 = t_3 - t_2 = \ldots = t_\mathrm{N} - t_{\mathrm{N}-1} = \text{const}$. This interval in the economy (in contrast to technical sciences) has values in the range of minutes, hours, days, weeks, months, years, and their fractions. In this respect we speak about time series with very high (minutes), high (hours), medium (days), low (weeks), and very low (year) frequencies.

The following verbal notes were chosen for the solution of predictions by means of the FuzzyTech program: Delta_1 = $x_\mathrm{N} - x_{\mathrm{N}-1}$, Delta_2 = $x_{\mathrm{N}-1} - x_{\mathrm{N}-2}$, Delta_3 = $x_{\mathrm{N}-2} - x_{\mathrm{N}-3}$, Delta_4 = $x_{\mathrm{N}-3} - x_{\mathrm{N}-4}$ (the signs of these differences express the trend of the time series). The build-up model for prediction has four input variables Delta_1, Delta_2, Delta_3, Delta_4, one rule box, and one output variable Prediction. See the fuzzy model in Fig. 1.

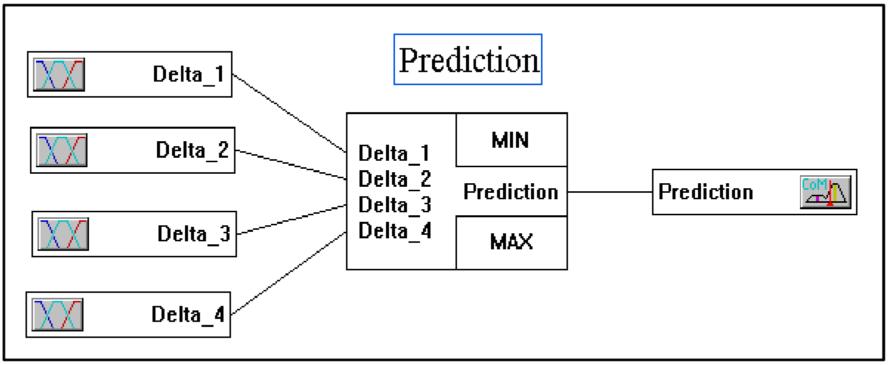


Fig. 1 Fuzzy logic – prediction

The input variables have five attributes defined by values Delta, specified by their signs and the size of difference of neighbouring values (high positive, positive, zero, negative, high negative difference). The shapes Λ, S, and Z are used as membership functions. The output variable Prediction has five attributes that evaluate the future course of the time series (high increase, increase, stagnation, decrease, high decrease), specifying the situation at time N+1 (the value of prediction •x_{N+1}). The membership functions are spline curves of Λ, Π, S, and Z shapes.

The procedure followed by the program FuzzyTech includes set up of membership functions of input variables Delta_1, Delta_2, Delta_3, Delta_4 and the fuzzy rule box. The fuzzy rule box must be set up on the basis of knowledge, preferably by the experts who understand the problem. The setup of a fuzzy rule box depends on the type of solved case. For example a suitable rule can be similar to the following:

When inputs Delta_1 and Delta_2 and Delta_3 and Delta_4 are negative, it means that the time series is decreasing and increase of the time series Prediction is expected in future.

This situation can be verbally described in capital markets: after long decrease of share values they tend to start increase with 90% probability. The rule can be described by this form:

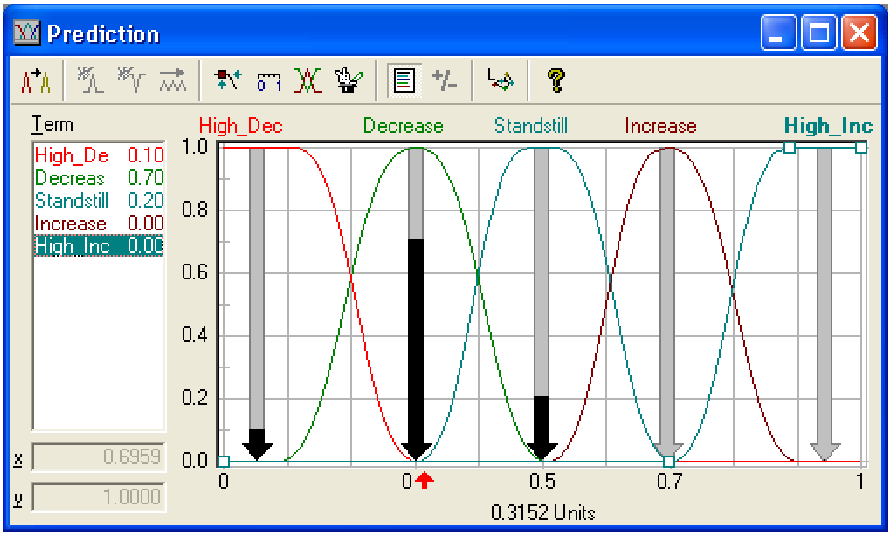
<When> Delta_1 < 0 <And> Delta_2 < 0 <And> Delta_3 < 0 <And> Delta_4< 0 <Then> Prediction = Increase <With> s = 0.90.

The rule for the opposite case can be verbally described in a capital market: after long increase of share values they tend to start decrease. The rule has the form

<When> Delta_1 > 0 <And> Delta_2 > 0 <And> Delta_3> 0 <And> Delta_4 > 0 <Then> Prediction = Decrease <With> s = 0.90.

It is necessary to set up other rules that are combinations of these two described extreme variants. Figure 2 presents setup attributes and membership functions for the output variable Prediction.

For another case an unsatisfactory model could be set up and it is necessary to choose another number of variables, to define variables in another way, to choose other attributes and membership functions. The model must be tuned to give us good predictions.



Term	
High_De	0.10
Decreas	0.70
Standstill	0.20
Increase	0.00
High_Inc	0.00

x 0.6959

y 1.0000

0.3152 Units

Fig. 2 Membership functions of output variable Prediction

3 The Case Study 2

Let us mention an example that solves the problem of decision making in capital markets: whether to trade in the stock market or not. The model for the FuzzyTech program is presented in Fig. 3.

The input variables and their attributes are as follows: Margin (insignificant, significant), Interest rates (low, medium, high), Strength of market (low, medium, high), and Trend the course of time series (deterministic, stochastic).

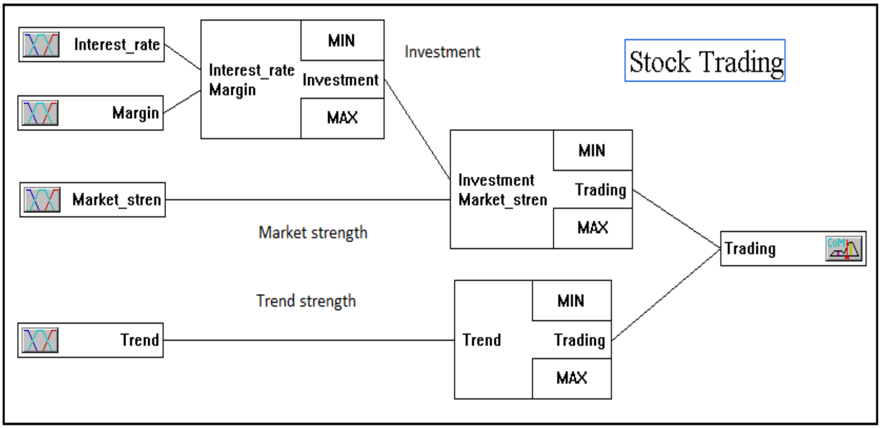


Fig. 3 The diagram of the model – the stock market

The rules and attributes are as follows: the box Investment (unsuitable, neutral, suitable) determines the rule of desirability of depositing money in a stock market; the first block Trading (yes, no) evaluates whether trading in the market is suitable from the point of profitability of investment and the strength of the market; the second block Trading (yes, no) gives the decision for trading when the time series is stochastic, meaning that there is no possibility to make a good prediction of future development of time series. The output variable Trading evaluates whether to trade with share, index, commodity, or currency ratio. The membership functions were used in the shapes of Λ, Π, S, and Z.

4 The Case Study 3

Fuzzy logic can be used in decision making in the stock market. The model will be used for decisions whether to buy, sell, or hold with a share or index. The FuzzyTech program is used for this purpose when the inputs are the values from various analyses and information from the Internet. The diagram of such a model is presented in Fig. 4.

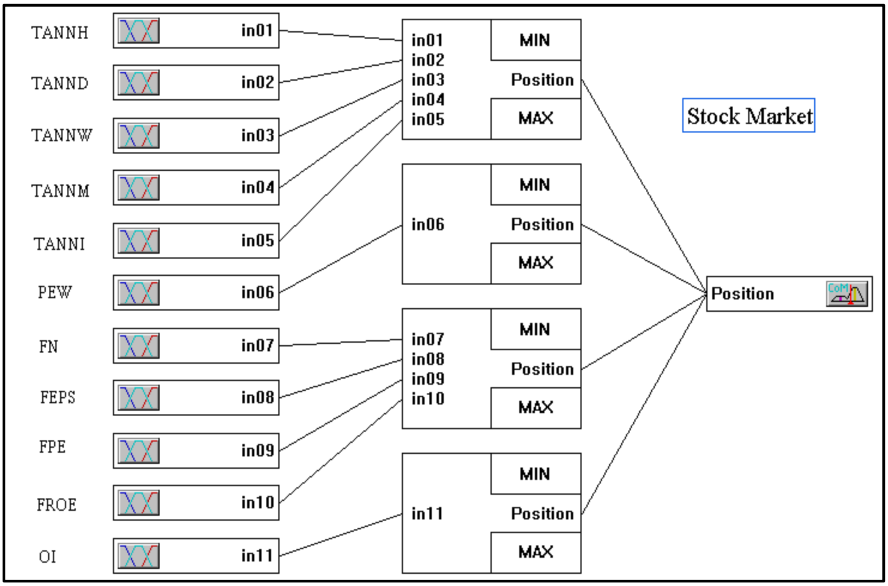


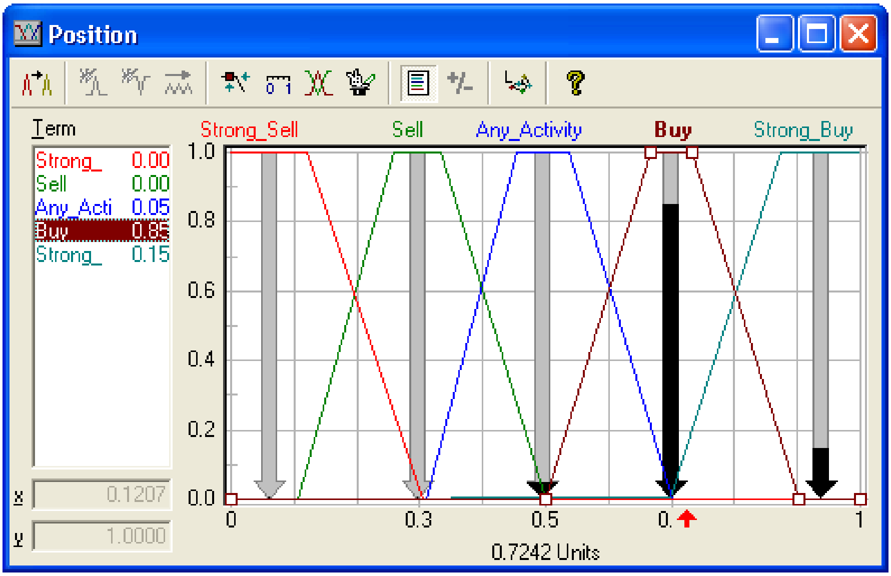
Fig. 4 Diagram of fuzzy model – stock market

The model has eleven inputs with seven attributes, four rule boxes, and one output variable with five attributes. The input variables are the information obtained from technical analyses represented by predictions of share prices by means of a neural network with hour TANNH, day TANND, week TANNT and month periodicity TANNM; predictions of values of an index having influence on predictions of a searched share TANNI; psychological analyses represented by means of Elliott's waves PEW; fundamental analyses represented by information from news FN, economic indexes EPS (earning per share) FEPS, P/E (price-to-earnings ratio) FPE, and ROE (return on equity) FROE, and other knowledge OI, such as for example intuition. The attributes of all input variables are the same and they express the rate of influence on the tendency of a time series (high, medium, low positive, neutral, low, high negative). A positive influence indicates the influence of an increasing tendency of a time series, and a negative influence signifies the influence of a decreasing tendency of a time series.

The rule boxes include technical analyses, psychological, fundamental, and other analyses.

The output variable Position tells the investor what he/she has to do in the stock market: Strong Sell, Sell, Hold, Buy, Strong Buy.

As a membership function, spline curves of shapes of Λ, Π, S, and Z were used. The attributes and membership function for output variable Position are presented with result to Buy in Fig. 5.



Term	
Strong_	0.00
Sell	0.00
Any_Acti	0.05
Buy	0.85
Strong_	0.15

x 0.1207

y 1.0000

Fig. 5 The attributes and membership function of output variable – Position

5 Conclusion

The aim of the paper is focused on the use of fuzzy logic for forecasting purposes. The advantage of the use of fuzzy logic is in processing imprecision, uncertainty, vagueness, semi-truth, approximated and nonlinearity data. The forecasting of the operation in the stock marked are quite often reduced on the decision whether to operate in the stock or not and/or forecasting of the trends of time series (increasing, constant, decreasing trend; sell, hold, buy position). Therefore, the research field was focussed on this usage of fuzzy logic in forecasting. It was proved by the tests in practice, that the presented case studies have their justness to be used as a support for decision making in the stock market.

Examples mentioned above are only a fraction of possible use of fuzzy logic for forecasting. We can use it for any time series created by indexes, shares, commodities, currency ratios, etc. The fuzzy logic theory could be used also in business, economics, finance, public services, etc. [10], [11]. The fuzzy logic helps not only to support decision making, but also to standardize, reproduce, and document processes, which are important factors in the business field.

References

[1] Alliev, A., Alliev, R.: Soft Computing and Its Applications. World Scientific Publishing, UK (2002)

[2] Altrock, C.: Fuzzy Logic & Neurofuzzy -Applications in Business & Finance. Prentice Hall, USA (1996)

[3] Dostál, P.: Advanced Decision Making in Business and Public Services. CERM, Brno (2011)

[4] Dostál, P.: The use of soft computing methods for forecasting in business, their applications in practice. In: Zelinka, I., Snasel, V., Rössler, O.E., Abraham, A., Corchado, E.S. (eds.) Nostradamus: Mod. Meth. of Prediction, Modeling. AISC, vol. 192, pp. 49–60. Springer, Heidelberg (2013)

[5] Chen, S., Wang, P., Wen, T.: Computational Intelligence in Economic and Finance. Springer (2004)

[6] Chen, S., Wang, P., Wen, T.: Computational Intelligence in Economic and Finance, vol. II. Springer (2007)

[7] Kazabov, K.: Neuro-Fuzzy – Techniques for Intelligent Information Systems. Physica-Verlag (1998)

[8] Klir, G.J., Yuan, B.: Fuzzy Sets and Fuzzy Logic, Theory and Applications. Prentice Hall, New Jersey (1995)

[9] Ribeiro, R., Yager, R.: Soft Computing in Financial Engineering. A Springer Verlag Company (1999)

[10] Dostál, P.: The Use of Optimization Methods in Business and Public Services. In: Zelinka, I., Snasel, V., Abraham, A. (eds.) Handbook of Optimization. ISRL, vol. 38, pp. 717–777. Springer, Heidelberg (2013)

[11] Dostál, P.: The Use of Soft Computing for Optimization in Business, Economics, and Finance. In: Vasant, P. (ed.) Meta-Heuristics Optimization Algorithms in Engineering, Business, Economics, and Finance,.IGI Globe, USA (2012)

[12] Li, Z., Halong, W.A., Chen, G.: Integration of Fuzzy Logic and Chaos Theory. Springer (2006)

[13] Peters, E.E.: Fractal Market Analysis—Applying Chaos Theory to Investment & Economics. John Wiley, USA (1994)

[14] Peters, E.E.: Chaos and Order in the Capital Markets: A New View of Cycles, Prices. Wiley Finance Edition, USA (1996)

[15] Trippi, R.R.: Chaos & Nonlinear Dynamics in the Financial Markets. Irwin Professional Publishing, USA (1995)

[16] The Mathworks.MATLAB – User’s Guide, The MathWorks (2012)

[17] The Mathworks. MATLAB – Fuzzy Logic Toolbox - User’s Guide, The MathWorks (2012)

Unknown Input Proportional Integral Observer Design for Chaotic TS Fuzzy Models

T. Youssef, Mohammed Chadli*, Ivan Zelinka, and M. Zelmat

Abstract. In this paper, the chaos synchronization problem is treated by an unknown input proportional integral observer (PIO) for a Takagi-Sugeno (TS) fuzzy chaotic model subject to unknown input and unmeasurable decision variables. This unknown input is considered like a message to encode by a chaotic system then to decode or to reconstruct by the PIO after to be transmitted by a public transmission canal in order to a secure communication system. In our case, the unknown input affects both state and output of the chaotic system. The synthesis conditions of this PIO are based on the hypothesis that the unknown input is under polynomial form with its k^{th} derivative zero. At the end the measurable decision variables are also considered in this work like a particular case. The Lyapunov theory is used to develop the stability conditions of the unknown input PIO in LMIs formulation. A simulation example is proposed through a TS fuzzy chaotic model to validate the proposed design.

Keywords: TS fuzzy models, Unmeasurable decision variables, Unknown input reconstruction, Proportional Integral Observer, Synchronization, Chaotic system.

1 Introduction

Chaotic systems are well-known for their complex dynamical behavior, and the chaos is one of the most important properties of dynamical systems. The chaos is a

M. Chadli
The Laboratory of Modeling, Information & Systems (MIS), University of Picardie Jules Verne (UPJV), 33 rue Saint-Leu, 80039 Amiens Cedex 1, France
e-mail: mchadli@u-picardie.fr

T. Youssef · M. Zelmat
The Laboratory of Automatic Applied (LAA), M'hamed Bougara University of Boumerdès (UMBB), Algeria

I. Zelinka
Department of Computer Science, Faculty of Electrical Engineering and Computer Science VSB-TUO, Ostrava-Poruba Czech Republic
e-mail: ivan.zelinka@ieee.org

* Corresponding author.

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 163–174.
DOI: 10.1007/978-3-319-00542-3_17

source of the generation of oscillation and instability. Indeed, the chaotic systems are highly sensitive to initial conditions and parameters variation, which give them a long-term unpredictable behavior [1]. This phenomenon could lead systems to undesirable performance or even exhibit instability.

The interest in chaotic systems lies on nonstandard control problems including stabilization and synchronization. The idea of synchronization between two identical chaotic systems was first introduced by Carroll and Pecora [2, 3] in 1990 based on the Lorenz's chaotic system [4].The last has led to a new era in the study of chaos. Since chaos control and synchronization methods have attracted a great deal of interest among researchers from different disciplines.

The chaos has many potential applications such as physical systems, chemical reactions, system identification, biological systems, secure communication (see, e.g. [5-10] and references therein). Furthermore, various approaches have been developed for chaos synchronization including feedback control [11], adaptive control [12], impulsive control [13], fuzzy control [14], lag synchronization [15] and sliding mode control [16]. Particularly the chaos is ideal to hide information efficiently and securely [17] since the chaotic signals have the characteristic of broadband, noise-like, and difficult to predict whence their interest to applied them in engineering applications of secure communication. To recover the message the receiver has to synchronize with the transmitter in secure communications [18]. In this field, many methods have been proposed such as chaos masking [19], chaos shift key [20] and chaos modulation [21]. Among various control and synchronization schemes for chaotic systems, the TS fuzzy systems have become one of the most fields of control since Takagi and Sugeno (TS) have proposed the TS fuzzy models [22]. Indeed, a highly nonlinear system such as chaotic system can be represented exactly by TS fuzzy model. The last is described by fuzzy IF–THEN rules where the consequent parts represent local linear models. Then, the linear models are combined by using nonlinear activation functions in order to obtain the nonlinear behavior of the plant. The TS fuzzy models have been investigated widely in control and synchronization schemes, e.g. [23-25]. Because of their simple structure with local dynamics, the conventional linear system theory can be easily applied to the analysis and synthesis of these schemes.

There is well-known, states are partially or fully not accessible in many practical control problems. Therefore, a state observer can be used to estimate the measurements of unavailable sensors or in the event of failures. For these reasons, the observer design of chaotic systems is essential. Recently, some researches which deal with unknown input observer in secure communication have been studied in [26-29]. In this framework, the authors in [26] present a new scheme for the secure communication of information based on chaotic synchronization via smooth adaptive unknown input observer and robust to channel noise. The problem of chaos secure communication is also considered in [27] within the master-slave configuration. Where, some messages can be regarded as unknown disturbance of parameters in chaotic systems. In [28] the authors propose a robust adaptive high-gain fuzzy observer design scheme and its application to synchronization and secure communication of chaotic systems where their states are supposed unavailable and their parameters are unknown. In [29] is designed an unknown input observer for continuous and discrete-time of chaotic TS fuzzy model by using Lyapunov stability theory and Linear Matrix Inequality (LMI) formulation. In addition, to improve the performance of the observer, the pole assignment in a LMI region is proposed. All previous studies have not discussed the case of the unmeasurable decision variables in activation functions for chaotic TS fuzzy models. This is our contribution in the present study.

In this paper, inspired by previous researches [28, 29], we consider the problem of chaotic synchronization scheme for chaotic TS fuzzy models based on the polynomial unknown input Proportional Integral Observer (PIO) with unmeasurable decision variables. The synthesis of the proposed PIO is derived by using Lyapunov stability theory and LMI formulation. We note that, our PIO estimate simultaneous the state and the unknown input. In the formwork of secure communication, the unknown input is be supposed the information signal (message) which is injected into chaotic TS fuzzy model (transmitter or driver system). The message is transmitted via public canal and later, the receiver signal should be treated in order to reconstruct them by the PIO (receiver or response system). Finally, a simulation example is included to illustrate the results developed in this paper.

In this work the modified PIO is considered for TS fuzzy model subject to unknown input. The latter is considered as a message to be encoded by a chaotic system then to be decoded or reconstruct by the observer in a secure communication procedure. Indeed, supplementary parameters are introduced to compensate the effect due to unmeasurable decision variables and unknown polynomial input. The particular case when the decision variables are supposed measurable is also considered. The proposed approach treats the k^{th} derivative of unknown input like zero. The design conditions are established on the basis of the Lyapunov theory and LMIs formulation.

This paper is organized as follows. In section II, we present the structure of TS fuzzy model with unmeasurable decision variables. This model is subject to unknown input which can affect the dynamics and the output signals. In section III, the structure and synthesis of the unknown input PIO is presented. Then the particular case when the decision variables are supposed measurable is studied. Finally in section IV, a simulation example is given to show through a TS fuzzy chaotic model the good estimation of both state and unknown input like a message in a secure communication system.

2 Structure of the Unknown Input TS Fuzzy Model

Consider a TS Fuzzy model with unmeasurable decision variables and subject to unknown input:

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^{r} \mu_i(x)\big(A_i x(t) + B_i u(t) + F_i v(t)\big) \\ y(t) = Cx(t) + Fv(t) \end{cases} \tag{1}$$

where $x(t) \in R^n$ is the state vector, $u(t) \in R^{n_u}$ is the known input vector, $v(t) \in R^{n_v}$ is the unknown input, and $y(t) \in R^{n_y}$ represents the output vector. $A_i \in R^{n \times n}$ are the state matrices, $B_i \in R^{n \times n_u}$ are the input influence matrices, $F_i \in R^{n \times n_v}$ and $F \in R^{n_y \times n_v}$ are the unknown input influence matrices, and $C \in R^{n_y \times n}$ is the output matrix. The $\mu_i(x)$ represent the activation functions which depend on the state $x(t)$ of the system. These functions have the following properties:

$$\begin{cases} \sum_{i=1}^{r} \mu_i(x) = 1, & \forall\, t \geq 0 \\ 0 \leq \mu_i(x) \leq 1, & \forall\, i \in \{1, .., r\} \end{cases}$$

where r represents the local models number.

Hypothesis: The unknown input $v(t)$ is a polynomial in time function, of *k-1* degree, and its *k*th derivative is equal to zero. With the following notations:

$$\begin{cases} \dot{v}(t) = v_1(t) \\ \dot{v}_1(t) = v_2(t) \\ \vdots \\ \dot{v}_{k-1}(t) = v_k(t) \\ v_k(t) = 0 \end{cases} \tag{2}$$

In the diagnosis framework, the assumption of the unknown input $v(t)$ in polynomials form allows considering a large range of faults.

3 Structure and Synthesis of the Unknown Input Observer

The considered PIO is:

$$\begin{cases} \dot{\hat{x}}(t) = \sum_{i=1}^{r} \mu_i(\hat{x})\Big(A_i \hat{x}(t) + B_i u(t) + F_i \hat{v}(t) + K_{Pi}\big(y(t) - \hat{y}(t)\big)\Big) + z_x(t) \\ \hat{y}(t) = C\hat{x}(t) + F\hat{v}(t) \\ \dot{\hat{v}}(t) = \sum_{i=1}^{r} \mu_i(\hat{x}) K_{Ii}\big(y(t) - \hat{y}(t)\big) + \hat{v}_1(t) + z_v(t) \\ \dot{\hat{v}}_j(t) = \sum_{i=1}^{r} \mu_i(\hat{x}) K_{Ii}^{j}\big(y(t) - \hat{y}(t)\big) + \hat{v}_{j+1}(t) + z_{vj}(t) \qquad \text{for } j{:}\ 1 \ldots k-1 \\ \qquad \text{if } j = k-1, \quad \hat{v}_{j+1}(t) = 0 \end{cases} \tag{3}$$

where K_{Pi} and K_{Ii}, K_{Ii}^{j} represent the proportional and integral gains respectively. The variables $z_x(t)$ and $z_v(t)$, $z_{vj}(t)$ are introduced in order to compensate the influence of the unmeasurable decision variables.

The proposed PIO (3) not only allows the states estimation of TS model (1) subjected of unknown input but also the reconstruction of this input in the presence of unmeasurable decision variables.

Based on the *hypothesis*, the TS fuzzy model (1) and the PIO (3) can be written respectively under the following augmented forms:

$$\begin{cases} \dot{x}_a(t) = \sum_{i=1}^{r} \mu_i(x)(\bar{A}_i x_a(t) + \bar{B}_i u(t)) \\ y(t) = \bar{C} x_a(t) \end{cases} \tag{4}$$

$$\begin{cases} \dot{\hat{x}}_a(t) = \sum_{i=1}^{r} \mu_i(\hat{x}) \left(\bar{A}_i \hat{x}_a(t) + \bar{B}_i u(t) + \bar{K}_i\big(y(t) - \hat{y}(t)\big)\right) + z(t) \\ \hat{y}(t) = \bar{C} \hat{x}_a(t) \end{cases} \tag{5}$$

where

$$x_a(t) = \begin{bmatrix} x(t) \\ v(t) \\ v_1(t) \\ \dots \\ v_{k-1}(t) \end{bmatrix}, \hat{x}_a(t) = \begin{bmatrix} \hat{x}(t) \\ \hat{v}(t) \\ \hat{v}_1(t) \\ \dots \\ \hat{v}_{k-1}(t) \end{bmatrix}, \ z(t) = \begin{bmatrix} z_x(t) \\ z_v(t) \\ z_{v1}(t) \\ \dots \\ z_{vk-1}(t) \end{bmatrix} \tag{6a}$$

with

$$e_a(t) = x_a(t) - \hat{x}_a(t), \ e_{ay} = y(t) - \hat{y}(t) \tag{6b}$$

$$\bar{A}_i = \begin{bmatrix} A_i & F_i & 0 & 0 & \dots & 0 \\ 0 & 0 & I_{n_v} & 0 & \dots & 0 \\ 0 & 0 & 0 & I_{n_v} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & I_{n_v} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \ \bar{B}_i = \begin{bmatrix} B_i \\ 0 \\ 0 \\ \dots \\ 0 \end{bmatrix}, \ \bar{K}_i = \begin{bmatrix} K_{Pi} \\ K_{Ii} \\ K_{Ii}^1 \\ \dots \\ K_{Ii}^{k-1} \end{bmatrix} \tag{6c}$$

$$\bar{C} = [C \quad F \quad 0 \quad \dots \quad 0] \tag{6d}$$

and I_{n_v} is the identity matrix of dimension n_v.

3.1 Synthesis of the Unknown Input PIO

The dynamics of state estimation error $e_a(t)$ is represented by:

$$\dot{e}_a(t) = \sum_{i=1}^{r} \mu_i(\hat{x}) \bar{\mathcal{A}}_i e_a(t) + \bar{\Delta} A x_a(t) + \bar{\Delta} B u(t) - z(t) \tag{7}$$

where

$$\bar{\mathcal{A}}_i = \bar{A}_i - \bar{K}_i \bar{C}, \ \bar{\Delta} A = \textstyle\sum_{i=1}^{r} \bar{\mu}_i \bar{A}_i, \ \bar{\Delta} B = \sum_{i=1}^{r} \bar{\mu}_i \bar{B}_i, \ \bar{\mu}_i = \mu_i(x) - \mu_i(\hat{x}) \tag{8}$$

Remark: The convex sum property of activation functions allows to write $-1 < \bar{\mu}_i < 1$, hence the variables matrices $\bar{\Delta} A$ and $\bar{\Delta} B$ are bounded and the following properties are verified:

$$\|\bar{\Delta} A\| \leq \delta_1, \quad \delta_1 = \textstyle\sum_{i=1}^{r} \delta_{1i} \ \text{ and } \ \|\bar{\Delta} B\| \leq \delta_2, \quad \delta_2 = \sum_{i=1}^{r} \delta_{2i} \tag{9}$$

with $\delta_{1i} > 0$ and $\delta_{2i} > 0$ are respectively the matrices norms $\bar{A}_i$ and $\bar{B}_i$.

Lemma: For any X and Y of appropriate dimensions matrices, the following property is verified:

$$X^TY + Y^TX \leq \lambda\, X^TX + \lambda^{-1}Y^TY \quad \textit{with} \quad \lambda > 0$$

Theorem: The system (7) *is asymptotically stable if there exist a matrix* $P = P^T > 0$, *matrices* $\bar{M}_i$ *and the positive scalars* α *and* α_0 *such as:*

$$\begin{bmatrix} P\bar{A}_i + \bar{A}_i^T P - \bar{M}_i\bar{C} - \bar{C}^T\bar{M}_i^{\ T} + \alpha_0\delta_1^2 I & P \\ P & -\alpha I \end{bmatrix} < 0 \qquad (10)$$

where the parameters $\bar{A}_i$, $\bar{C}$ *and* δ_1, δ_2 *are defined in* (6c), (6d) *and* (9). *The observer* (3) *parameters are given by:*

$$\bar{K}_i = P^{-1}\bar{M}_i$$

and

$$\begin{cases} z = 0 & if \quad \left|e_{a_y}\right| < \varepsilon \\ z = \sigma_1\, \delta_1^2 \dfrac{\hat{x}_a^T\hat{x}_a}{2e_{a_y}^T e_{a_y}} P^{-1}\bar{C}^T e_{a_y} + \sigma_2\, \delta_2^2 \dfrac{u^Tu}{2e_{a_y}^T e_{a_y}} P^{-1}\bar{C}^T e_{a_y} & if \quad \left|e_{a_y}\right| \geq \varepsilon \end{cases}$$

with variables $\hat{x}_a$, z *and* e_{ay} *are defined in* (6a), (6b) *and* $\sigma_1 = \left(\frac{\alpha_0}{\lambda}\right)$, $\sigma_2 = \left(\frac{\alpha\alpha_0}{\alpha(1+\lambda)-\alpha_0}\right)$ *where* ε *is a small scalar arbitrarily fixed.*

Proof: Consider the Lyapunov function $V(t) = e_a^T(t)Pe_a(t)$ where $P = P^T > 0$. The time-derivative of $V(t)$ allows writing:

$$\dot{V} = \sum_{i=1}^{r} \mu_i(\hat{x})(e_a^T(\bar{\mathcal{A}}_i^T P + P\bar{\mathcal{A}}_i)e_a) + x_a^T\bar{\Delta}A^T\, Pe_a + e_a^T P\bar{\Delta}Ax_a$$
$$+u^T\bar{\Delta}B^T Pe_a + e_a^T P\bar{\Delta}Bu - z^T Pe_a - e_a^T Pz \qquad (11)$$

Then the *lemma* allows us to write:

$$\dot{V} \leq \sum_{i=1}^{r} \mu_i(\hat{x})(e_a^T(\bar{\mathcal{A}}_i^T P + P\bar{\mathcal{A}}_i)e_a) + \lambda_1\, \delta_1^2 x_a^T x_a + \lambda_1^{-1} e_a^T P^2 e_a$$
$$+u^T\bar{\Delta}B^T Pe_a + e_a^T P\bar{\Delta}Bu - z^T Pe_a - e_a^T Pz \qquad (12)$$

Taking account (6b) and using again the *lemma*, we obtain:

$$\dot{V} \leq \sum_{i=1}^{r} \mu_i(\hat{x})(e_a^T(\bar{\mathcal{A}}_i^T P + P\bar{\mathcal{A}}_i)e_a) + u^T\bar{\Delta}B^T Pe_a + e_a^T P\bar{\Delta}Bu$$
$$+\lambda_1\delta_1^2(e_a^T e_a + \hat{x}_a^T\hat{x}_a + \lambda^{-1}\hat{x}_a^T\hat{x}_a + \lambda\, e_a^T e_a) + \lambda_1^{-1}e_a^T P^2 e_a - z^T Pe_a e_a^T Pz \quad (13)$$

Hence

$$\dot{V} \leq \sum_{i=1}^{r} \mu_i(\hat{x})(e_a^T(\bar{\mathcal{A}}_i^T P + P\bar{\mathcal{A}}_i + \alpha_0\delta_1^2 I + \lambda_1^{-1}P^2)e_a) + u^T\bar{\Delta}B^T Pe_a$$
$$+e_a^T P\bar{\Delta}Bu + \lambda_1(1 + \lambda^{-1})\delta_1^2\hat{x}_a^T\hat{x}_a - 2e_a^T Pz \qquad (14)$$

with $\alpha_0 = \lambda_1(1+\lambda)$ and using again the *lemma*, we can write:

$$\dot{V} \leq \sum_{i=1}^{r} \mu_i(\hat{x})(e_a^T(\bar{\mathcal{A}}_i^T P + P\bar{\mathcal{A}}_i + \alpha_0\delta_1^2 I + \lambda_1^{-1}P^2)e_a) + \lambda_2\delta_2^2 u^T u$$
$$+\lambda_2^{-1}e_a^T P^2 e_a + \lambda_1(1+\lambda^{-1})\delta_1^2\hat{x}_a^T\hat{x}_a - 2e_a^T Pz \qquad (15)$$

Hence

$$\dot{V} \leq \sum_{i=1}^{r} \mu_i(\hat{x})(e_a^T(\bar{\mathcal{A}}_i^T P + P\bar{\mathcal{A}}_i + \alpha_0\delta_1^2 I + \propto^{-1} P^2)e_a) + \sigma_1\delta_1^2\hat{x}_a^T\hat{x}_a$$
$$+\sigma_2\delta_2^2 u^T u - 2e_a^T Pz \qquad (16)$$

with $\propto^{-1} = (\lambda_1^{-1} + \lambda_2^{-1})$, $\sigma_1 = \lambda_1(1+\lambda^{-1}) = \left(\frac{\alpha_0}{\lambda}\right)$, $\sigma_2 = \lambda_2 = \left(\frac{\alpha\alpha_0}{\alpha(1+\lambda)-\alpha_0}\right)$.

Substituting the variable expression of z into (16), we get:

$$2e_a^T Pz = 2\, e_a^T P\, \sigma_1\delta_1^2 \frac{\hat{x}_a^T\hat{x}_a}{2e_{a_y}^T e_{a_y}} P^{-1}\bar{C}^T e_{a_y} + 2\, e_a^T P\sigma_2\delta_2^2 \frac{u^T u}{2\, e_{a_y}^T e_{a_y}} P^{-1}\bar{C}^T e_{a_y}$$
$$= \sigma_1\delta_1^2\hat{x}_a^T\hat{x}_a + \sigma_2\delta_2^2 u^T u \qquad (17)$$

with $e_{a_y} = \bar{C}e_a$ and $e_{a_y}^T = e_a^T\bar{C}^T$. Hence, the expression (16) becomes:

$$\dot{\boldsymbol{V}} \leq \sum_{i=1}^{r} \boldsymbol{\mu_i(\hat{x}) e_a^T\left(\bar{\mathcal{A}}_i^T P + P\bar{\mathcal{A}}_i + \alpha_0\delta_1^2 I + \propto^{-1} P^2\right)e_a} \qquad (18)$$

The stability condition $\dot{V}(t) < 0$ is verified if:

$$\Psi_i + \alpha^{-1}P^2 < 0 \qquad (19)$$

with $\Psi_i = \bar{\mathcal{A}}_i^T P + P\bar{\mathcal{A}}_i + \alpha_0\delta_1^2 I$ $(i = 1, \ldots r)$. Then apply the Schur complement to the condition (19) with variables change $\bar{\mathcal{A}}_i = \bar{A}_i - \bar{K}_i\bar{C}$ and $\bar{M}_i = P\bar{K}_i$, we get the linear matrix inequalities (10). This completes the proof. □

The resolution of these constraints allows to obtain the unknown input PIO gains $\bar{K}_i = P^{-1}\bar{M}_i$ and as well as the scalars α and α_0.

In the following, a simulation example is given through a chaotic system to validate this proposed approach.

4 Simulation Example

Consider a chaotic systems represented by TS Fuzzy models subjected to unmeasurable decision variables. This chaotic system is introduced in order to show the effectiveness of the unknown input PIO observer developed in the previous section at to the simultaneous reconstruction of states and message used in a secure communication scheme. The nonlinear model is the Lorenz's system [29][31] described by the following equations:

$$\begin{cases} \dot{x}_1 = -10x_1 + 10x_2 \\ \dot{x}_2 = 28x_1 - x_2 - x_1x_3 \\ \dot{x}_3 = x_1x_2 - \dfrac{8}{3}x_3 \end{cases} \tag{23}$$

Let the decision variable $x_1(t) \in [-30, 30]$, the Lorenz's system can be exactly represented by TS fuzzy model $\dot{x}(t) = \sum_{i=1}^{r} \mu_i(x)A_ix(t)$ where $x(t) = [x_1(t), x_2(t), x_3(t)]$, $\mu_1(x_1(t)) = \frac{30+x_1(t)}{60}$, $\mu_2(x_1(t)) = \frac{30-x_1(t)}{60}$ and

$$A_1 = \begin{bmatrix} -10 & 10 & 0 \\ 28 & -1 & -30 \\ 0 & 30 & -8/3 \end{bmatrix},\ A_2 = \begin{bmatrix} -10 & 10 & 0 \\ 28 & -1 & 30 \\ 0 & -30 & -8/3 \end{bmatrix}$$

The Lorenz chaotic attractor is shown in Fig. 1.

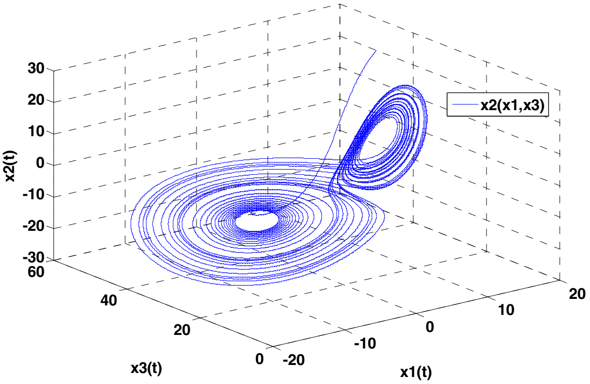


Fig. 1 Chaotic behavior of Lorenz fuzzy system

Therefore, the TS fuzzy model of the nonlinear Lorenz model (23) is:

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^{2} \mu_i(x)\big(A_ix(t) + E_iv(t)\big) \\ y(t) = Cx(t) + Ev(t) \end{cases} \tag{24}$$

with $B_1 = B_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$, $E_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$, $E_2 = \begin{bmatrix} 1 \\ 0.5 \\ 1 \end{bmatrix}$, $E = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $C = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$.

The unknown input $v(t)$ is considered as the message to be encoded by the TS fuzzy model (24). The output of this fuzzy model is transmitted using a public canal. On the receiving side, the unknown input PIO (3) plays the role of the decoder which allows the reconstruction of the message in order to secure the communication procedure.

The resolution of LMIs constraints (10) of the *theorem* leads to the PIO observer gains:

$$\overline{K}_i = \left[K_{Pi}^T \quad {K_{Ii}}^T \quad {K_{Ii}^1}^T\right]^T,\ Z = [{Z_x}^T \quad {Z_v}^T \quad {Z_{v1}}^T]^T$$

The gains of the PIO observer are given in the table:

Table

λ=2 10^3	$\alpha = 5.505\ 10^4$		$\alpha_0 = 0.001$	
i	*1*		*2*	
K_{pi}	-11.471 -22.084 -94.350	12.445 10.838 96.029	-11.057 02.100 97.909	11.926 11.650 -96.239
K_{Ii}	15.123	03.344	21.570	-03.137
K_{Ii}^1	83.889	-02.802	79.394	01.612

The considered unknown input like a transmitted message, with 2^{th} derivatives zero, is given in Fig. 2. The simulation results are carried out with the initial conditions: $x_0 = [1 \quad 1 \quad 1]$, $\hat{x}_0 = [0 \quad 0 \quad 0]$, and with $\varepsilon = 10^{-3}$.

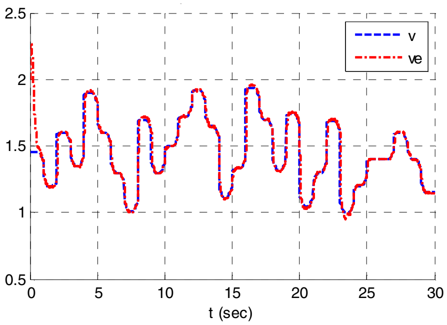
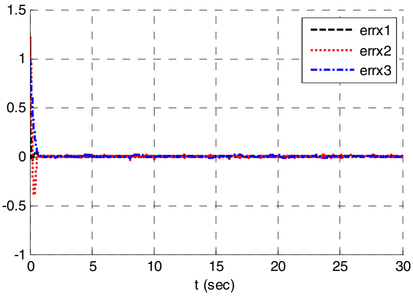


Fig. 2 The unknown input and its estimated



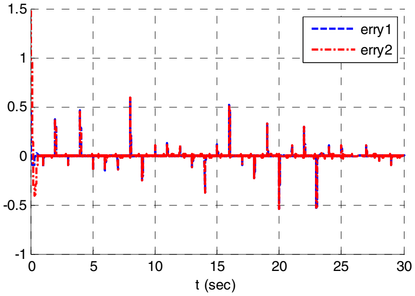


Fig. 3 The errors between states and their estimated

Fig. 4 The errors between outputs and their estimated

The Fig. 2 represents the unknown input and its estimated, and Fig. 3 their error dynamic. This error appears that at the level of the unknown input brusque variations. On the other hand when the unknown input is constant or slow variations, the error is equal to zero and we have a good reconstruction excepted around the time origin.

Fig. 3 and Fig. 4 represent respectively the errors dynamic of the states and the outputs. The latter is affected by changes of the unknown input. The simulation results show the good estimation of both state and unknown input.

5 Conclusion

In this work, a proportional integral observer for TS fuzzy model subjected to unknown input and unmeasurable decision variables is proposed. The unknown input is assumed to be a message to decode by this observer. The encoding is realized by using a chaotic fuzzy model in order to show the efficiency of our observer in there construction of this unknown input type. The latter is assumed in polynomial form and its k^{th} derivative is zero. The particular case when the decision variables are supposed measurable is also studied. Based on Lyapunov stability theory, the design conditions are established in LMIs form. In order to validate this approach a TS fuzzy chaotic models is given. The proposed example show the efficiency of the derived conditions since both state and unknown input are well estimated. The unknown input proportional integral observer with unmeasurable decision variables constitutes a good technique for message decoding as well as the reconstruction of chaotic system in order to secure a communication procedure.

References

[1] Jiang, G.P., Chen, G., Tang, W.K.S.: Stabilizing unstable equilibria of chaotic systems from a state observer approach. IEEE Trans. Circ. Syst. 51(6), 281–288 (2004)

[2] Pecora, L.M., Carroll, T.L.: Synchronization in chaotic systems. Phys. Rev. Lett. 64, 821–824 (1990)

[3] Carroll, T.L., Pecora, L.M.: Synchronizing chaotic circuits. IEEE Trans. Circ. Syst. 38, 453–456 (1991)

[4] Lorenz, E.N.: Deterministic Nonperiodic Flow. J. Atmosp. Sci. 20(2), 130–141 (1963)

[5] Yang, J.Z., Hu, G., Xiao, J.H.: Chaos synchronization in coupled chaotic oscillators with multiple positive Lyapunov exponents. Physical Review Letters 80(3), 496–499 (1998)

[6] Boccaletti, S., Grebogi, C., Lai, Y.C., Mancini, H., Maza, D.: The control of chos: theory and applications. Phys. Rep. 329, 103–197 (2000)

[7] Boccaletti, S., Kurths, J., Osipov, G., Valladares, D.L., Zhou, C.S.: The synchronization of chaotic systems. Phys. Rep. 366, 1–101 (2002)

[8] Shahverdiev, E.M.: Synchronization in systems with multiple time delays. Physical Review E 70(6), 67202 (2004)

[9] Xu, J.F., Min, L.Q., Chen, G.R.: A chaotic communication scheme based on generalized synchronization and hash functions. Chinese Physics Letters 21, 1445–1448 (2004)

[10] Haefner, J.W.: Modeling Biological Systems, Principles and Applications. Springer (2005)

[11] Tao, C.H., Yang, C.D., Luo, Y., Xiong, H.X., Hu, F.: Speed feedback control of chaotic system. Chaos SolitonFract 23(1), 259–263 (2005)

[12] Yu, Y.: Adaptive synchronization of a unified chaotic system. Chaos Solitons Fractals 36(2), 329–333 (2008)

[13] Wanga, B., Wang, J., Zhong, S.M.: Impulsive Synchronization Control for Chaotic Systems. Procedia Engineering 15, 2721–2726 (2011)

[14] Gonzalo, J., Cicese, B.R., Chen, G., Shieh, L.S.: Fuzzy chaos synchronization via sampled driving signals. Int. J. Bifurcat Chaos 14, 2721–2733 (2004)

[15] Li, C., Liao, X., Wong, K.-W.: Lag synchronization of hyperchaos with application to secure communications. Chaos, Solitons and Fractals 23, 183–193 (2005)

[16] Tavazoei, M.S., Haeri, M.: Determination of active sliding mode controller parameters in synchronizing different chaotic system. Chaos, Solitons Fractals 32, 583–591 (2007)

[17] Liao, T.-L., Tsai, S.-H.: Adaptive synchronization of chaotic systems and its application to secure communications. Chaos, Solitons and Fractals 11(9), 1387–1396 (2000)

[18] Cheng, C.-J.: Robust synchronization of uncertain unified chaotic systems subject to noise and its application to secure communication. Applied Mathematics and Computation 219, 2698–2712 (2012)

[19] Cuomo, K.M., Oppenheim, A.V., Strogatz, S.H.: Synchronization of Lorenzed-based chao-tic circuits with applications to communications. IEEE Trans. Circ. Syst. II 40(10), 626–633 (1993)

[20] Dedieu, H., Kennedy, M.P., Hasler, M.: Chaos shift keying: modulation and demodulation of a chaotic carrier using self-synchronizing Chua's circuits. IEEE Trans. Circ. Syst. II 40, 634–642 (1993)

[21] Yang, T., Chua, L.O.: Secure communication via chaotic parameter modulation. IEEE Trans. Circ. Syst.I 43(9), 817–819 (1996)

[22] Takagi, T., Sugeno, M.: Fuzzy identification of systems and its application to modeling and control. IEEE Trans. Syst., Man Cybern. SCM–15(1), 116–132 (1985)

[23] Lian, K.Y., Chiu, C.S., Chiang, T.S., Liu, P.: Synthesis of fuzzy model-based design to synchronization and secure communication for chaotic systems. IEEE Trans. Syst. Man Cybern. Part B 31, 66–83 (2001)

[24] Kim, J.H., Park, C.W., Kim, E., Park, M.: Adaptive synchronization of T-S fuzzy chaotic systems with unknown parameters. Chaos, Solitons Fractals 24, 1353–1361 (2005)

[25] Hyun, C.-H., Park, C.-W., Kim, J.-H.: Mignon Park Synchronization and secure communication of chaotic systems via robust adaptive high-gain fuzzy observer. Chaos, Solitons and Fractals 40, 2200–2209 (2009)

[26] Dimassi, H., Loria, A., Belghith, S.: A new secured transmission scheme based on chaotic synchronization via smooth adaptive unknown-input observers. Commun Nonlinear Sci. Numer.Simulat. 17, 3727–3739 (2012)

[27] Chen, M., Min, W.: Unknown input observer based chaotic secure communication. Physics Letters A 372, 1595–1600 (2008)

[28] Hyun, C.-H., Park, C.-W., Kim, J.-H.: Mignon Park Synchronization and secure communication of chaotic systems via robust adaptive high-gain fuzzy observer. Chaos, Solitons and Fractals 40, 2200–2209 (2009)

[29] Chadli, M.: Unknown input observer design for fuzzy systems with application to chaotic system reconstruction. Computers and Mathematics with Applications, 1–9 (2013)

[30] Chadli, M., Gahinet, P.: H design with pole placement constraints: an LMI approach. IEEE Transactions on Automatic Control 41(3), 358–367 (1996)

[31] Meng, X., Yu, Y., Wen, G., Chen, R.: Chaos Synchronization of Unified Chaotic System Using Fuzzy Logic Control. In: IEEE International Conference on Fuzzy Systems, FUZZ 2008, pp. 544–547 (2008)

[32] Tanaka, K., Ikeda, T., Wang, H.O.: A Unified Approach to Controlling Chaos viaan LMI-Based Fuzzy Control System Design. IEEE Transactions on Circuits and Systems – I: Fundamental Theory and Applications 45(10), 1021–1040 (1998)

Modeling of EEG Signal with Homeostatic Neural Network

Martin Ruzek

Abstract. Prediction and modeling of signal are tasks that can be done by many methods among which the neural networks have an important place due to the fact that it is data driven method that doesn´t require extensive understanding of the process. This paper presents a new type of neural network that was tested on the modeling of EEG signal. The performance of this network was compared to traditional NN methods. The novelty of this network consists in the fact that each neuron is processing its learning as an independent unit, without any higher process or structure. The learning rule can be simply described as ´improving the significance of the neuron for the rest of the network´. The growth of computational power in recent years opens new possibilities to the use of neural networks in artificial intelligence.

1 Introduction

Artificial neural networks that were inspired by biological neural network, are successfully used for a wide variety of tasks. Its main advantage its that it is data driven method which enables to models signals without the necessity of deep analysis of the data. On the other hand, in many applications of artificial intelligence the neural networks failed the expectations and another techniques are used instead.

This work is aimed to apply the idea of homeostasis to the learning of artificial neural network. Both topics are subject of extensive research, however, they are rarely discussed together. In [1] the previous work on the model of homeostatic neuron is presented. Inspiring ideas can be found in [2], where the role of homeostasis for the immune system is discussed. The hardware realization of neural networks with focus on homeostatic processes is presented in [3]. The interesting question of spatial memory is described on the background of cortical neural cell [4]. In this case, the homeostasis is playing an important role. There are many other applications that can be improved in connection with the principle of

Martin Ruzek
Faculty of Transportation Sciences, CTU in Prague

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 175–180.
DOI: 10.1007/978-3-319-00542-3_18

homeostasis. Majority of them focuses on the transmission of electric signal [5,6], however, the field of use is wider.

2 Homeostatic Neural Network

The proposed neural network is based on the idea of McCulloch-Pitts neuron that is described by the function

$$y = \sum_{i=0}^{n} x_i w_i \tag{1}$$

where f is the transfer function. The proposed neuron calculates the output of its forward phase in the same manner; as the transfer function the following logistic function is used:

$$\mathrm{f(x)} = \frac{2}{1+\mathrm{e}^{-\mathrm{x}}} - 1 \tag{2}$$

The similarity to biological neuron was the basic requirement; for this reason the commonly used back propagation algorithm is not a possible solution. In the back propagation a higher structure (or teacher) is used to train the neuron. instead of this, the homeostatic neuron is using its proper forward connection to improve its function. The ´axon´ in this model has two functions–one is the transmission of the output of the neuron (as in the back propagation), the second is the training of the neuron.

The idea of the training is that the neuron wants to improve its relative importance in the network, is other words, it is trying to maximize the part of its output signal that is accepted by other neurons. This idea is not in contradiction with the physical reality of the biological neuron, as the ´information´ transmitted by the axon has the form of energy (and is inseparable from the energy).

The process of learning can be described by the following procedure: first, the neuron compute its output with its initial random weight. The neuron in the higher layer set their weights according to their level of contentment with the reference neuron. In the next step the neuron changes its weights according to some algorithm. Several possibilities of this algorithm are describer later in this paper, however, there are many other possibilities; all the possible algorithms are using at least 2 values of the level of acceptance. This implies that the neuron must be equipped with a memory.

It is also possible to execute the described algorithm in batch mode. This algorithm can be simply realized in one–layer network, however, in multi layer perceptron the problem of delay arises. The delay will cause instabilities that unable direct use of this method. This work is focused on the learning of a neural network with one hidden layer.

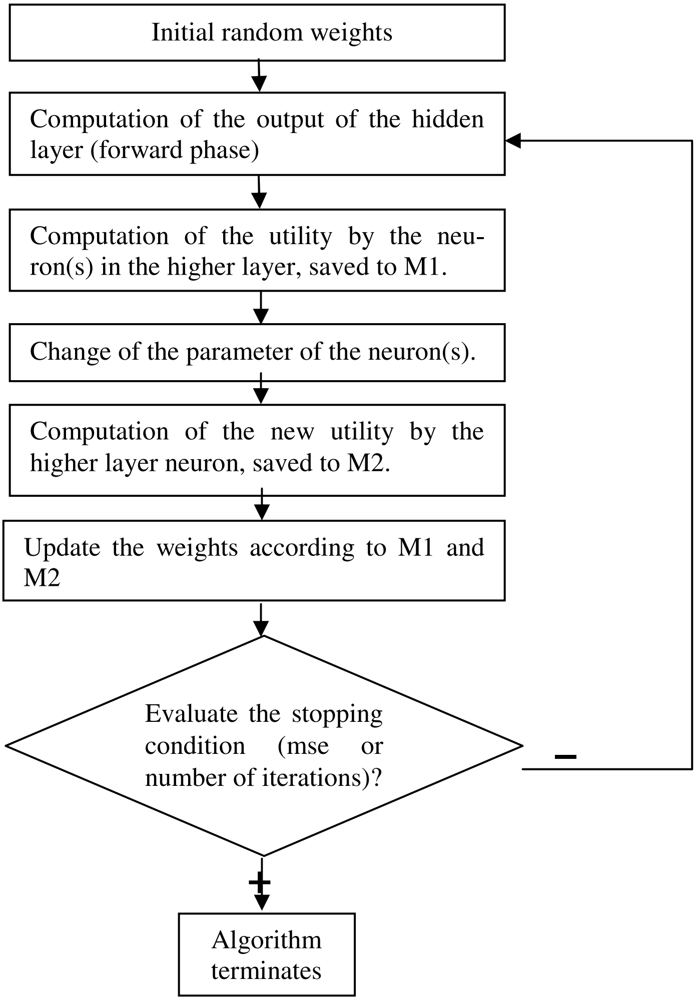

Initial random weights
Computation of the output of the hidden layer (forward phase)
Computation of the utility by the neuron(s) in the higher layer, saved to M1.
Change of the parameter of the neuron(s).
Computation of the new utility by the higher layer neuron, saved to M2.
Update the weights according to M1 and M2
Evaluate the stopping condition (mse or number of iterations)?
–
+
Algorithm terminates

Fig. 1 Process of learning of the homeostatic neuron

In detail, the process is described by the following algorithm:

1. neuron sets its initial input weights randomly
2. neuron computes its output
3. the output neuron computes the utility of the reference neuron and sets its input to equivalent level. There are several methods how to compute the utility, some of them are described in eq (3)-(6).
4. neuron changes one (or more) of its input weights
5. neuron again performs the forward phase with the same data
6. the output neuron compute the utility (the same as step 3)

7. the reference neuron evaluates the change in the step 4–if it improves the utility, it will keep it, otherwise will do a different change
8. neuron repeats steps 2 to 7 with all the connections and all the inputs

Several methods are used to calculate the importance of the neuron. The basic difference among these methods is the number of output neurons for which the reference neuron is ´working´. The first extreme is a neuron that is ´working´ for all output neurons. This neuron is improving all the output weights without taking any particular output neuron into consideration. The other extreme is a neuron that ´works´ only for one neuron in the higher layer, or that is optimizing its function in order to improve its utility for only one neuron. There are many compromise solutions between these cases.

3 Searching the Optimum Based on the Sum of the Output Weights

This idea corresponds to the first case (neuron is trying to be useful for all neurons and therefore is maximizing all the output weights). In other words the neuron is finding such a weight vector Λ={w_1, w_2, ... w_n} for which the sum of the absolute values of the output weights is maximal. This idea corresponds firmly to the biological model because there is only one axon and therefore the neuron can only be aware of the total amount of the signal that is accepted by others, not of the particular weights. In the artificial neuron we also count with negative weights; because of that, the neuron will sum the absolute or square values. The utility *q* is:

$$q = \sum_{j=1}^{n} \left| w_j^o \right| \tag{3}$$

respectively:

$$q = \sum_{j=1}^{n} \left(w_j^o \right)^2 \tag{4}$$

Eq. (5) puts stress on great values and reduces the importance of the small ones. This can be advantage for the learning but does nor correspond to the reality of biological neuron.

4 Searching the One Neuron Maximum–The Second Extreme

The other type of training is based on the presumption that the neuron is increasing its importance to only one neuron in the higher layer, therefore it maximizes the function:

$$q = \max \left| w_j^o \right|; \; j \in \{0,1,..n\} \tag{5}$$

If max $|w_o|$=1, no further improvement is possible, and the training stops. In real situation we expect a networks with many neurons, so the learning will stop soon. This is not a desired behavior, therefore in that case there should be an additional condition that ensures the continuation of the training. The solution of this problem is to use a compromise solution that takes into consideration more than one output neuron but not all of them. This can be done by optimization of some given number of maximal output weights:

$$u=\max(|w_0|)+\max(w_0-\max(w_0))+\ldots;\ w_0=\{w_0,w_1,\ldots,w_{n-1}\} \qquad (6)$$

5 Modeling of EEG Signal

The EEG (electroencephalography) is the recording of electrical activity of the brain. The EEG is commonly used in medical practice to detect epileptic seizures, movement disorders, migraine and other problems. There are many other fields where the EEG possibly can be used; one example is the detection of the level of vigilance of the human operator. The signal that was used in the test was achieved during the tests of driver´s vigilance. The selected signal is shown on the graph (1). The dataset was normalized into <-1;1> interval and interval of 150 values was selected.

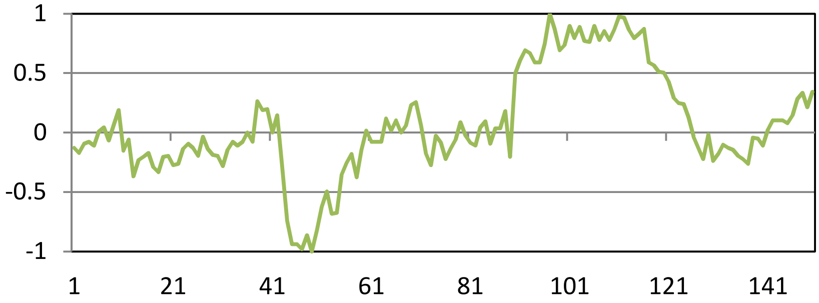


Fig. 2 The EEG signal

The signal was modeled by two different methods. First was the back propagation network. The network had 4 input nodes, 4 hidden nodes and 1 output node, the momentum was set to 0.9 and training rate to 0.5. The result was compared to prediction with the homeostatic networks that were based on eq. (4), (5) and (6) respectively. The desired sum of square error was 10^{-3}.

5 Results

The classical back propagation network converged to desired *sse* after 9718 iterations. The homeostatic neural network that was based on the idea of ‘every neuron works for everybody' didn't converge, which is not surprising if we take into consideration the number of input data and relatively simple definition of the neuron. When the dataset was reduced to 20 samples, the network started to converge; the desired sse (10^{-3}) was achieved after 2509 iterations.

Better results were achieved with neural networks based on eq. (5) or (6). Here, the convergence is still slower than in the case of back propagation (21 512, resp. 26 012 iterations), greater oscillations of sse that occur during the learning process may have an positive effect on avoiding the local minima problem.

6 Conclusion

The test on the EEG signal confirmed the presumption that the proposed 'homeostatic' neural network has lower rate of convergence than the back propagation learning. On the other hand, the proposed network has different way of learning which can avoid several problems of neural networks, such as falling into local minima or over fitting. Probably, learning according to rule (3) or (5) are not ideal, the best solution is to use some combined criterion. Due to the complex nature of neural network, only extensive tests on different datasets can reveal the strengths and weaknesses.

References

[1] Růžek, M.: Model of biological ANN based on homeostatic neurons. In: Recent Researches in Neural Networks, Fuzzy Systems, Evolutionary Computing & Automation, pp. 66–69. WSEAS, Athens (2011) ISBN 978-960-474-292-9

[2] Downing, J., Miryan, J.: Neural immunoregulation: emerging rules for nerves in immune homeostasis and disease. Immunology Today 21(6) (June 2000)

[3] Bartolozzi, C., Giacomo, I.: Global scaling of synaptic efficacy: Homeostasis in silicon synapses. Neurocomputing 72, 726–731 (2009)

[4] Miyashita, S., Tabuchi, Y., Tanaka, S.: Cortico-thalamocortical operations of multitarger spatial working memory. Neurocomputing 52(4), 327–333 (2003)

[5] Pedrycz, W., Vukovich, G.: Granular neural networks. Neurocomputing 36, 205–224 (2001)

[6] Takizawa, Y., Fukasawa, A.: Neural Network Modeling based on Biological and Electrical Behaviors. In: Proceedings of the 9th WSEAS Conference on Applications of Electrical Engineering, Penang, Malaysia, pp. 221–224 (March 2010)

Model Identification from Incomplete Data Set Describing State Variable Subset Only – The Problem of Optimizing and Predicting Heuristic Incorporation into Evolutionary System

Tomas Brandejsky

Abstract. Presented paper describes the application of evolutionary system GPA-ES in difficult task of chaotic system symbolic regression from incomplete training data set describing only some of model variables. The algorithm uses many heuristics which are described below and which will be subject of future development. The first test of algorithm was applying the Lorenz attractor system data, where only the original system x and y variable data were used and z variable data were estimated.

1 Incomplete Variable Data Subset

In many real world situations, some of the system variables are not observable, but the system model needs to be identified. In the past few years, symbolic regression based on application of evolution techniques to these tasks has established as the one at least equal to standard identification approaches. Standard symbolic regression is applied in situations, when set of input, output and state variables is known and observable. Then, for each state and output variable complete training set is accessible and fitness function might be formed.

Different situation occurs when one, or more state or output variables are not observable. Because it is known that evolutionary techniques give reasonable results under conditions of uncertainty (and lack of information is the source of uncertainty), this paper presents the attempt to solve the system symbolic regression from incomplete training data set represented by missing/inaccessible variable magnitudes vector.

The used evolutionary system is based on GPA-ES evolutionary system described latter with many extensions given by the need to estimate fitness of unknown variables – variables which cannot be compared with training data set.

Tomas Brandejsky
Faculty of Transportation Sciences, CTU in Prague

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 181–189.
DOI: 10.1007/978-3-319-00542-3_19

This main problem tend to application of many next solutions complicating the algorithm, like the need of parallel evolution of all variables together, increase of optimization problem complexity and concluding decrease of the algorithm efficiency, as it will be also discussed latter.

Optimization problem is mentioned herein because Genetic Programming Algorithm technique was formed by Koza [1] as specific optimization problem. This optimization viewpoint is even forced e.g. in symbolic regression tasks, when not only structures, but also parameters need to be identified.

2 GPA-ES Algorithm

The used GPA-ES system combines Genetic Programming Algorithm with Evolutionary Strategy. Evolutionary Strategy is used in each cycle of GPA work for each individual of its population for optimization of parameters of model developed by this GPA to optimize developed system or model parameters. Only this computationally expensive way gives certainty that GPA algorithm compares models with perfectly fitted parameters, it means, it compares them without noise caused by random influences of ill-identified parameters. Other words, hybrid GPA-ES algorithms are more efficient in symbolic regression of complex systems than original GPAs not looking that there is large computational effort consumed by created structures parameter estimation. Similar ideas tend to development different combinations of algorithms like Analytic Programming [2] combining Differential Evolution (AP uses standard genetic programming and then interprets resulting string on the place of tree developments applied in GPAs) to structure development with Evolutionary Algorithm in the form of Self Organizing Migrating Algorithm (SOMA).

The GPA-ES algorithm is suitable to Precise Symbolic Regression (PSR). When the sizes of GPA and ES populations and numbers of their iterations are small, the founded solution is approximate – imprecise. Above some limit given by complexity of the solved problem, the structure and parameters of regressed solution equal to original model are reached. Is such situation it is possible to speak about PSR.

In the previous works as [3 and 4], there is described PSR of differential equations describing deterministic chaos systems like Lorenz attractor, Roesler attractor, Rabinovich-Fabrikant equations and Van der Pol oscillator. These works also present the structure of GPA-ES algorithm, its parameters, computational complexity and correspondence of this complexity relation with practical experiments.

3 Model Identification from Incomplete Data Set Problem

Symbolic regression of the model might be illustrated e.g. on Lorenz attractor system. This system is described by three variables x, y and z and proper equations (1):

$$\begin{aligned} x'[t] &= \sigma\,(y[t] - x[t]), \\ y'[t] &= x[t]\,(\rho - z[t]) - y[t], \\ z'[t] &= x[t]\,y[t] - \beta\,z[t] \end{aligned} \tag{1}$$

When some variable(s) are immeasurable, e.g. variable z, the data set used for symbolic regression (reconstruction) of Lorenz system is incomplete and the task is much difficult than the task described in the work [4].

When the training data for variable z are missing, it is still possible to reconstruct it from training data for variable y, which is dependent on variable z. Unfortunately, the symbolic regression of the model does not start with perfectly fitted equations describing variables x and y. This fact increases risk of the trapping of solution in the local minima. Also, when the symbolic regression of variable y requires n_y cycles of the GPA-ES algorithm and the identification of z data n_z steps (in the case of complete training data set availability), the number of evolutionary data steps for simultaneous identification of equations describing both variables y and z from single data set for variable y is not the number n_y+n_z, but it is possible to estimate that this number will reach magnitude at least n_y*n_z! It might be even bigger for decrease of GPA and ES algorithms efficiency for much complex tasks observed by many researchers [5]. The ability of used algorithm to reach PSR gives some evidence of its ability for this task of symbolic regression from incomplete data set.

4 Algorithm Extensions

Standard GPA-ES algorithm uses separate identification of variables, which is independent on identification of the other variables. This simplification is enabled by complete training data set only. In the other cases there are two groups of variables – separable and inseparable ones. Separable variables are represented as the function of variables present in the training data set. Also magnitudes of the separable variable must be present in the training data set. Inseparable variables are the others. Unfortunately, on the beginning of symbolic regression, each variable must be reasoned as the function of the whole variable set. There is no additional information. Thus, the simultaneous evolution of all variables must be used. This tends to additional dimension to many system variables to allow parallel evolution for all variables together.

In addition to this modification of the GPA-ES algorithm, the magnitudes of unmeasured variables (e.g. variables which are not contained in the training data set) have to be estimated. It is the serious problem, because the function describing their behavior is subject of the symbolic regression. Presented algorithm uses actual models developed by the algorithm and their fitness is the sum of fitness functions of the rest variables with available training data sets. The idea is, that only on the base of prediction produced by well estimated function, the other regressed functions might be able to reach optimal model structure and minimal fitness function error. The problem is increase of complexity of the function measured by given training data column.

In the standard symbolic regression model, training data represent function vector (2):

$$v = \{f_i(t)\}_i \tag{2}$$

The regressed model then represents function vector

$$v' = \{f_i'(t)\}_i \tag{3}$$

In the paper [6], the significance of reliable distinguishing of "good" and "poor" ideas (genes) is discussed. Because the original and especially the estimated functions are nonlinear, the metric space is complicated. Especially in the case of chaotic systems, the local similarity is not significant and it is need to measure distance of original and regressed model functions (4).

$$|v - v'| = \{f_i(t) - f_i'(t)\}_i \tag{4}$$

The paper [6] discuses three basic methods of function distance. They are:

4.1 Algorithm V1

Algorithm V1 is based on computation of estimated model output in each step of time series when original data are used on the place of initial state. Main advantage of this approach is simplicity of its implementation. On the contrary, well known behavior of chaotic systems is divergence of trajectories, when initial state moves. Thus, this approach, which is frequently used in time series modeling, is less sensitive, than the algorithm V2. The situation is illustrated by Fig 1.

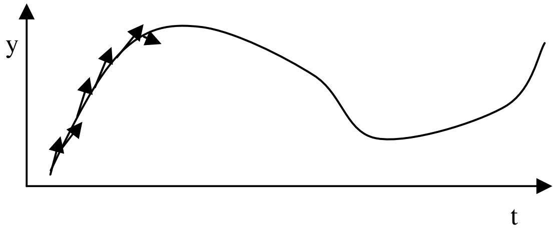


Fig. 1 V1 method of state prediction – in each time step novel prediction is computed only for next time step

4.2 Algorithm V2

Continuous computation of whole time series of chosen variable from single starting point is the pedestal of this strategy. Because there are closed information loops among variables in chaotic multidimensional systems, also this case is not precise. It is possible to tell, that variables in chaotic systems are inseparable, that it is not possible to compute them independently. On the opposite side, it is possible to expect that the resulting distances will be more significant than in the previous case. This situation is illustrated by Fig 2.

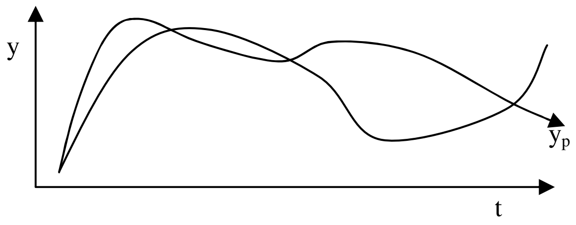


Fig. 2 V2 algorithm of state prediction – novel state trajectory y_p from the given starting point is computed

4.3 Algorithm V3

It is also possible to form the algorithm V3 which consists in simultaneous prediction of movement in all dimensions (variables) simultaneously from given starting point. The next magnitudes of all variables are estimated by regressed model, not from the training data set. The main problem of this strategy is the inability to simply separate influence of error of one dimension to another from causes analogical to effect causing limitations of V2 algorithm. Thus, from practical viewpoint, the implementation this strategy is more difficult than in the case of previous strategies and it was used only in complicated situation described in the following chapter.

In many practical experiments with symbolic regression of systems, the algorithm V1 gives better results than algorithm V2. It is caused by sensitivity of V2 algorithm to influences such are the offset or monotonously increasing derivation, which are not significant for final solution. Problems of algorithm V3 extends problems of V2, but in the case of model identification from incomplete data set it is possibly the only applicable.

4.4 GPA-ES Modification for Symbolic Regression of Variables without Training Data Set

The estimation quality of functions describing of directly immeasurable variables requires the following significant modifications of original GPA-ES algorithm. Especially, it was needed to add common initialization for all variables together, whilst the standard GPA-ES solves each variable independently. Individuals are now represented by vector of genes, so the usual relation "one gene describes one individual" is not valid. Thus, the GPA-ES output needs to be modified analogically. Evolution in this modification ends when all variables satisfies final condition. When the best gene of a well described variable is changed, it is need to recalculate fitness measures of all genes of all variables without fitness data. Change of the best individual of well described variable usually changes order of variables without fitness data, because their fitness is estimated using it.

5 Used Heuristics

Many above mentioned modifications – extensions of the original algorithm might be solved many ways. It decreases evidence that chosen set of techniques is optimal. On the other, there is open space to optimize algorithm and increase its efficiency in the future. There is need to provide careful testing of the algorithm too, because the size (number of lines of C++ code) of the key component of the algorithm was increased about fifty percent.

Especially, it is not sure if it is better to modify all, many or only one variable regression function in a single evolution step of the algorithm. Simultaneous modification of many variables decreases risk of trapping in the local minima, but decreases efficiency of the algorithm too.

Another significant question is how to measure quality of variables without training data set. Is it better to use sum of errors of the best individuals of the rest variables (as it is used now), sum of errors of the related individuals of the rest variables, to find the optimal set of individuals for whole variable set (this method tends to combinatorial explosion of reasoned combinations), etc.

The estimation of fitness error of variable without training data set on the base of their prediction by the algorithm V3 also represents hardly verifiable heuristic. There vas observed many situations, when this heuristics gives poor results and its replacement will be subject of future research.

These optimizing and predicting techniques were used in the presented algorithm, but their use must be accepted as first draft of feasibility study of future algorithm, not the final version.

6 The First Experiments

As training data set for the first experiments, the Lorenz attractor described by equation (1) was used. These experiments conclude the required number of iterations estimation outlined in the chapter 3. The rapid decrease of the algorithm efficiency is caused not only by the complexity of the task, but also by the well known decrease of the efficiency of the evolutionary algorithms with the problem complexity.

While the GPA-ES algorithm with size of GPA population equal to 40 and sizes of ES populations are equal to 400, the average number of iterations for precise symbolic regression (obtaining of original quality model) is equal to 436. On the other hand, when the z variable data are unavailable, the poor quality model is obtained after 16000 iterations (for all 3 variables together), GPA population consists of 1000 individuals and ES populations of 400 individuals.

Whilst in any population size the algorithm is able to find perfect model of the variable x, because it depends on the variables x and y.

```
    0 : (16)*((var(1))-(var(0)))   fitness:=3.54198e-15
    The models of y and z are
    10
    y:((((74.0568)-(var(1)))-((-64.0508)+(-91.1846)))-
((var(0))*(var(0))))*((var(0))*(0.0442184))   fitness:=1019.55
    z : (((0.751822)*(var(1)))*((0.401957)*(var(1))))*((0.879567)*(var(0)))
      fitness:=3522.96
    40
    y:(((-1.03459)+((0.286)*(var(1))))-(((-3.22615)*(var(0)))-(-
21.8724)))+(((var(0))-((var(1))-(var(0))))*(((-1.03459)*(var(0)))+(var(0))))
    fitness:=225.125
    z : (0.0353779)*(0.136017)   fitness:=3663.04

    100
    y:((((14.5111)*(var(1)))-(1.03206))+(((var(0))*((-
0.0416169)*(((var(0))*(var(0)))+((var(1))-(-1.54639)))))-((2.16954)*(var(0)))))-
((var(0))+((((14.5111)-(1.03206))-(-0.0416169))-((var(1))*(-1.54639))))
    fitness:=331.847
    z : (4.19226)+(((-0.831369)*(var(0)))*((-3.01553)*(var(0))))   fitness:=3263.96
    400
    y:(((var(0))-((var(0))*(var(1))))-(((var(1))-(var(0)))*(-7.99824)))-
(((((var(0))+(4.64307))*((var(1))-(0.0641911)))*(-0.914411))-(((-
7.99824)+((4.64307)*(var(0))))-((0.0641911)*((var(0))*((-
0.914411)+(((0.667165)*(var(0)))*(var(0))))))))   fitness:=324.686
    z : (var(0))*(var(1))   fitness:=1579.63
    1000
    y:(((var(2))-(-0.971768))-((27.5105)+((15.7336)+(2.44241))))*((var(0))*(-
0.971768))   fitness:=2.24364
    z:(((-4.17068)+((-8.59701)*(var(0))))*((-0.118576)*(var(1))))-
((4.14134)*(var(2)))   fitness:=3.29511
```

It is also need to remark that the fitness of the best fitness of the given variable is calculated on the base of related (but not the best) independent variables regressions. Equation for y variable might contain elimination of error inherited in related regression of z and vice versa. Thus presented equations for y and z variables are not related together and does not form consistent system, they must be accepted separately. Discoverer models are presented by the following figures.

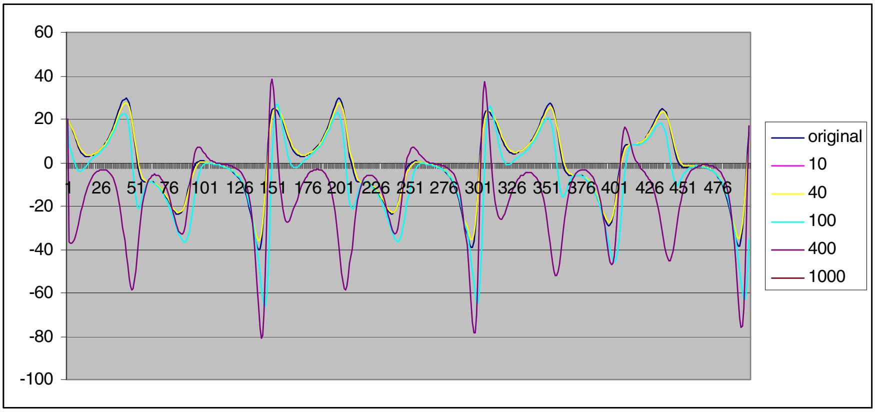


Fig. 3 The best approximations of y variable discovered by GPA and ES populations of sizes 10, 40, 100, 400 and 1000 individuals

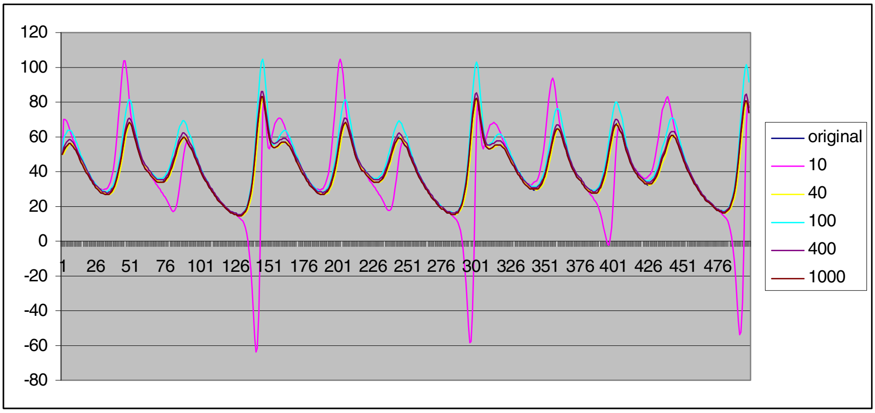


Fig. 4 The best approximations of z variable discovered by GPA and ES populations of sizes 10, 40, 100, 400 and 1000 individuals

7 Conclusions

The applicability of the presented method is limited due to enormous computational complexity, but the algorithm contains large degree of freedom from the possible future development viewpoint. Now, the speed of evolution is enormously decreased in comparison to situation when all variables are included in the training data set.

In the nearest future it is probably impossible to expect the application this technique in such complex tasks as e.g. in reconstruction of information encrypted via deterministic chaos, but the possibility to develop models of only partially observable systems including chaotic ones is the extremely significant task which call for future research.

Acknowledgements. The work is supported by research project of MŠMT ČR No 6840770043 “Improvement of methods of design and employment of transportation networks from optimization viewpoint”.

References

[1] Koza, J.R.: Genetic Programming: On the Programming Computers by Means of Natural Selection. The MIT Press, Boston (1992)

[2] Zelinka, I., Davendra, D., Senkerik, R., Jasek, R., Oplatkova, Z.: Analytical Programming - a Novel Approach for Evolutionary Synthesis of Symbolic Structures. In: Eisuke, K. (ed.) Evolutionary Algorithms. InTech (2011)

[3] Brandejsky, T.: Multi-layered evolutionary system suitable to symbolic model regression. In: Recent Researches in Applied Informatics, vol. 1, pp. 222–225. WSEAS Press, Athens (2011)

[4] Brandejsky, T.: Nonlinear system identification by GPA-ES. In: Petráš, I., Podlubny, I., Kostúr, K., Kačúr, J., Mojžišová, A. (eds.) Proceedings of the 2012 13th International Carpathian Control Conference (ICCC), CD, IEEE Catalog Number: CFP1242L-CDR (2012) ISBN: 978-1-4577-1866-3

[5] Langdon, W.B., Poli, R.: Foundations of Genetic Programming. Springer, Heidelberg (1998)

[6] Brandejsky, T.: Symbolic regression of deterministic chaos. In: Proceedings of 17th International Conference on Soft Computing (MENDEL 2011), pp. 90–93 (2011)

Supervised and Reinforcement Learning in Neural Network Based Approach to the Battleship Game Strategy

Ladislav Clementis

Abstract. In our study the Battleship game we concern as an example of a simple pattern matching problem in correspondence with the Partially observable Markov decision process. We provide comparison of supervised and reinforcement learning paradigms used as neural network learning mechanisms applied by solving the Battleship game. We examine convergence of the neural network adaptation process by using these techniques. While concerning our pattern matching problem of the Battleship game solution by the neural network the reinforcement learning technique is not as straightforward as the supervised learning. On the other hand the neural network adaptation by the supervised learning mechanism has a faster convergence in our case. We use the Battleship game probability model to determine next position in an environment to be shot at with the highest probability of resulting into a successful hit attempt.

1 Introduction

In various topics like modeling, prediction, artificial intelligence, optimization [18] and many others, artificial neural networks and machine learning paradigms have been credited since developed. These paradigms if well adapted to a given problem domain are capable of solving various complex tasks [5].

Basic neural networks can use various methods of learning to adapt to a given problem. This learning procedure is not trivial and has to be discussed.

We use the simplified version of the Battleship game environment to study various learning processes like the supervised learning and reinforcement learning in this contribution.

Ladislav Clementis
Institute of Applied Informatics, Faculty of Informatics and Information Technologies,
Slovak University of Technology, Ilkovičova, 842 16 Bratislava, Slovakia
e-mail: clementis@fiit.stuba.sk

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 191–200.
DOI: 10.1007/978-3-319-00542-3_20

2 Reinforcement Learning

In general, the reinforcement learning [6, 15] is a machine learning paradigm which uses interaction with an environment to maximize a cumulative reward (or minimize a long-term cost). In a current state of an environment we can perform multiple actions. Each possible action a performed changes this current state of an environment s_1 into another state s_2 replacing state s_1 as shown in equation 1. Our goal is to determine the optimal action for each state of an environment.

$$s_2 \leftarrow (s_1, a) \tag{1}$$

Each pair of state and action (s_1, a) does have reward value r which is used as a response from an environment. The closer to solution we get the higher reward from an environment is gained. Our goal is to choose an action which ensures the highest possible reward from an environment according to a current state. Thus by performing actions and changing states of an environment we get to the solution by collecting the highest rewards.

2.1 Q-Learning Algorithm

The one of commonly used reinforcement learning algorithms which propagates reward prediction values across a problem space is the Q-learning algorithm [16]. Propagated reward prediction values we call Q-values. In each iteration of the Q-learning algorithm (also called an episode), Q-values for concerned states are updated by equation 2.

$$Q(s,a) \leftarrow Q(s,a) + \beta(r + \gamma \max_{a'} Q(s',a') - Q(s,a)) \tag{2}$$

Q-values are backpropagated from a solution until an absorbing state (which is a final one) is reached. The reward value gained from an environment by performing an action a (which changes the current state s_1 into another state s_2) is the Q-value set to the state s_2 by the Q-learning algorithm.

The Q-learning algorithm uses the learning rate β and the discount factor γ to parameterize reward propagation across a state space.

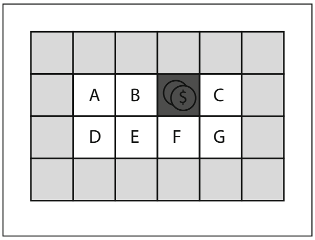


Fig. 1 The simple environment example showing positions in the environment. Gray boxes are showing problem space boundaries. The position marked with coins and the dollar sign symbolizes the treasure position.

We provide the simple environment example shown in figure 1. This environment consists of eight positions $(A, B, \$, C, D, E, F, G)$. One of these positions contains treasure which is our goal. An imaginary agent holds one of these positions in the environment. The position he holds is the current state of the environment. The agent can move up (↑), right (→), down (↓) or left (←) within boundaries. A single move of agent in an allowed direction is an action which changes the position of agent i.e. the state of the environment.

We have set the initial treasure reward value to 1000, the learning rate $\beta = 1$ and the discount factor $\gamma = 0.9$. The Q-values and the position-action reward prediction values are shown in table 1.

Table 1 The position Q-values and reward prediction values shown in table sorted by positions

Current Position	Predicted Q-value	move ↑	move →	move ↓	move ←
A	810	–	900	729	–
B	900	–	1000	810	810
\$	1000	–	900	900	900
C	900	–	–	810	1000
D	729	810	810	–	–
E	810	900	900	–	729
F	900	1000	810	–	810
G	810	900	–	–	900

The agent chooses a movement direction. His decision is based on the Q-values of the nearest neighboring positions, specifically on the highest state-action reward prediction value in his neighborhood.

Note that there can be multiple optimal paths to the treasure-containing goal position. These optimal paths have same length and therefore are equal. For example, if the agent is standing on E position he can choose the path either through B or F position. The both B and F positions have the reward prediction value of 900 and therefore there are two optimal path options - $(E, B, \$)$ and $(E, F, \$)$.

In general, if a current state of an environment is fully observable we are dealing with the Markov decision process (MDP). On the other hand if a current state of an environment is not fully observable we are dealing with the Partially observable Markov decision process (POMDP). For example in a Battleship game instance, current ship placements are unknown until completely revealed. In this case there are multiple states which are corresponding with our (observable) state of an environment and a heuristic approach like a probability model or another should be taken into account.

3 Neural Network Learning Paradigms

Artificial neural network [9, 14, 17] as a mathematical sub-symbolic system inspired by biological neural networks is widely used to address various classification problems [10]. There are many neural network types, for example the most simple feedforward neural networks, recurrent neural networks like Hopfield networks or Boltzmann machines, dynamic neural networks, neuro-fuzzy networks, stochastic neural networks and many others.

There are different approaches of learning paradigms used as a neural network training methods. The most used categorization is:

1. Supervised learning
2. Unsupervised learning
3. Reinforcement learning

In our contribution we are concerned about supervised and reinforcement learning paradigms for our purposes.

3.1 Neural Network Adapted by Supervised Learning

Supervised learning is the machine learning task which uses labeled training data as a set of examples (X,y). This labeled training data consist of inputs as vectors X for a trained system and a desired system output y for each input.

Supervised learning algorithm uses training data to produce an inferred function $y = f(X)$ which can be a classifier or a regression function. Classifier is produced if outputs are discrete and regression function is produced if outputs are continuous. The main idea of the supervised learning algorithm is to generalize similar inputs for a given output and thus inferred function can be produced.

While dealing with the supervised learning method for training a neural network, usually the mean-squared error between current and desired output is used as the cost we want to minimize. One of appropriate algorithms for minimizing this cost is the gradient descent algorithm.

3.2 Neural Network Adapted by Unsupervised Learning

While using the unsupervised learning a cost function is defined to evaluate neural network input-output pairs. Usually the cost function is based on analysis of a given problem which neural network should deal with. If a neural network output y is considered as a function value based on input vector X, the cost function $C(X,y) = C(X,f(X))$ is used to evaluate neural network outputs for given inputs. This approach can be used if we have an additional information about given problem. The cost function can be defined accordingly to this additional information, e.g. mathematical or statistical properties like problem probability model, distribution of desired neural network inferred function etc.

3.3 Neural Network Adapted by Reinforcement Learning

As mentioned in section 2 the reinforcement learning is paradigm which uses an interaction with an environment to discover an unknown policy. A response from an environment is used to estimate this unknown policy.

In an artificial neural network learning process an inner representation of neuron functions and parameters are adjusted. While using the reinforcement learning, this adjustment is driven by environment response to approximate desired response recall. An adapted neural network should give a high-rewarded output to a current state of an environment.

In comparison to the supervised learning, no correct outputs of neural network are given. It is up to neural network to adapt and learn from an experience. Dynamic programming and other approaches are used for this purpose. Thus an inferred function $y = f(X)$ can be estimated.

4 The Battleship Game Strategy Based on Probability Model

As an example we provide the instance of simplified version of the Battleship game [2]. Probabilistic approaches are widely used [4] while dealing with decision-making problems [12]. The Battleship game in general we can divide into these main tasks:

1. Initial ship deployment in player battlefield
2. Attempting to hit an enemy ship
3. Completely sinking damaged enemy ship

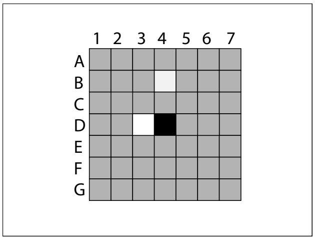
The first task of ship deployment is initial. The second and third tasks are repeated until a fleet of a player or a fleet of an enemy is completely destroyed. We are taking the third task of the Battleship game from the point of a single player into account as an optimization problem. Our goal is to fully discover the enemy ship placement in a battlefield with minimum shots fired. A ship may be considered as a (visual) pattern we want to match.

We use the battlefield of size of $7 \times 7 = 49$ positions. This battlefield is our environment. The battlefield contains two ships of size of 1×3. These two ships are placed in the battlefield in the initial deployment stage. After initialization (from our point of view) the enemy battlefield contains only unrevealed positions because hit attempts have not been performed yet.

As the main part of the game starts, several attempts to hit an enemy ship are made. Each shot reveals us one position of an enemy battlefield as a response from the environment. After a few unsuccessful hit attempts performed a successful shot occurs. The example how the enemy battlefield from our point of view looks is shown in figure 2.

The updated example in figure 2 shows that revealed position $B4$ and $D3$ do not contain a part of a ship. Shot at position $D4$ was successful and has revealed a part of a ship. Our goal is to maximize the probability of the next shot to be successful.

Fig. 2 Our limited view of the example environment. Gray positions are still unrevealed, white positions are those where unsuccessful hit attempts were performed and the $D4$ identifies single successful hit attempt position. [2]

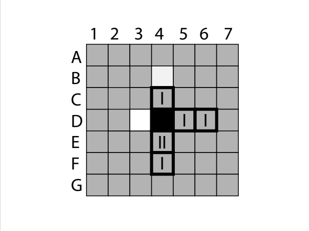


There are three possible placements that the damaged ship can occupy:

1. $(C4,D4,E4)$ (vertical placement)
2. $(D4,E4,F4)$ (vertical placement)
3. $(D4,D5,D6)$ (horizontal placement)

All three ship placements are considered to be equally possible.

Fig. 3 The example environment fom figure 2 enriched with flags. A single flag means that position is covered with a single ship placement option. Position $E4$ contains two flags because it is covered with two placement options. [2]



As shown in figure 3, positions $C4$, $D5$, $D6$ and $F4$ are covered with one possible ship placement. The $E4$ position is covered with two possible ship placements. Therefore the $E4$ position have the highest probability to contain another part of the ship pattern and it is reasonable to shoot at this position.

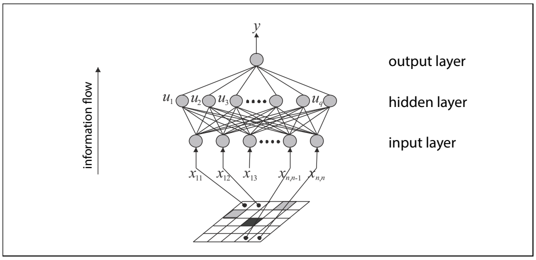
Because the current state of the environment is not fully visible (there are uncovered positions) we consider this as the Partially observable Markov decision process. Shooting on unrevealed position gives us a response from the environment. This response is result of shooting - hit or miss. Acting with correspondence to this probability model we consider as the optimal strategy to the Battleship game example shown in figure 2 and 3.

5 Neural Network Adaptation to the Battleship Game Strategy

For application to the Battleship game example shown in figure 2 and 3 we use the three layer feedforward neural network shown in figure 4.

We use transformed current state of the environment as the two-dimensional neural network input vector X defined by equation 3.

Fig. 4 Diagram showing the abstract overview of used three-layer feedforward neural network



$$X = (x_{11}, x_{12}, ..., x_{ij}, ..., x_{nn}) \in \{-1, 0, 1\}^{n \times n} \tag{3}$$

The neural network shown in figure 4 contains the hidden layer defined by equation 4.

$$U = (u_1, u_2, ..., u_\alpha, ..., u_q) \in [0, 1]^q \tag{4}$$

The neural network output y defined by equation 5 returns the real number variable from the range $[0, 1]$.

$$y = (y_1) \in [0, 1]^1 \tag{5}$$

The output real variable is mapped back into the two dimensional environment consisting of 7×7 positions. The output variable value identifies position to be shot at at this state of the environment. The neural network we used is able to learn the mapping between the input and output layer by using the hidden layer.

We compare the supervised and the reinforcement learning approach to the neural network adaptation to the Battleship game example shown in figure 2 and 3. If we would define the cost function to evaluate input-output pairs we could also use the unsupervised learning approach mentioned in subsection 3.2. This cost function is difficult to define explicitly but the evaluation based on the probability model is easy to implement as an algorithm.

5.1 Neural Network Adaptation to the Battleship Game Strategy Using Supervised Learning

As we mentioned in subsection 3.1 the supervised learning uses the training set of input-output pairs for the neural network learning process. We used 50 pre-defined input vectors X and corresponding outputs y. The outputs were defined in correspondence with the Battleship game probability model as positions with the highest hit probabilities as described in section 4.

We considered the neural network ”adapted” when it was able to response correctly to an input with 90% or a higher precision.

1200 iterations of the learning process were sufficient for the neural network to adapt to correct output position deciding. Each input-output pair was used 24 times in a shuffled order.

5.2 Neural Network Adaptation to the Battleship Game Strategy Using Reinforcement Learning

For the neural network adaptation process by the reinforcement learning mentioned in subsection 3.3 the environment reinforcement feedback was used. The environment response was information about a hit attempt success. We used the same 50 input vectors as we used in the supervised learning adaptation process.

20000 reinforcement learning iterations were needed for the neural network to adapt to solve the position decision problem. In comparison, while using the supervised learning it took 1200 iterations to sufficiently adapt the neural network what is only 6% of 20000 reinforcement learning iterations.

Table 2 Three simulation result examples of each supervised and reinforcement learning used for neural network adaptation. Results are shown by a number of learning iterations and adaption rate which is neural network success rate (decision making precision)

Method used	Iterations	Adaptation rate
Supervised learning (*example* 1)	1134	91.7%
Supervised learning (*example* 2)	1249	92.1%
Supervised learning (*example* 3)	1178	90.4%
Reinforcement learning (*example* 1)	18394	90.8%
Reinforcement learning (*example* 2)	17360	91.1%
Reinforcement learning (*example* 3)	18297	90.5%
Supervised learning (average)	1187	91.4%
Reinforcement learning (average)	18017	90.8%

We have studied the supervised and the reinforcement learning processes on this example closely. The rapidly slower neural network adaptation by using the reinforcement learning was caused by:

1. The fact that the probability model of the Battleship game does not ensure a successful hit attempt but effectively rises probability of successful attempt.
2. The reinforcement learning itself is the learning method based on successful and mistaken actions performed in an environment. The optimal action is not known until executed because learning is performed by the response from an environment and no additional information is included.

As generally known, the reinforcement learning approach to a neural network adaptation is more close to a human-like learning process. A human being is usualy learning a game strategy by repeated playing with other opponents by successes and mistakes (the rules may be transparent but a strategy has to be constructed). Even if learning from an example collection of given situations and appropriate reactions can be more effective, player could be confused by unknown pattern which was not included in a set of examples.

6 Conclusion

It is not a trivial task to address complex problems with support of artificial intelligence ways and means. This methods and know-hows can be adjusted to fit a given problem. Range of these addressed problems is so wide it cannot be simply defined because of too many application domains [1, 3, 11] where artificial intelligence is used [7, 8] including soft-computing applications [13, 17].

We have compared two different approaches of learning paradigms used to train the simple feedforward neural network. Probability model-based decision process is not an easy task for an artificial neural network. Therefore various training methods should be discussed while dealing with non-trivial problems.

Taking into account the simulation results in 5.2, selecting proper methods as neural network adaptation process should be considered. Depending on addressed problem, proper learning method can influence the whole adaptation process.

We found the supervised learning more efficient if used to train the neural network to help solve the Battleship game decision problem but the training set can easily lack important pattern information.

It is important to study and compare multiple approaches on various tasks. Then we will learn their differences and we will understand their inner processes.

Acknowledgements. This contribution was supported by the VEGA (Slovak Scientific Grant Agency) of the Ministry of Education of the Slovak Republic (ME SR) and of the Slovak Academy of Sciences (SAS) under the contract No. VEGA 1/0553/12 and VEGA 1/0458/13.

References

1. Abraham, A.: Hybrid soft computing and applications. International Journal of Computational Intelligence and Applications 8(1), 5–7 (2009)
2. Clementis, L.: Model driven classifier evaluation in rule-based system. In: Snasel, V., Abraham, A., Corchado, E.S. (eds.) SOCO Models in Industrial & Environmental Appl. AISC, vol. 188, pp. 267–276. Springer, Heidelberg (2013)
3. Corchado, A., Arroyo, A., Tricio, V.: Soft computing models to identify typical meteorological days. Logic Journal of the IGPL 19(2), 373–383 (2011)
4. Drugowitsch, J.: Design and Analysis of Learning Classifier Systems: A Probabilistic Approach. SCI. Springer, Heidelberg (2008)
5. Halavati, R., Shouraki, S., Lotfi, S., Esfandiar, P.: Symbiotic evolution of rule based classifier systems. International Journal on Artificial Intelligence Tools 18(1), 1–16 (2009)
6. Harmon, M., Harmon, S.: Reinforcement learning: A tutorial (1996), `http://www.nbu.bg/cogs/events/2000/Readings/Petrov/rltutorial.pdf`
7. Holland, J.: Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor (1975)
8. Holland, J.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. MIT Press, Cambridge (1992)

9. Kriesel, D.: A Brief Introduction to Neural Networks, Zeta version (2007), http://www.dkriesel.com
10. Krömer, P., Platos, J., Snášel, V., Abraham, A.: Fuzzy classification by evolutionary algorithms. In: SMC, pp. 313–318. IEEE (2011)
11. Lanzi, P.L., Stolzmann, W., Wilson, S.W. (eds.): IWLCS 1999. LNCS (LNAI), vol. 1813. Springer, Heidelberg (2000)
12. Qudrat-Ullah, H., Spector, J., Davidsen, P.: Complex decision making: theory and practice. Understanding complex systems. Springer (2008), http://books.google.sk/books?id=DDs1ps3YRWQC
13. Sedano, J., Curiel, L., Corchado, E., de la Cal, E., Villar, J.: A soft computing method for detecting lifetime building thermal insulation failures. Integrated Computer-Aided Engineering 17(2), 103–115 (2010)
14. Smith, M.: Neural Networks for Statistical Modeling. Thomson Learning (1993)
15. Sutton, R., Barto, A.: Reinforcement learning: an introduction. Adaptive computation and machine learning. MIT Press (1998), http://books.google.sk/books?id=CAFR6IBF4xYC
16. Watkins, C., Dayan, P.: Q-learning. Machine Learning 8(3-4), 279–292 (1992), http://jmvidal.cse.sc.edu/library/watkins92a.pdf
17. Zadeh, L.: Fuzzy logic, neural networks, and soft computing. Communication of the ACM 37(3), 77–84 (1994)
18. Zelinka, I., Davendra, D.D., Chadli, M., Senkerik, R., Dao, T.T., Skanderova, L.: Evolutionary dynamics as the structure of complex networks. In: Zelinka, I., Snasel, V., Abraham, A. (eds.) Handbook of Optimization. ISRL, vol. 38, pp. 215–243. Springer, Heidelberg (2013)

Evolutionary Algorithms for Parameter Estimation of Metabolic Systems

Anastasia Slustikova Lebedik and Ivan Zelinka

Abstract. For many years, computational tools have been widely applied to study such complex systems as metabolic networks. One of the principal questions in modeling of metabolic systems is the parameter estimation of model, which is related to a nonlinear programming problem. Two types of evolutionary algorithms, Differential Evolution and Self-Organizing Migrating Algorithm, are applied to the well-studied metabolic system, the urea cycle of the mammalian hepatocyte. The algorithms provide an effective approach in parameters identification of the model.

1 Introduction

Metabolic systems are highly non-linear with complex structure and dynamics. The complexity of interactions between components of metabolic system makes the prediction of the system behavior extremely challenging [1], [2]. To overcome this challenge, many researchers use computational and mathematical modeling methods.

From mathematical point of view, metabolic systems can be described in terms of ordinary differential equations (ODEs). In particular, the formulation of the system of ODEs for metabolic systems requires knowledge of biochemical reaction mechanisms. The dynamic behavior of the system is characterized by a set of parameters such as kinetic constants or rate constants. These parameters are often unknown due to highly complicated or even impossible experimental determination. In addition, most parameters defined in experiments *in vitro* are different

Anastasia Slustikova Lebedik
Tomas Bata University in Zlin, Faculty of Applied Informatics, T.G. Masaryka 5555,
760 01 Zlin, Czech Republic
e-mail: lebedik@fai.utb.cz

Ivan Zelinka
VŠB-Technical University of Ostrava, Faculty of Electrical Engineering and
Computer Science, Department of Computer Science, 17. listopadu 15,
708 33 Ostrava-Poruba, Czech Republic
e-mail: ivan.zelinka@vsb.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 201–209.
DOI: 10.1007/978-3-319-00542-3_21

from parameters measured *in vivo* [3]. Therefore, the estimation of the model parameters is required for prediction of the system dynamics.

Due to a large number of reactions, non-linear interactions between different metabolites, enzymes and other components of the system, the parameters estimation of metabolic systems can be formulated as non-linear programming (NLP) problem [4], [5]. A common approach for solving such a problem is the application of optimization methods. The optimization task is to minimize the difference between experimentally measured and the simulated data.

Different techniques are used to find the model parameters that make the model best fitting to experimental measurements. Application of local optimization algorithms is usually limited because of convergence to local optima [6]. Whereas using global optimization methods have been successfully applied to the parameter estimation problem. Among global optimization techniques, the use of evolutionary algorithms (EAs) should be highlighted. EAs are known as an efficient approach that can cope with large-scale systems. Recent studies have successfully applied EAs to parameter optimization problem [3], [5], [6] and others.

In present study, we have applied two modern evolutionary techniques to parameter estimation of the well-studied metabolic system, the urea cycle of the mammalian hepatocyte.

In our investigation, we consider three main questions:

1. Are the algorithms capable of precise parameter estimation of the urea cycle model?
2. Does the performance of studied evolutionary techniques depend on various algorithms settings?
3. Which algorithm performs best on parameter estimation of the metabolic system?

2 Experiment Design

In this research, the urea cycle is chosen for parameter optimization using evolutionary algorithms. Two modern efficient evolutionary techniques are applied to an optimization task which is described below.

2.1 The Urea Cycle Model

Our simulations were based on the mathematical model of the urea cycle of the mammalian hepatocyte described by P. W. Kuchel and P. J. Mulquiney [7]. Variation of different kinetic parameters of the enzymes of the urea cycle can effect metabolite concentration.

The model of urea cycle was developed to investigate dependence of metabolite concentrations on various kinetic parameters of the enzymes. The model includes four enzyme reaction schemes: arginase, ornithine carbomoyl transferase, argininosuccinate lyase, and argininosuccinate synthetase (see Fig. 1.). The metabolic model was formed by 12 ODEs and a set of kinetic parameters.

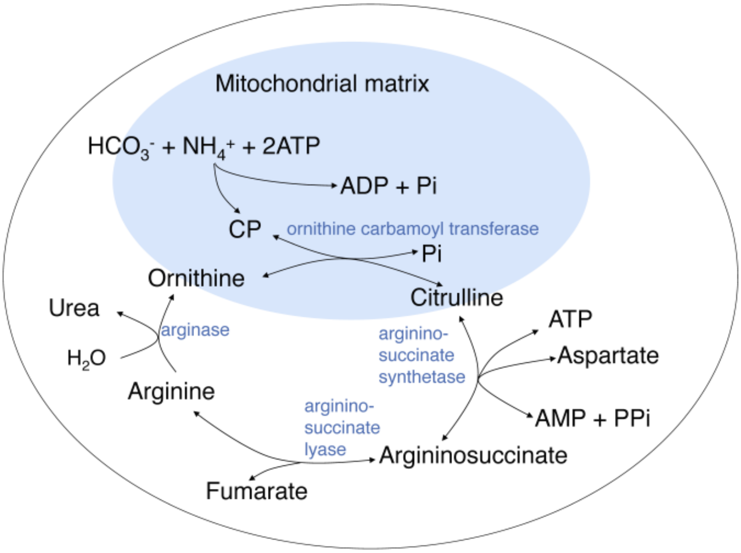


Fig. 1 The urea cycle in mammalian hepatocyte [7]

The detail description of the system can be found in the original source [7]. Overall, the number of kinetic parameters was 45. Due to the large number of unknown parameters, the parameter estimation of this problem is related to NLP problem.

2.2 Cost Function

Generally, parameter estimation of nonlinear systems is formulated as a task of minimization of cost function. The cost function in our research was stated as the sum of differences between experimentally measured and simulated data.

In our case, the experimentally measured data were replaced by simulated data using the true values of the model parameters obtained from [7]. Initial concentration of the studied metabolites: ornithine, citrulline, arginine and urea, were taken from the same source as the true values of the parameters.

2.3 Used Algorithms

Two types of evolutionary algorithms: Differential Evolution [8] and Self-Organizing Migrating Algorithm [9] have been considered in the experiments. The choice of DE was based on its published performance. And SOMA is a new and perspective global optimization technique.

DE is a floating-point encoded evolutionary algorithm for global optimization introduced by Storn and Price, see [8]. The initial population is randomly selected. New population members are generated using recombination and mutation. The performance of DE depends on the choice of the mutation and crossover strategies and control parameters such as the population size, crossover rate and the scale factor. The main advantages of DE are finding true global minimum regardless of the initial parameter values, fast convergence and using few control parameters.

The second used algorithm, called SOMA (see [9]), was used in this research together with DE. The SOMA algorithm is based on a cooperative-competitive principle. The main difference SOMA from other evolutionary techniques is that the new population is formed by principle, which is based on the social behavior of cooperating individuals. Moreover, unlike other evolutionary algorithms, SOMA uses a special parameter (PRT) instead of mutation. The crossover operator can be thought as the movement of an individual.

2.4 Algorithms Setting and Used Hardware

For the DE approach, we have firstly studied the impact of population size on the algorithm performance. Then, we have investigated the influence of factor F, which controls the amplification of the differential variation. The above mentioned settings with the minimum cost function value have been used for identification the best value of CR, the crossover constant. The details of the experiments are described in the section Results.

To study the performance of SOMA algorithm, we have varied only population size (PopSize).

The minimum of cost function value has been used as a quality measure of every set of algorithm settings.

The experiments were conducted using *Mathematica* 7. Each experiment was repeated 40 times. We have used the DERand1Bin version of DE and the AllToOne version of SOMA.

All calculations have been done using grid computer that includes 16 XServers, each 2x2 GHz Intel Xeon, 1 GB RAM, 80 GB HD i.e. 64 CPUs.

3 Results

Two evolutionary computation techniques: DE and SOMA were applied to well-studied metabolic system, the urea cycle of the mammalian hepatocyte. The model includes four enzyme reaction schemes: arginase, ornithine carbomoyl transferase, argininosuccinate lyase, and argininosuccinate synthetase.

The efficiency of particular method was judged based on cost function value.

All optimization techniques yield meaningful results. Each algorithm is capable of finding the model parameters. Performance of DE and SOMA is presented in Figure 2.

DE gives the best result with F=0.8, CR=0.6 and population size 900. The minimum of cost function value with these settings is 2.13×10^{-4}. In contrast to DE, SOMA reaches the best cost function of 5.61×10^{-5} with PopSize=135. However, in order to compare the performance of these two algorithms, we also take into account number of cost function evaluations. Hence, there have been chosen the results of SOMA simulations with PopSize=90. In this case, the cost function value is 8.43×10^{-5}, which is slightly higher but still lower in comparison with the best DE result. The implemented experimental setting in the above mentioned simulations are presented in Table 1.

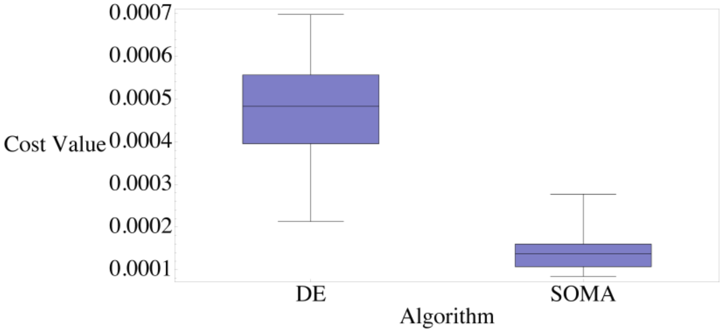


Fig. 2 Comparison of the optimization algorithms applied to the urea cycle model

Table 1 Settings for the algorithms

DE settings		SOMA settings	
NP	900	PathLength	3
F	0.8	Step	0.11
CR	0.6	PRT	0.1
Generations	150	PopSize	90
		Migrations	50
		MinDiv	-0.1

3.1 DE Experiments

To find the best settings for DE algorithm, we vary population size, the F value and CR. The minimum of cost function value is used as a quality measure of every set of algorithm settings.

Figure 3 depicts dependence of cost function value on various population sizes. To investigate the impact of population size, we apply a population size of 90, 450 and 900. These settings are equal to 2D, 10D and 20D, where D is a number of cost function arguments. In our case, it is 45.

We limit the investigation to only 3 sets of population size because of the execution time, which depends on the dimension of the problem. The dimension of the studied problem is high (45 parameters). Each experiment is repeated 40 times.

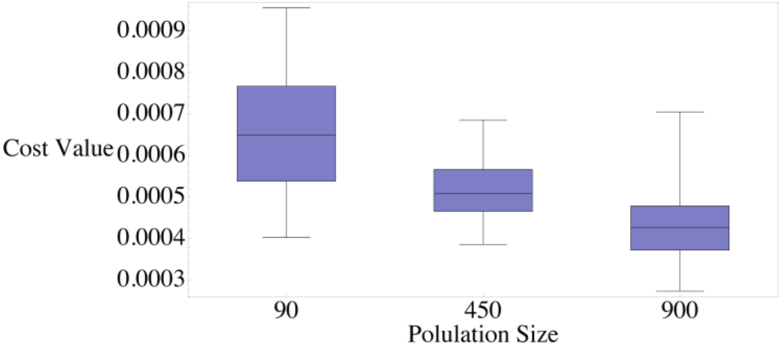


Fig. 3 The impact of population size on the cost function value for DE

The boxplots show that a population size of 900 reaches the best minimum result. The average cost function value decreases with increasing population size. However, all results give very low cost function value.

We continue to study the DE performance by varying the values for F (see Fig.4.) and CR (see Fig.5.). The influence of the F is tested using three F values: 0.1, 0.5 and 0.8. The CR value is set on 0.5. The DE algorithms yields the best results for F=0.8. Therefore, the best settings F=0.8 and population size of 900 are used for the next investigation.

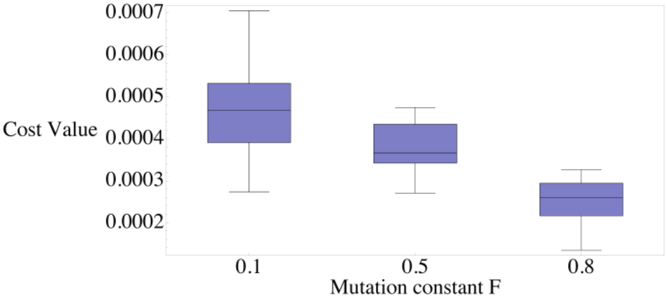


Fig. 4 The influence of mutation constant F on the cost function value for DE

To find the most successful combination of the algorithm settings, we vary CR from 0.1 to 0.9. Similarly to the above mentioned experiments, each calculation is repeated 40 times. Figure 5 depicts that CR=0.6 yields the best minimum result. It should be noticed that DE with all values of CR reaches meaningful results with the cost function value from 2.13×10^{-4} to 9.48×10^{-4}.

Figure 6 shows simulation of the system dynamics using predicted parameters (dashed) together with original parameters (solid). There are time courses of 4 main metabolites concentrations in the urea cycle simulation: ornithine, citrulline, arginine and urea. The figure depicts the best result of DE algorithm with F=0.8, CR=0.6 and population size 900.

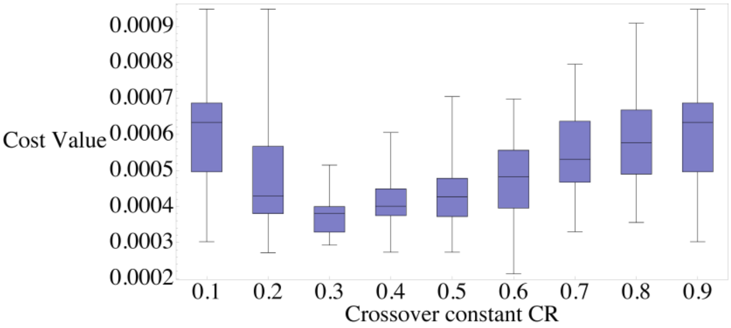


Fig. 5 The influence of crossover constant CR on the cost function value for DE

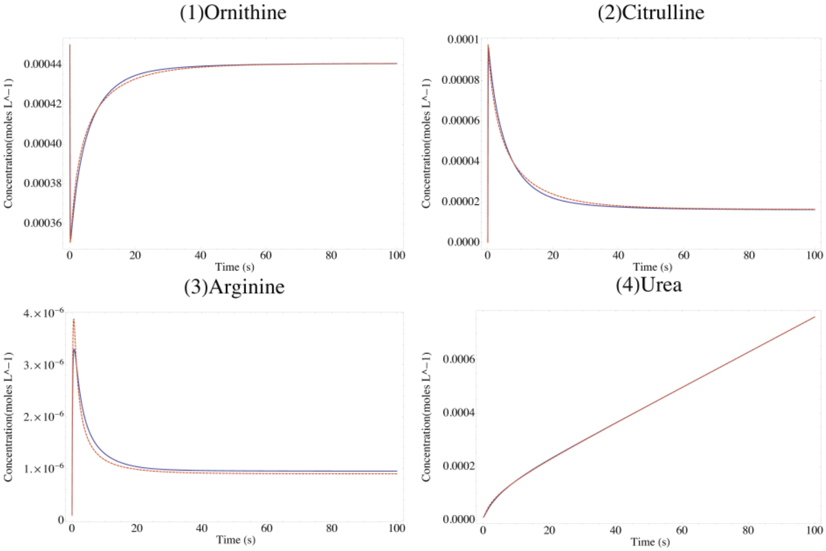


Fig. 6 The time courses of (1) ornithine, (2) citrulline, (3) arginine and (4) urea where predicted behavior by DE is dashed and original is solid

The DE algorithm performs very well. It is obvious that parameters of the model are predicted precisely.

3.2 SOMA Experiments

To study the performance of SOMA algorithm, we vary only population size (PopSize). The minimum of cost function value is used as a quality measure of every set of algorithm settings.

Similarly to DE, the study is limited to 3 sets of settings with population size of 45, 90 and 135, which equal 1D, 2D and 3D. The choice of population sizes for

SOMA algorithm is based on recommendations in [9]. The number of repetitions is again 40. The results of the experiments are shown in Figure 7.

The boxplots show that varying PopSize has similar impact on estimation process as in case of DE - the higher population size, the lower cost function value. SOMA yields the minimum of cost function value5.61×10^{-5} with PopSize=135. The worst result of SOMA algorithm is 2.77×10^{-4}, which is slightly higher than the best result of DE algorithm 2.13×10^{-4}.

Figure 8 depicts the time courses of 4 urea cycle metabolites with predicted and original parameters. The behavior of the system is predicted precisely.

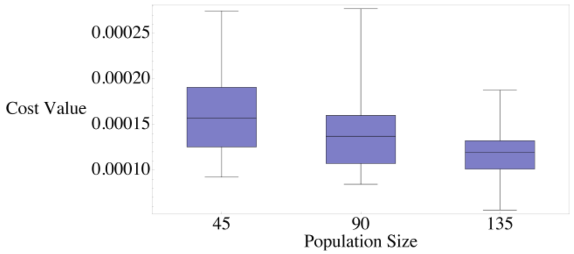


Fig. 7 The impact of population size (PopSize) on the cost function value for SOMA

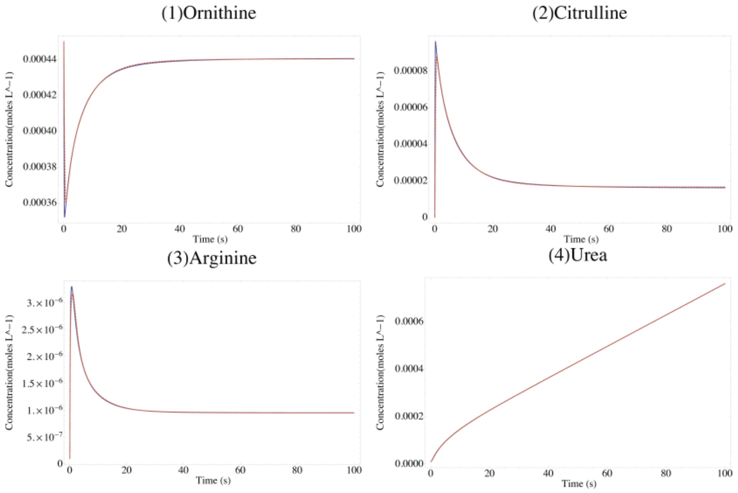


Fig. 8 The time courses of (1) ornithine, (2) citrulline, (3) arginine and (4) urea where predicted behavior by SOMA is dashed and original is solid

4 Conclusions

All applied optimization techniques yielded meaningful results. Each algorithm was capable of precise parameter estimation of the urea cycle model.

Varying algorithms settings could improve the algorithm performance. In DE simulations the influence of mutation constant F and crossover constant CR was significant. The best combination of these constants for the urea cycle simulations is F=0.8 and CR=0.6.In both cases DE and SOMA, increasing population size gave significantly better results, see Fig.3 and Fig.7. On the other hand, it required large computational effort. Considering computational time, the most time-consuming calculations were observed in SOMA simulations. However, it should be noted that SOMA provided the best performance in estimating parameters.

Taking into account that all algorithms performed well, and the fastest was DE, it is reasonable to apply DE in experiments with limited computational time.

Acknowledgments. This work was supported by grant No. IGA/FAI/2013/005 of the Internal Grant Agency of Tomas Bata University in Zlin.

References

1. Sun, J., Garibaldi, J.M., Hodgman, C.: Parameter Estimation Using Metaheuristics in Systems Biology: A Comprehensive Review. IEEE/ACM Transactions On Computational Biology and Bioinformatics 9(1), 185–202 (2012)
2. Mendes, P., Kell, D.B.: Non-Linear Optimization of Biochemical Pathways: Applications to Metabolic Engineering and Parameter Estimation. Bioinformatics 14(1), 869–883 (1998)
3. Dräger, A., Kronfeld, M., Ziller, M.J., Supper, J., Planatscher, H., Magnus, J.B., Oldiges, M., Kohlbacher, O., Zell, A.: Modeling metabolic networks in C. glutamicum: a comparison of rate laws in combination with various parameter optimization strategies. BMC Systems Biology 3(5) (2009)
4. Moles, C.G., Mendes, P., Banga, J.R.: Parameter Estimation in Biochemical Pathways: A Comparison of Global Optimization Methods. Genome Research 13, 2467–2474 (2003)
5. Polisetty, P.K., Voit, E.O., Gatzke, E.P.: Identification of Metabolic System Parameters Using Global Optimization Methods. Theoretical Biology and Medical Modelling 3(4), 1–15 (2006)
6. Rodriguez-Fernandez, M., Egea, J.A., Banga, J.R.: Novel Metaheuristic for Parameter Estimation in Nonlinear Dynamic Biological Systems. BMC Bioinformatics 7, 483–501 (2006)
7. Mulquiney, P.J., Kuchel, P.W.: Modeling metabolism with Mathematica. CRC Press (2003)
8. Price, K., Storn, R.M., Lampinen, J.A.: Differential Evolution: A Practical Approach to Global Optimization. Natural Computing. Springer (1995)
9. Zelinka, I.: SOMA – Self Organizing Migrating Algorithm. In: Babu, B.V., Onwubolu, G. (eds.) New Optimization Techniques in Engineering, vol. 33, ch. 7. Springer (2004)

Evolutionary Synthesis of Complex Structures – Pseudo Neural Networks for the Task of Iris Dataset Classification

Zuzana Kominkova Oplatkova and Roman Senkerik

Abstract. This research deals with a novel approach to classification. This paper deals with a synthesis of a complex structure which serves as a classifier. Classical artificial neural networks, where a relation between inputs and outputs is based on the mathematical transfer functions and optimized numerical weights, were an inspiration for this work. The proposed method utilizes Analytic Programming (AP) as the tool of the evolutionary symbolic regression. AP synthesizes a whole structure of the relation between inputs and output. Iris data (a known benchmark for classifiers) was used for testing of the proposed method. For experimentation, Differential Evolution for the main procedure and also for meta-evolution version of analytic programming was used.

1 Introduction

This paper deals with a new method for classification problems, which is based on evolutionary symbolic regression. The symbolic regression is able to synthesize a complex structure which is optimized by means of evolutionary computation. Such a structure can be used as a classifier because it can simulate the behavior of the Artificial Neural Networks (ANN) [1]-[4], from the inspiration for this work came from.

All artificial neural networks are based on some relation between inputs and output(s), which utilizes mathematical transfer functions and optimized weights from training process. The setting-up of layers, number of neurons in layers, estimating of suitable values of weights is a demanding procedure. On account of this fact, the novelty approach using symbolic regression with evolutionary computation is proposed in this paper.

Zuzana Kominkova Oplatkova · Roman Senkerik
Tomas Bata University in Zlin, Faculty of Applied Informatics, T.G. Masaryka 5555, 760 01 Zlin, Czech Republic
e-mail: {senkerik,oplatkova}@fai.utb.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 211–220.
DOI: 10.1007/978-3-319-00542-3_22

Symbolic regression in the context of evolutionary computation means to build a complex formula from basic operators defined by users. The basic case represents a process in which the measured data is fitted and a suitable mathematical formula is obtained in an analytical way. This process is widely known for mathematicians. They use this process when a need arises for mathematical model of unknown data, i.e. relation between input and output values. The symbolic regression can be used also for design of electronic circuits or optimal trajectory for robots and within other applications [5]-[11]. Everything depends on the user-defined set of operators. The proposed technique is similar to synthesis of analytical form of mathematical model between input and output(s) in training set used in neural networks. Therefore we can call this technique Pseudo Neural Networks. There is no optimization of number of nodes, connections or transfer function in nodes. This technique synthesizes a structure between inputs and output which transfer input values from training set items to output. The training is done by means of optimization procedure in evolutionary symbolic regression on the basis of output error function. The obtained structure is not possible to redraw to a pure ANN structure of nodes and connections.

This paper uses Analytic Programming (AP) [9]-[11] for evolutionary symbolic regression procedure. Besides AP, other techniques for symbolic regression computation can be found in literature, e.g. Genetic Programming (GP) introduced by John Koza [5]-[7] or Grammatical Evolution (GE) developed by Conor Ryan [8].

The above-described tools were recently commonly used for synthesis of artificial neural networks but in a different manner than is presented here. One possibility is the usage of evolutionary algorithms for optimization of weights to obtain the ANN training process with a small or no training error result. Some other approaches represent the special ways of encoding the structure of the ANN either into the individuals of evolutionary algorithms or into the tools like Genetic Programming. But all of these methods are still working with the classical terminology and separation of ANN to neurons and their transfer functions [13].

In this paper, iris plant dataset [14], [15] was used as a benchmark case for classification, which has been introduced by Fisher [15] for the first time. It is a well known dataset with 4 features and 3 classes. The attributes consist of sepal length, sepal width, petal length and petal width, which divides the plants into iris virginica, iris versicolor and iris sentosa. The data set was analyzed in a lot of papers by means of supervised and unsupervised neural networks [15]-[19], variations like distribution based ANN [20], piecewise linear classifier [21] or rough sets [22]. Not only pure ANN were used for classification but also evolutionary algorithms connected with fuzzy theory were employed [23], [24]. The tool from symbolic regression called Gene expression programming was used for classification too [25]. The last mentioned tool was used as a classifier, which contain procedures if-then rules in the evolutionary process. The basic components consist of greater then, less then, equal to, etc. and pointers to attributes.

The proposed technique in this paper is different. It synthesizes the structure without a prior knowledge of transfer functions and inner potentials. It synthesizes the relation between inputs and output of training set items used in neural networks so that the items of each group are correctly classified according the rules for cost function value.

Firstly, Analytic Programming used as a symbolic regression tool is described. Subsequently Differential Evolution used for main optimization procedure within Analytic Programming and also as a second algorithm within metaevolution purposes is mentioned. After that a brief description of artificial neural network (ANN) follows. Afterwards, the proposed experiment with differences compared to classical ANN is explained. The result section and conclusion finish the paper.

2 Analytic Programming

This tool was used for the synthesis of a complex structure which can behave similarly to supervised ANN and classify items from the training set into specified groups. Basic principles of the AP were developed in 2001 [9] - [12]. Until that time only genetic programming (GP) [6], [7] and grammatical evolution (GE) [8] had existed. GP uses genetic algorithms while AP can be used with any evolutionary algorithm, independently on individual representation.

The core of AP is based on a special set of mathematical objects and operations. The set of mathematical objects is set of functions, operators and so-called terminals (as well as in GP), which are usually constants or independent variables. This set of variables is usually mixed together and consists of functions with different number of arguments. Because of a variability of the content of this set, it is called here "general functional set" – GFS. The structure of GFS is created by subsets of functions according to the number of their arguments. For example GFS_{all} is a set of all functions, operators and terminals, GFS_{3arg} is a subset containing functions with only three arguments, GFS_{0arg} represents only terminals, etc. The subset structure presence in GFS is vitally important for AP. It is used to avoid synthesis of pathological programs, i.e. programs containing functions without arguments, etc. The content of GFS is dependent only on the user. Various functions and terminals can be mixed together [9].

The second part of the AP core is a sequence of mathematical operations, which are used for the program synthesis. These operations are used to transform an individual of a population into a suitable program. Mathematically stated, it is a mapping from an individual domain into a program domain. This mapping consists of two main parts. The first part is called discrete set handling (DSH) [9]-[12], [26] and the second one stands for security procedures which do not allow synthesizing pathological programs. The method of DSH, when used, allows handling arbitrary objects including nonnumeric objects like linguistic terms {hot, cold, dark…}, logic terms (True, False) or other user defined functions. In the AP DSH is used to map an individual into GFS and together with security procedures creates the above mentioned mapping which transforms arbitrary individual into a program (Fig. 1. and Fig. 2).

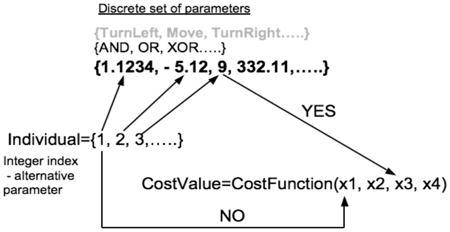

Discrete set of parameters
{TurnLeft, Move, TurnRight.....}
{AND, OR, XOR.....}
{1.1234, - 5.12, 9, 332.11,.....}
YES
Individual={1, 2, 3,.....}
Integer index - alternative parameter
CostValue=CostFunction(x1, x2, x3, x4)
NO

Fig. 1 Discrete set handling

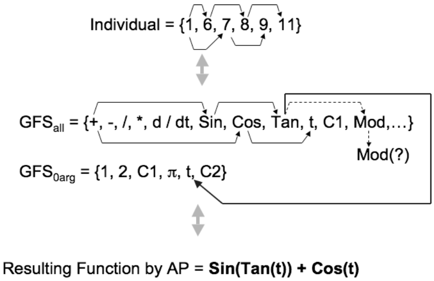

Individual = {1, 6, 7, 8, 9, 11}
GFS_all = {+, -, /, *, d / dt, Sin, Cos, Tan, t, C1, Mod,...}
Mod(?)
GFS_0arg = {1, 2, C1, π, t, C2}
Resulting Function by AP = Sin(Tan(t)) + Cos(t)

Fig. 2 Main principles of AP

AP needs some evolutionary algorithm that consists of population of individuals for its run. Individuals in the population consist of integer parameters, i.e. an individual is an integer index pointing into GFS. The individual contains numbers which are indices into GFS. The detailed description is represented in [9], [10].

AP exists in 3 versions – basic without constant estimation, AP_{nf} – estimation by means of nonlinear fitting package in *Mathematica* environment and AP_{meta} – constant estimation by means of another evolutionary algorithms; meta means meta-evolution.

3 Used Evolutionary Algorithm – Differential Evolution

This research used one evolutionary algorithm: Differential Evolution (DE) [27], [28] for main process and also meta-evolutionary process in AP. Future simulations expect a usage of soft computing GAHC algorithm (modification of HC12) [29] and a CUDA implementation of HC12 algorithm [30].

DE is a population-based optimization method that works on real-number-coded individuals [27], [28], [26]. DE is quite robust, fast, and effective, with global optimization ability. It does not require the objective function to be differentiable, and it works well even with noisy and time-dependent objective functions.

For each individual $\vec{x}_{i,G}$ in the current generation G, DE generates a new trial individual $\vec{x}'_{i,G}$ by adding the weighted difference between two randomly selected individuals $\vec{x}_{r1,G}$ and $\vec{x}_{r2,G}$ to a randomly selected third individual $\vec{x}_{r3,G}$. The resulting individual $\vec{x}'_{i,G}$ is crossed-over with the original individual $\vec{x}_{i,G}$. The fitness of the resulting individual, referred to as a perturbed vector $\vec{u}_{i,G+1}$, is then compared with the fitness of $\vec{x}_{i,G}$. If the fitness of $\vec{u}_{i,G+1}$ is greater than the fitness of $\vec{x}_{i,G}$, then $\vec{x}_{i,G}$ is replaced with $\vec{u}_{i,G+1}$; otherwise, $\vec{x}_{i,G}$ remains in the population as $\vec{x}_{i,G+1}$. Description of used DERand1Bin strategy is presented in (1). Please refer to [27], [28] for the description of all other strategies.

$$u_{i,G+1} = x_{r1,G} + F \bullet \left(x_{r2,G} - x_{r3,G}\right) \tag{1}$$

4 Artificial Neural Networks

Artificial neural networks are inspired in the biological neural nets and are used for complex and difficult tasks [1] - [4]. The most often usage is classification of objects as also in this case. ANNs are capable of generalization and hence the classification is natural for them. Some other possibilities are in pattern recognition, control, filtering of signals and also data approximation and others.

There are several kinds of ANN. Simulations were based on similarity with feedforward net with supervision. ANN needs a training set of known solutions to be learned on them. Supervised ANN has to have input and also required output.

The neural network works so that suitable inputs in numbers have to be given on the input vector. These inputs are multiplied by weights which are adjusted during the training. In the neuron the sum of inputs multiplied by weights are transferred through mathematical function like sigmoid, linear, hyperbolic tangent etc. Therefore ANN can be used for data approximation [2] – a regression model on measured data, relation between input and required (measured data) output.

These single neuron units (Fig. 3) are connected to different structures to obtain different structures of ANN (e.g. Fig. 4), where $\sum\delta = TF[\sum(w_i x_i + bw_b)]$ and $\sum = TF[\sum(w_i x_i + bw_b)]$; TF is for example logistic sigmoid function.

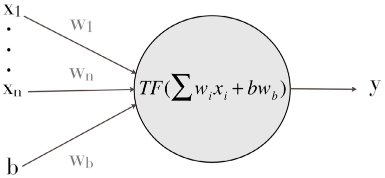


Fig. 3 Neuron model, where TF (transfer function like sigmoid), x1 - xn (inputs to neural network), b – bias (usually equal to 1), w1 – wn, wb – weights, y – output

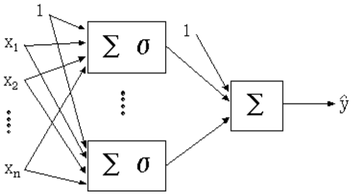


Fig. 4 ANN models with one hidden layer

The example of relation between inputs and output can be shown as a mathematical form (2). It represents the case of only one neuron and logistic sigmoid function as a transfer function.

$$y = \frac{1}{1 + e^{-(x_1 w_1 + x_2 w_2)}} \ , \tag{2}$$

where y – output
x1, x2 – inputs
w1, w2 – weights.

The aim of the proposed technique is to find similar relation to (2). This relation is completely synthesized by evolutionary symbolic regression – Analytic Programming.

5 Problem Design and Iris Dataset Definition

For this classification problem, iris plant dataset [14], [15] was used as a benchmark case for classification. It is a well known dataset with 4 features and 3 classes. The attributes consist of sepal length, sepal width, petal length and petal width, which divides the plants into iris virginica, iris versicolor and iris sentosa. This set contains 150 instances. Half amount was used as training data and the second half was used as testing data. The cross validation is planned for future testing. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other. The attributes were of real values. Usually, the class is defined by 3 output neurons in classical artificial neural net and binary code. In this paper, AP was able to create pseudo neural net structure between inputs and one output. Therefore, three classes were designed as continues value of the output, i.e. iris setosa was -1 for output less than -1, iris virginica has value 1 for output greater than 1, and iris versicolor was 0 for outputs between -1 and 1. The AP was used as a MISO classifier (Multi Inputs Single Output). The future investigation will include the methodology for synthesis MIMO classifiers by means of AP.

The cost function value is given in eq. (3), i.e. if the cv is equal to zero, all n training patterns are classified correctly.

$$cv = \sum_{i=1}^{n} \left| requiredOutput - currentOutput \right| \tag{3}$$

6 Results

As described in section about Analytic Programming, AP requires some EA for its run. In this paper AP$_{meta}$ version was used. Meta-evolutionary approach means usage of one main evolutionary algorithm for AP process and second algorithm for coefficient estimation, thus to find optimal values of constants in the structure of pseudo neural networks.

In this paper, DE was used for main AP process and also in the second evolutionary process. Settings of EA parameters for both processes were based on performed numerous experiments with chaotic systems and simulations with AP$_{meta}$ (Table 1 and Table 2).

Table 1 DE settings for main process of AP

PopSize	40
F	0.8
Cr	0.8
Generations	50
Max. CF Evaluations (CFE)	2000

Table 2 DE settings for meta-evolution

PopSize	40
F	0.8
Cr	0.8
Generations	150
Max. CF Evaluations (CFE)	6000

The set of elementary functions for AP was inspired in the items used in classical artificial neural nets. The components of input vector x contain values of each attribute (x_1, x_2, x_3, x_4). Thus AP dataset consists only of simple functions with two arguments and functions with zero arguments, i.e. terminals. Functions with one argument, e.g. Sin, Cos, etc., were not required.

Basic set of elementary functions for AP:

GFS2arg= +, -, /, *, ^, exp
GFS0arg= x_1, x_2, x_3, x_4, K

The length of the individual (number of arguments in the individual) is equal to 40, which means the maximum amount of used basic operators is 40. The complexity of the search space for evolutionary computation is high and the number of combinations in the numbers of order 10^{41} for 11 basic elements and 40 places in the individual. Total number of cost function evaluations for AP was 2000, for the second EA it was 6000, together 12 millions ($12x10^6$) cost function evaluations per each simulation. This means 8 times less possibilities than a number of combinations in the search space. It will be necessary about 10^{32} years for evaluating of all cases if one simulation would take 0.1 second (some takes even several seconds because of the meta-evolutionary approach). Evolutionary computation is an effective tool for optimization because the solution is found without evaluating of all simulation cases. The one whole simulation according to the above mentioned settings takes around 27 hours dependent on the processor speed. When the simulations finished, the obtained minimum was searching in the whole history when this value was reached for the first time. The values were reached from 5 to 501 generations of the main AP process, which have to be multiplied by 6000 of the meta-evolutionary cost function evaluations, i.e. $3\text{x}10^4$ to $3\text{x}10^6$.

In this preliminary study, 15 simulations were carried out. The following table (Table 3) shows the results of achieved cost value. It seems that the function has two local extremes close to each other with the value 2 and 3.

Table 3 Cost value results statistics

Achieved cost value	Number of cases	Number of cases [%]
2 misclassified items	11	73.3
3 misclassified items	4	26.7

From carried simulations, several notations of input dependency were obtained. The presented result (4) had a training error equal to 2 misclassified items (2.66% error, 97.34% success from 75 training patterns) and testing error equal to 3 misclassified items (4% error, 96% success from 75 testing patterns). The results correspond with the similar results presented in [16], [18], [22].

$$\begin{aligned} y = & \left(e^{-0.0066007(x_4+693.44)}\right)^{-0.00311885\left(x_3+e^{x3}-144.991\right)} \\ & -953.535e^{x_3(-x_4-637.691)}\left(-x_2-x_4+\frac{6.11200456962\times 10^{625}}{(x_2x_4)^{223.656}}\right) \end{aligned} \tag{4}$$

The testing showed that items nr. 19 and 21 in the group of iris versicolor were incorrectly classified as iris sentosa.

8 Conclusions

This paper deals with a novel approach – pseudo neural networks. The proposed approach synthesizes a complex structure, which behaves similarly to artificial neural networks. Within this research, classical optimization of the structure or weights was not performed. The proposed technique produces the final analytical shape of the dependency of inputs to output. It is based on symbolic regression with evolutionary computation. It synthesizes a whole structure in symbolic form without a prior knowledge of the ANN structure, transfer functions and weights.

The complexity of search space was high but evolutionary computation is able to find the complex structure with the required behaviour within a fraction of the total amount of possibilities. As can be seen from the result section, such approach is promising. Future plans will be focused on further tests and comparisons with classical neural nets and other benchmark cases for classification problems. The cross validation will be applied in future simulations too to find out if the better behavior with zero misclassification can be obtained.

Acknowledgments. This work was supported by European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089.

References

[1] Gurney, K.: An Introduction to Neural Networks. CRC Press (1997) ISBN: 1857285034

[2] Hertz, J., Kogh, A., Palmer, R.G.: Introduction to the Theory of Neural Computation. Addison – Wesley (1991)

[3] Wasserman, P.D.: Neural Computing: Theory and Practice. Coriolis Group (1980) ISBN: 0442207433

[4] Fausett, L.V.: Fundamentals of Neural Networks: Architectures, Algorithms and Applications. Prentice Hall (1993) ISBN: 9780133341867

[5] Back, T., Fogel, D.B., Michalewicz, Z.: Handbook of evolutionary algorithms. Oxford University Press (1997) ISBN 0750303921

[6] Koza, J.R., et al.: Genetic Programming III; Darwinian Invention and problem Solving. Morgan Kaufmann Publisher (1999) ISBN 1-55860-543-6

[7] Koza, J.R.: Genetic Programming. MIT Press (1998) ISBN 0-262-11189-6

[8] O'Neill, M., Ryan, C.: Grammatical Evolution. Evolutionary Automatic Programming in an Arbitrary Language. Kluwer Academic Publishers (2003) ISBN 1402074441

[9] Zelinka, et al.: Analytical Programming - a Novel Approach for Evolutionary Synthesis of Symbolic Structures. In: Kita, E. (ed.) Evolutionary Algorithms. InTech (2011) ISBN: 978-953-307-171-8

[10] Oplatkova, Z.: Metaevolution: Synthesis of Optimization Algorithms by means of Symbolic Regression and Evolutionary Algorithms. Lambert Academic Publishing, Saarbrücken (2009) ISBN: 978-3-8383-1808-0

[11] Zelinka, I., Varacha, P., Oplatkova, Z.: Evolutionary Synthesis of Neural Network. In: Mendel 2006 – 12th International Conference on Softcomputing, Brno, Czech Republic, May 31-June 2, pp. 25–31 (2006) ISBN 80-214-3195-4

[12] Zelinka, I., Oplatkova, Z., Nolle, L.: Boolean Symmetry Function Synthesis by Means of Arbitrary Evolutionary Algorithms-Comparative Study. International Journal of Simulation Systems, Science and Technology 6(9), 44–56 (2005) ISSN: 1473-8031

[13] Fekiac, J., Zelinka, I., Burguillo, J.C.: A review of methods for encoding neural network topologies in evolutionary computation. In: ECMS 2011, Krakow, Poland (2011) ISBN: 978-0-9564944-3-6

[14] Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annals of Eugenics 7(2), 179–188 (1936), doi:10.1111/j.1469-1809.1936.tb02137.x

[15] Machine learning repository with Iris data set, http://archive.ics.uci.edu/ml/datasets/Iris

[16] Swain, M., et al.: An Approach for Iris Plant Classification Using Neural Network. International Journal on Soft Computing 3(1) (2012), doi:10.5121/ijsc.2012.3107

[17] Shekhawat, P., Dhande, S.S.: Building and Iris Plant Data Classifier Using Neural Network Associative Classification. International Journal of Advancements in Technology 2(4), 491–506 (2011) ISSN: 0976-4860

[18] Avci, M., Yildirim, T.: Microcontroller Based Neural Network Realization and Iris Plant Classifier Application. In: Proceedings of the Twelfth Turkish Symposium on Artificial Intelligence and Neural Networks (TAINN 2003), Canakkale, Turkey, July 2-4 (2003)

[19] Osselaer, S.: Iris Data Analysis Using Back Propagation Neural Networks. Journal of Manufacturing Systems 13(4), 262 (2003)

[20] Chen, S., Fang, Y.: A New Approach for Handling Iris Data Classification Problem. International Journal of Applied Science and Engineering (2005) ISSN: 1727-2394

[21] Kostin, A.: A simple and fast multi-class piecewise linear pattern classifier. Pattern Recognition 39(11), 1949–1962 (2006) ISSN 0031-3203, doi:10.1016/j.patcog.2006.04.022

[22] Kim, D.: Data classification based on tolerant rough set. Pattern Recognition 34(8), 1613–1624 (2001) ISSN 0031-3203, doi:10.1016/S0031-3203(00)00057-1

[23] Agustı´n-Blas, L.E., et al.: A new grouping genetic algorithm for clustering problems. Expert Systems with Applications 39(10), 9695–9703 (2012) ISSN 0957-4174, doi:10.1016/j.eswa.2012.02.149

[24] Zhou, E., Khotanzad, A.: Fuzzy classifier design using genetic algorithms. Pattern Recognition 40(12), 3401–3414 (2007) ISSN 0031-3203, doi:10.1016/j.patcog.2007.03.028

[25] Ferreira, C.: Gene Expression Programming: Mathematical Modelling by an Artificial Ingelligence, ISBN: 9729589054

[26] Lampinen, J., Zelinka, I.: New Ideas in Optimization – Mechanical Engineering Design Optimization by Differential Evolution, vol. 1, 20p. McGraw-Hill, London (1999) ISBN 007-709506-5

[27] Price, K., Storn, R.M., Lampinen, J.A.: Differential Evolution: A Practical Approach to Global Optimization, 1st edn. Natural Computing Series. Springer (2005)

[28] Price, K., Storn, R.: Differential evolution homepage (2001), http://www.icsi.berkeley.edu/~storn/code.html (accessed February 29, 2012)

[29] Matousek, R., Zampachova, E.: Promising GAHC and HC12 algorithms in global optimization tasks. Optimization Methods & Software 26(3), 405–419 (2011) ISSN 1055-6788

[30] Matousek, R.: HC12: The Principle of CUDA Implementation. In: Proceedings of 16th International Conference On Soft Computing, Mendel 2010, pp. 303–308 (2010) ISBN 9788-0-214-4120-0

Speech Emotions Recognition Using 2-D Neural Classifier

Pavol Partila and Miroslav Voznak

Abstract. This article deals with a speech emotion recognition system. We discuss the usage of a neural network as the final classifier for human speech emotional state. We carried our research on a database of records of both genders and various emotional states. In the preprocessing and speech processing phase, we focused our intent on parameters dependent on the emotional state. The output of this work is a system for classifying the emotional state of a man's voice, which is based on a neural network classifier. For output-stage classifier was used self-organizing feature map, which is specific type of artificial neural nets. The number of input parameters must be limited for hardware and time consuming computation of neurons positions. Therefore we discuss the accuracy of the classifier whose input is the fundamental frequency calculated by different methods.

Keywords: Fundamental frequency, Digital speech processing, Emotions, Neural network, Auto-correlation.

1 Introduction

There are many areas in which the information about the emotional state is needed. Nowadays, technological development puts more emphasis on the increased accuracy and simplicity of communication between man and computer. Modern applications use the speech for input-output interface increasingly. In this type of interaction two problems can occur, caused by the absence of information about the emotional state. The first one is an incorrect recognition of a word or a command from a person who is under stress. The machine recognizes human speech differently than a man with his hearing. The accuracy is affected by changes in the voice signal due to stress on the vocal tract. The second problem is that we feel the absence of emotional state in the machine speech of the loudspeaker. Classic applications such as Text-To-Speech combine parts of speech

Pavol Partila · Miroslav Voznak
VSB-Technical University of Ostrava, Dpt. of Telecommunications,
17. Listopadu 15/2172, 708 33 Ostrava-Poruba, Czech Republic
e-mail: {pavol.partila,miroslav.voznak}@vsb.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 221–231.
DOI: 10.1007/978-3-319-00542-3_23

sounds that are truly correct, but ultimately this signal is without any emotion. Such speech acts on the man and is synthetically unreliable.

There are several physiological criteria such as for example heart rate, breathing changes and sweating, which enable determining the emotional state of a man. A number of speech signal parameters is used in speech processing. An imperfect human ear responds to parameters such as intensity, intonation, and speech rate. Fundamental frequency of speech, zero crossings rate, energy and cepstral coefficients are parameters which are used in digital speech processing. The following sections contain a description of signal processing methods, selection of the training sets and neural network classifier [1], [2].

2 Pre-processing

Speech signal is stochastic by nature. However, a speech signal has a number of characteristics that may be unwanted during processing. This chapter describes a process called pre-processing that is an important part of the digital speech signal processing. These few steps prepare the signal for subsequent extraction of signal parameters. The values of these parameters could be wrong without the pre-processing process. Figure 1 shows this pre-processing.

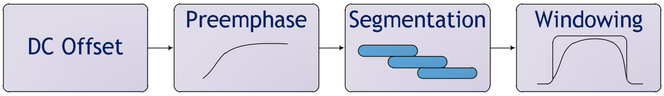


Fig. 1 Pre-Processing Process

2.1 Removing DC Component

Speech signal digitizing and sound cards processing can also have side effects. Sound cards insert the DC component into the speech signal. It may not be suitable for the calculation of parameters such as signal energy or other. It is therefore necessary to remove the dc component from the speech signal. The DC component of speech signal is calculated as the mean of all samples, as described by Equation (1). Variable *s(n)* is a sample of *n* of the total number of *N*.

$$\mu_s = \frac{1}{N}\sum_{n=1}^{N} s(n) \tag{1}$$

Unwanted DC offset is removed by subtracting the mean of each sample, which is shown in Equation (2).

$$s'(n) = s(n) - \mu_s \tag{2}$$

In many cases, we do not have all the audio. This case is typical for real-time applications. This means that we cannot estimate the true mean because the signal changes unpredictably. Thus, in real-time processing it is necessary to calculate the mean value for each sample. The mean value for the current sample $\mu_s(n)$ from formula (3) can be determined once we know the mean value of the previous sample $\mu_s(n-1)$, which is linked to the actual sample by constant γ. Its value approaches 1. The DC component is removed by a simple subtraction of the mean value [2], [3].

$$\mu_s(n) = \gamma\mu_s(n-1) + (1+\gamma)s(n) \tag{3}$$

2.2 Increasing the Energy of Higher Frequencies

One of the characteristics of the speech signal is also the energy of the signal decreases with increasing frequency. Most speech signal energy is in the first 300Hz of spectra, which means that the information of the higher frequencies expires against higher energies from the bottom of the spectrum. Saving the higher end of the spectrum is achieved by artificially increasing the energy in the higher part of spectra.

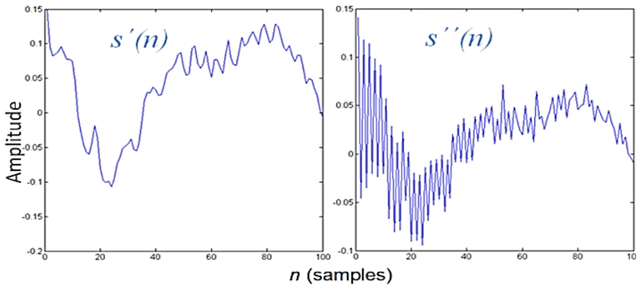


Fig. 2 One frame before and after pre-emphasis

Increasing energy of the spectrum part is performed using pre-emphasis. The speech signal is applied to the first order FIR filter whose transfer function H(z) is described by Equation (4). FIR filter applied on real values (5) return the speech signal samples *s´´(n)* with the enhanced energy at higher frequencies of spectra. Variable k approaches 1, i.e. it ranges from 0.95 to 1. The effect of pre-emphasis is shown in Fig. 2.

$$H(z) = 1 - kz^{-1} \tag{4}$$

$$s''(n) = s'(n) - ks(n-1) \tag{5}$$

2.3 Time Dividing

As mentioned above, speech signal has a stochastic character. From a mathematical point of view, it is very difficult to find dependency and frequency in this signal. For this reason it is necessary to divide the speech signal into smaller parts called frames. A frame length is chosen between 20 and 30ms. This length is derived from the lag of the human vocal tract. Values of samples in neighboring frames may vary rapidly, therefore frames overlapping are appropriate. The frame overlap is selected in half, as shown in Fig. 3.

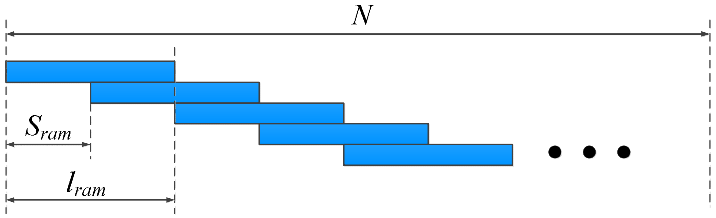


Fig. 3 Time divided speech signal to frames with half-frame overlapping

The number of frames is calculated using Equation (6). Brackets in the formula represent the floor function, which ensures rounding down. Variable *N* is the number of all samples of tracks, l_{frame} the frame length and s_{frame} their shift [7].

$$N_{frame} = 1 + \left\lfloor \frac{N - l_{frame}}{s_{frame}} \right\rfloor \tag{6}$$

2.4 Window Function

Processing speech signals between frames can have a side effect because the edges of neighboring frames can have sharp transitions. This can have a bad influence on speech processing and frequency analysis mostly. The disadvantage mentioned can be removed by applying the window function on each frame. A lot of window functions are used in speech processing and the choice depends on the characteristics of the following processing methods. Hamming window function is used in most such cases due to its suitable properties in both the time domain and frequency.

$$w(n) = 0{,}54 + 0{,}46\cos\left[\left(\frac{1}{2}N - n\right)\frac{2\pi}{N}\right] \tag{7}$$

Hamming window function is defined by the Equation (7), where variable *N* represents all samples and n sample specific (position in time) [5].

3 Speech Processing Parameters

Volume, intonation and tempo are speech characteristics which can be recognized by a human ear. In DSP (digital signal processing) and speech processing in particular, we use some other parameters which characterize speech signal and human vocal tract. These parameters and their computing are described in next subchapters.

3.1 Signal Energy and ZCR

Signal energy and Zero Crossing Rate are also important in speech processing. These two parameters were used as a voice activity detector to eliminate silence or noise. Signal energy is characterized by intensity, and the human ear can sense it as a volume. Energy is influenced by the way of recording and digitizing speech, speaker distance from the microphone, and other features. Voiced and unvoiced parts can be separated using a sound energy profile. The calculation is described in Equation (8).

$$E = \frac{1}{N}\sum_{n}^{N-1}\left[s(n)\right]^2 \tag{8}$$

Zero Crossing Rate describes how many times the polarity of the signal changes, in other words, how many times it crosses zero. This parameter can also carry information about F0 change. ZCR carries information about both the voice activity energy. ZCR is calculated using sign function, as shown in Equation (9).

$$ZCR(m) = \sum \left| sign\left[s(n)\right] - sign\left[s(n-1)\right] \right| \tag{9}$$

3.2 Fundamental Frequency of Vocal Cords

This parameter (F0) is one of the most important in speech processing because it carries a lot of the information about the vocal tract, and thus also the about basic features of man.

Age, gender, speech errors and emotional state of a man can be determined using this parameter. There are several methods in signal processing which enable estimating the fundamental frequency. Human speech consists of voiced and unvoiced speech sounds. The vocal cords are almost completely open in the creation of voice-less phonemes. The basic tone does not arise with opened vocal cords and therefore F0 can be calculated only from the voiced parts. Each method to calculate F0 has advantages and disadvantages. This article compares four methods and their use in a self-organizing feature map classifier.

Auto-correlation function. This simple method can be used where we have a full signal. The auto-correlation function is defined in Equation (10).

$$R(m) = \sum_{n=m}^{N-1} s(n)\,s(n-m) \tag{10}$$

Lag k is determined by the position of maxima.

$$k = \arg MAX_m R(m) \tag{11}$$

The autocorrelation process is reversible and therefore the signal has a double length after this process. The distance between the first and the second peak is k, as it shown below [4].

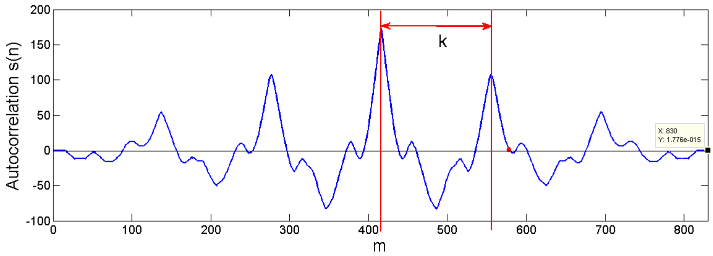


Fig. 4 Autocorrelation and pitch extraction

The disadvantage of this method is that the very high energy of original framework can absorb all the information from the other. The fundamental frequency of vocal cords is calculated using Equation (12).

$$F_0 = \frac{F_{sampling}}{k} \tag{12}$$

Normalized cross-correlation function. The difference in energy shifted and original framework is settled by normalization:

$$NCCF(m) = \frac{\sum_{n=m}^{N-1} s(n)s(n-m)}{\sqrt{E_1 E_2}} \tag{13}$$

Where E1 is energy of original frame and E2 is energy of shifted frame, of course the lag and F0 are calculated. F0 extraction is performed just as in the ACF expressed in equations (11) and (12) as is described in [5] and [9].

Auto-correlation with central clipping. Voiced and voiceless parts of the signal are separated with the threshold level. The speech signal constantly changes and thus it is not possible to to use a single thresh-old for the entire signal. The threshold is determined for each frame separately, using the equation bellow.

$$P(i) = \alpha MIN(MAX_{i-1}, MAX_{i+1}) \tag{14}$$

Peaks of neighboring frames *i-1* and *i+1* are established. The threshold level is determined as the lower value of the two. Constant α is set within the range 0.8 to 1. The original signal of frame *i* is weighted with threshold *P(i)*. Samples with a higher value are clipped and normalized to 1 and lower to 0, as can be seen in Figure 5 bellow.

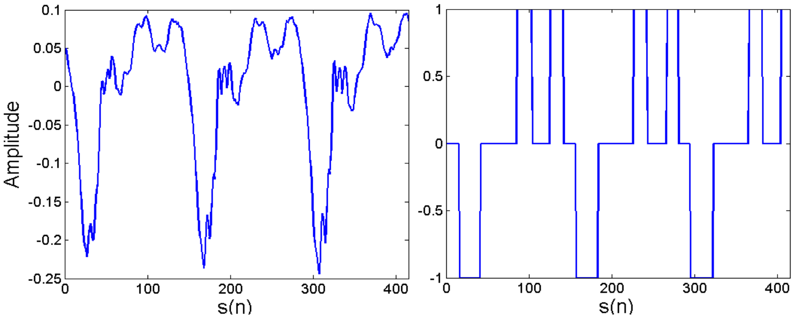


Fig. 5 Frame of speech signal before (left) and after clipping and normalization (right)

Next step is the auto-correlation and establishing the position of the second main lobe. F0 calculation is simple and the same as described in previous methods. It is only necessary to find the position lag and use Equation (12).

Subharmonic-To-Harmonic Ratio. This method is developed in frequency domain. *A(f)* is short-term spectrum function. Suppose that f_0 is fundamental frequency. The sum of harmonic amplitude is defined below.

$$SH = \sum_{n=1}^{N} A(nf_0) \tag{15}$$

N is number of all harmonics which can be considered. If we consider the subharmonic frequency, that is the one half of f_0. The sum of subharmonic amplitude is described in Equation (16).

$$SS = \sum_{n-1}^{N} A\left[\left(n - \frac{1}{2}\right) f_0\right] \tag{16}$$

Subharmonic-to-harmonic ratio can be obtained by dividing SS and SH:

$$SHR = \frac{SS}{SH} \tag{17}$$

The conversion between time and frequency domain is done by classically read and understood procedures of signal processing. This method is described in more detail in references [6].

Self-organizing Feature Map

The emotional state classifier is based on self-organizing maps (SOM). These maps represent a specific type of neural networks with uncontrolled competitive learning [11]. There are generally two-dimensional maps of neurons (Fig. 6.). The learning process of SOM is uncontrolled which means that the input data do not need to know the output. In the process of learning SOM themselves determine how to classify the inputs. At the beginning of learning, the weights of all the inputs of all neurons can be set randomly. Randomly selected input vectors are applied to neurons and then analyzed in order to find one which is most similar to input. This neuron is called the winner. The weights of neighboring neurons are adjusted according to the following rule in equation (18). This equation describes the weight between neurons *i* and *j* for *t+1* iteration and input $x_j(t)$. Next iteration means a new vector on input, finding new winner and changing weights between neurons again. After the learning process is completed, the map has a shape that represents the characters of input parameters.

$$w_{ij}(t+1) = w_{ij}(t) + \gamma\left(x_j(t) - w_{ij}(t)\right) \tag{18}$$

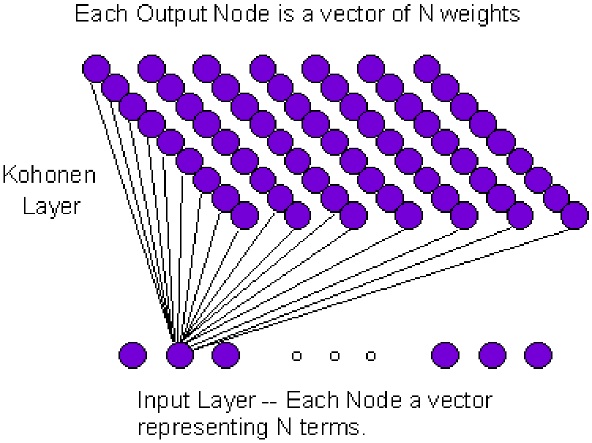

Each Output Node is a vector of N weights
Kohonen Layer
Input Layer -- Each Node a vector representing N terms.

Fig. 6 Two-dimensional neural map of Kohonen's layer

The aim of this article is to clarify the appropriate choice of parameters for the two-dimensional maps. They are two-dimensional because counting the change weights between neurons for more than 2D dimensions poses high demands on hardware. This should be taken into account as these days real time speech processing is sought after in the entire telecommunications services sector [10].

5 Results

The system which was designed in software Matlab finally consists of three main blocks. These have been described above. Figure 7 shows the block diagram.

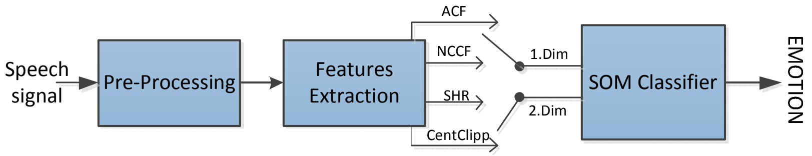


Fig. 7 Speech emotion recognition system

The input speech signal is embedded into the system sequentially. Audio records of different emotional states were used as the speech signals. The first block algorithm performed the pre-processing operations. The next block is designed to extract the basic tone using the four above-mentioned methods. These parameters are input data for the neural network. The combination of the four methods ACF, NCCF, Clipping Central and SHR yielded pairs of inputs.

Table 1 Pairs of SOM classifiers inputs

	ACF	ACF	ACF	NCCF	NCCF	SHR
Input	NCCF	SHR	CClipp	SHR	CClipp	CClipp
Res.	Bad	Normal	Normal	**Worst**	Bad	**Best**

Mixed couples all calculation methods F0 yielded different results. The worst decision was level classifier at NCCF versus SHR. The best classifier had a distinctive character input methods in SHR and Clipping Central, these two methods are less prone to adverse extract F0 from unvoiced parts of the signal. Best and worst results are shown in Fig. 8.

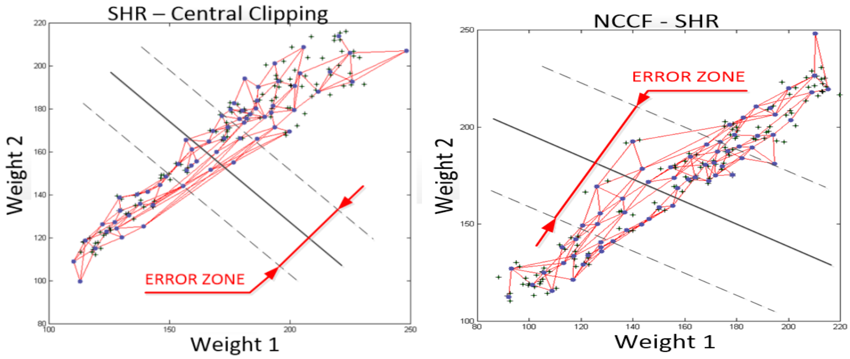


Fig. 8 Trained self-organizing map with error zones for NCCF and SHR methods (left) and for SHR and Central Clipping methods (right)

6 Conclusion

This paper focused on the distinctive character classifier. Using different methods for calculating the fundamental frequency of human voice can yield results of different accuracy. Therefore, it is recommended to apply at least two methods for calculating F0. The second perspective is not appropriate to use high dimensional maps for the classification of emotional state. Such classifiers are hardware and time consuming. The problem of delay is unacceptable for telecommunications services and applications that require real-time access.

The block algorithm for pitch extraction is designed to calculate F0 using four methods. Inputs for the two-dimensional neural network were created by combining these four methods. Subharmonic-To-Harmonic Ratio, and Central Clipping methods were considered more reliable than others. It is interesting to note that the classifier has the highest error rate for combined SHR and NCCF. The SOM classifier has the lowest error rate, and thus the best resolving power between normal and stress emotional state, provided SHR and Central Clipping methods had been applied on its inputs. The aim of the article was to determine a suitable combination of the two methods to determine F0 for classifying human emotional state.

Acknowledgments. This work was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070) and by the Development of human resources in research and development of latest soft computing methods and their application in practice project (CZ.1.07/2.3.00/20.0072) funded by Operational Programme Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic.

References

1. Nicholson, J., Takahashi, K., Nakatsu, R.: Emotion Recognition in Speech Using Neural Networks. Neural Computing & Applications 9(4), 290–296 (2006)
2. Partila, P., Voznak, M., Mikulec, M., Zdralek, J.: Fundamental Frequency Extraction Method using Central Clipping and its Importance for the Classification of Emotional State. Advances in Electrical and Electronic Engineering 10(4), 270–275 (2012)
3. Psutka, J., Muller, L., Smidl, L.: Feature space reduction and decorrelation in a large number of speech recognition experiments. In: Proc. the 9th IASTED International Conference on Signal and Image Processing, SIP 2007, Honolulu, pp. 158–161 (2007)
4. Rabiner, L.: On the use of autocorrelation analysis for pitch detection. IEEE Transactions on Acoustics, Speech and Signal Processing 25(1), 24–33 (1977)
5. Kasi, K., Zahorian, S.A.: Yet Another Algorithm for Pitch Tracking. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 1, pp.I-361–I-364. IEEE (2002)
6. Sun, X.: Pitch Determination and Voice Quality Analysis Using Subharmonic-To-Harmonic Ratio. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 333–336. IEEE (2002)

7. Gerhard, D.: Pitch Extraction and Fundamental Frequency: History and Current Techniques. University of Regina, Regina (2003)
8. Solé-Casals, J., Martí-Puig, P., Reig-Bolaño, R., Zaiats, V.: Score Function for Voice Activity Detection. In: Solé-Casals, J., Zaiats, V. (eds.) NOLISP 2009. LNCS, vol. 5933, pp. 76–83. Springer, Heidelberg (2010)
9. Picone, J.W.: Signal modeling techniques in speech recognition. Proceedings of the IEEE 81(9), 1215–1247 (1993)
10. Beale, M., Howard, B., Hudson, M.: Neural network design. Campus Publ. Service, Boulder (2002)
11. Roussinov, D., Chen, H.: A Scalable Selforganizing Map Algorithm for Textual Classification: A Neural Network Approach to Thesaurus Generation. Communication Cognition and Artificial Intelligence (1998)
12. Snášel, V., Húsek, D., Frolov, A.A., Řezanková, H., Moravec, P., Polyakov, P.: Bars Problem Solving - New Neural Network Method and Comparison. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) MICAI 2007. LNCS (LNAI), vol. 4827, pp. 671–682. Springer, Heidelberg (2007)

Predictive Control of Radio Telescope Using Multi-layer Perceptron Neural Network

Sergej Jakovlev, Miroslav Voznak, Kestutis Ruibys, and Arunas Andziulis

Abstract. Radio telescope (RT) installations are highly valuable assets and during the period of their service life they need regular repair and maintenance to be carried out for delivering satisfactory performance and minimizing downtime. With the growing automation technologies, predictive control can prove to be a better approach than the traditionally applied visual inspection policy and linear models. In this paper, Irbene Radio telescope RT-16 disk rotation control motors are analysed. Retrieved data from the small DC motor is used for the predictive control approach. A Multilayer Perceptron (MLP) network approach is used for prediction of the indicator voltage output which affects the monitoring of the disk rotating angle.

Keywords: Predictive control, multi-layer perceptron, neural network, data processing.

1 Introduction

Modernisation of old RTs is a serious mechanical and software problem. In this case, RT-16 Radio telescope in Irbene, Latvia. RT-16 (16 meters in diameter) was built in 1964 and was in operation till 1990‘s (see Fig. 1). In 1999, some basic control mechanisms and renovated DC engines were installed. These motors are suited for the control of RT disk rotation. The estimation of the rotation angle is done using motor speed indicator voltage parameters. Despite the fact that the position of the disk is changing very slowly during the monitoring of stars and space debris, evaluation of the coordinates, in real time, and their prediction is a

Sergej Jakovlev · Kestutis Ruibys · Arunas Andziulis
Klaipeda University, Department of Informatics Engineering, Bijunu str. 17, LT-91225, Klaipeda, Lithuania
e-mail: {s.jakovlev.86,kruibys,arunas.iik.ku}@gmail.com

Miroslav Voznak
VSB-Technical University of Ostrava, Department of Telecommunications,
17. Listopadu 15/2172, 708 33 Ostrava-Poruba, Czech Republic
e-mail: miroslav.voznak@vsb.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 233–244.
DOI: 10.1007/978-3-319-00542-3_24

serious computation problem, as the amount of data is huge. Several minutes of observation can account for several terabytes of stored data.

The huge amount of statistical data is transferred not only from the positioning technologies, but also from the associated sensors. Additional „noise“ can cause the required signal to be lost. Therefore, control algorithms are developed to react to these changing situations and to store data. These tools control the disk smaller rotating and lifting DC engines (see Fig. 2). The control is done using simple switch desktop so there is no agility in control. Therefore, simple computer based prediction of DC motor speed indicator voltage output can be used to minimize the time for disk rotation and limit unwanted mechanical corrections. Such predictive control tool is a valuable addition to the existing systems in RT-16. Received parameters can be used not only for control, but also for maintenance. The equipment is quite old and need constant monitoring and repair. If this tool can predict voltage fluctuation that has a direct impact to the rotation speed, then same data can be used to signal that a malfunction may occur.

Fig. 1 Irbene RT-16 (picture taken in 2012)

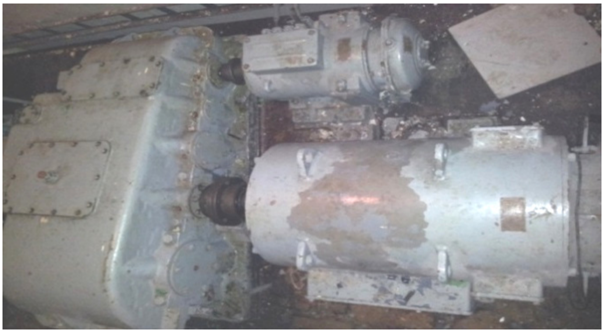


Fig. 2 RT-16 gearbox with small and big disk rotating engines

Predictive control of the machinery is linked to predictive maintenance [1-3]. RT installations are highly valuable assets and during the period of their service life they need regular repair and maintenance to be carried out for delivering satisfactory performance and minimizing downtime. Currently, visual maintenance expenses with human interaction account for a considerable amount of total operating expenditure. As the equipment begins to fail, it may display signs that can be used to identify the onset of equipment degradation and potential failures in control. Metal corrosion can cause the rotating parts of the disk to fail and to work with a high margin of error.

2 Linear and Neural Network Models

The system uses several technologies. In this paper, we present the multi-layer perceptron neural network approach to analyse the working condition of small DC motor. A neural network was used for the prediction of the voltage fluctuation for disk rotating. This fluctuation can have a serious effect on the working stability and measurement results. These fluctuations occur when the disk of the RT is controlled and its movement is constantly corrected. Correction is done using DC motors. But due to the poor condition of the installation, some grid noise and other deviations in the voltage output measurements occur (see Fig. 3).

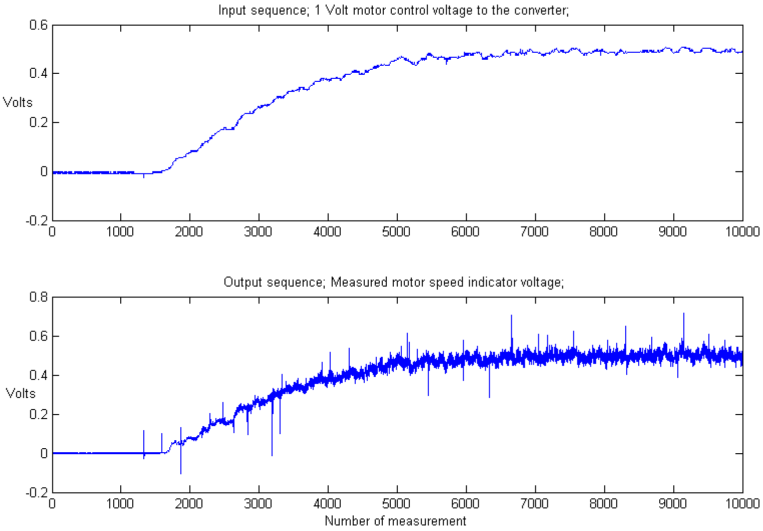


Fig. 3 Comparison between control voltage and small motor speed indicator voltage

1 Volt control voltage was presented to the converter for the small DC motor. When control and indicator values are fluctuating around 0 Volts is the period of time when the engines are not manually launched from the control desktop. The

same control process is performed constantly when disk rotation is necessary. Figure 4 presents data from a different rotation process that followed the one in Fig. 3. Next we determine the model order, i.e., the number of past signals used as regressors. The indices for system orders from 1 to 8 were investigated and the order of the system and its regressors were determined (see Fig. 5 and 6).

It is difficult to conclude anything certain from Fig. 5 and 6, which probably has to do with the noise corrupting the measurements. It is, however, not unreasonable to assume that the system can be modelled by a fifth order model since the slope of the curve is decreasing for model orders >= 5.

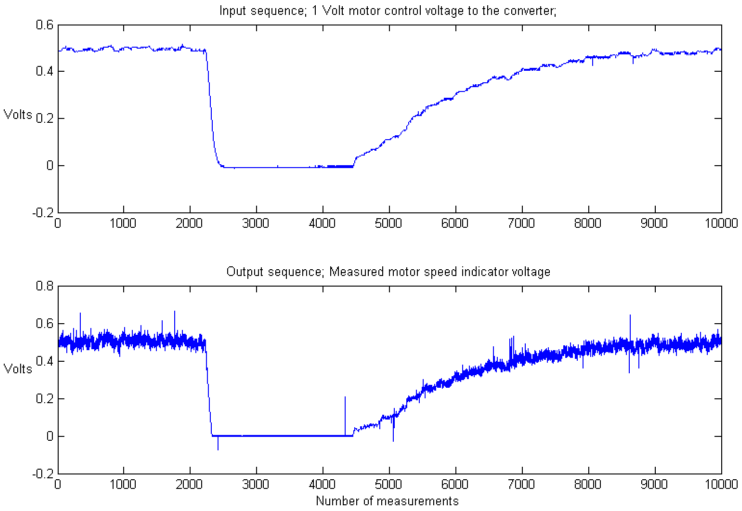


Fig. 4 Comparison between control voltage and small motor speed indicator voltage (second)

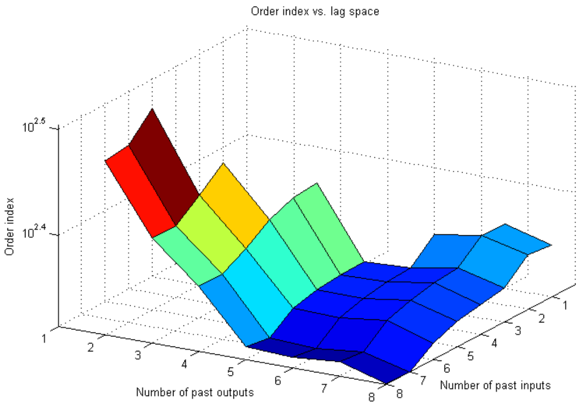


Fig. 5 Observation of the order of the system with number of past inputs and past outputs

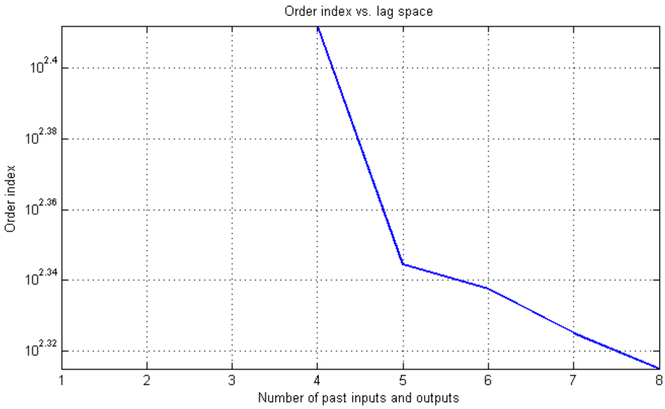


Fig. 6 Observation of the order of the system

2.1 Linear Model

Next, we tried to apply a Linear Model. If it does a good job, one should not bother wasting time on neural network based model structures like MLP. We identified a linear OE-model (Discrete-time IDPOLY model). The prediction is the smooth curve while the noisier one represents the measurements (see Fig. 7).

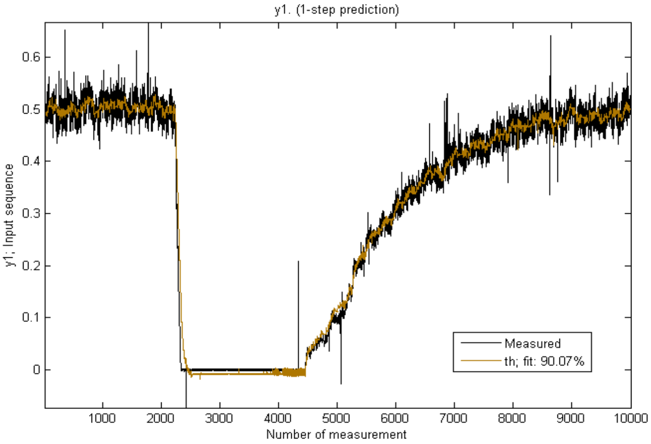


Fig. 7 Linear Model fit

Next we validated an estimated linear model, by displaying the auto correlation function of the residuals and the cross correlation function between input and residuals. From Fig. 7 and 8 we can assume that the linear model has many problems regarding its accuracy, especially for large magnitudes. Therefore a conclusion is made that this is due to the underlying system being nonlinear.

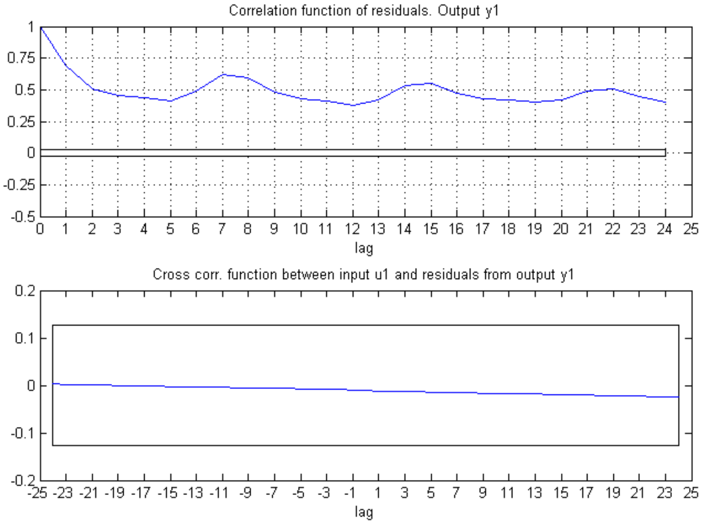


Fig. 8 Model correlation plot

Review of Multilayer Perceptron Network

In this paper we use an MLP network that has only one hidden layer and only hyperbolic tangent and linear activation functions (f, F) in the form of:

$$\widehat{y}_i(w,W) = F_i\left(\sum_{j=0}^{q} W_{ij} f_j\left(\sum_{l=1}^{m} w_{jl} z_l + w_{j0}\right) + W_{i0}\right) \tag{1}$$

The weights (specified by the vector θ and alternatively by the matrices *w* and *W*) are the adjustable parameters of the network, and they are determined through network training. The training data is a set of inputs u(t), and corresponding desired outputs, *y(t)*. The training set specified by (2):

$$Z^N = \{[u(t), y(t)] | t = 1,..., N\} \tag{2}$$

The objective of training is then to determine a mapping from the set of training data to the set of possible weights:

$$Z^N \to \widehat{\theta}. \tag{3}$$

The network will produce predictions $\widehat{y}(t)$, which in some sense are close to the true outputs $y(t)$. The prediction error approach, which is the strategy applied here, is based on the introduction of a measure of closeness in terms of a mean square error criterion (4):

$$V_N\left(\theta, Z^N\right) = \frac{1}{2 \cdot N} \sum_{t=1}^{N} \left[y(t) - \hat{y}(t|\theta)\right]^T \left[y(t) - \hat{y}(t|\theta)\right] \tag{4}$$

The weights are (5):

$$\hat{\theta} = \arg\min_{\theta} V_N\left(\theta, Z^N\right). \tag{5}$$

By iterative minimization scheme (6):

$$\theta^{(i+1)} = \theta^{(i)} + \mu^{(i)} f^{(i)}. \tag{6}$$

Here: $\theta^{(i)}$ specifies the current iterate (number 'i'), $f^{(i)}$ is the search direction, and $\mu^{(i)}$ the step size. The network was trained with Basic Levenberg-Marquardt method as it is a very popular curve-fitting algorithm used in many software applications for solving generic curve-fitting problems [4], [5], [7]. It was used for minimization of mean-square error criteria, due to its rapid convergence properties and robustness [6]. Some improvements were made to the method (7, 8). The size of the elements of the diagonal matrix added to the Gauss-Newton Hessian was adjusted according to the size of the ratio between actual decrease and predicted decrease.

$$r^{(i)} = \frac{V_N\left(\theta^{(i)}, Z^N\right) - V_N\left(\theta^{(i)} + f^{(i)}, Z^N\right)}{V_N\left(\theta^{(i)}, Z^N\right) - L^{(i)}\left(\theta^{(i)} + f^{(i)}\right)}, \tag{7}$$

$$L\left(\theta^{(i)} + f\right) = V_N\left(\theta^{(i)}, Z^N\right) + f^T G\left(\theta^{(i)}\right) + \frac{1}{2} f^T R\left(\theta^{(i)}\right) f. \tag{8}$$

Here: *G* denotes the gradient of the criterion with respect to the weights and *R* is the so-called Gauss-Newton approximation to the Hessian. The following algorithm was used:

1. Select an initial parameter vector $\theta^{(0)}$ and an initial value $\lambda^{(0)}$;
2. Determine the search direction from $\left[R\left(\theta^{(i)} + \lambda^{(i)} \cdot I\right)\right] \cdot f^{(i)} = -G\left(\theta^{(i)}\right)$, I being a unit matrix;
3. $r^{(i)} > 0.75 \Rightarrow \lambda^{(i)} = \lambda^{(i)} / 2$ (If predicted decrease is close to actual decrease let the search direction approach the Gauss-Newton search direction while increasing step size);
4. $r^{(i)} < 0.25 \Rightarrow \lambda^{(i)} = 2 \cdot \lambda^{(i)}$ (If predicted decrease is far from the actual decrease let the search direction approach the gradient direction while decreasing step size);
5. If $V_N\left(\theta^{(i)} + f^{(i)}, Z^N\right) < V_N\left(\theta^{(i)}, Z^N\right)$ then accept $\theta^{(i+1)} = \theta^{(i)} + f^{(i)}$ as a new iterate and let $\lambda^{(i+1)} = \lambda^{(i)}$ and *i=i+1*;
6. If the stopping criterion is not satisfied go to 2).

Next step is to select structure of the model. It is necessary to choose a set of regressors and to determine network architecture. The idea is to select the regressors based on the linear system and then determine the network architecture with the given regressors as inputs. In this work the following regressors and the predictor vectors are used:

$$\varphi(t)=\left[\hat{y}(t-1|\theta)\ldots\hat{y}(t-n_a|\theta)u(t-n_k)\ldots u(t-n_b-n_k+1)\right]^T, \tag{9}$$

$$\hat{y}(t|\theta)=a(\varphi(t),\theta). \tag{10}$$

Here: t specifies sampling instant number t, $\varphi(t)$ is a vector containing the regressors, θ is a vector that contains the weights and g is the function realized by the neural network, n_a, n_b, n_k are used in order for the function to be able to determine the structure of the regression vector used in multi-output and multi-input systems.

2.3 Neural Network Model

Firstly, we selected a fully connected network architecture with 10 hidden hyperbolic tangent units. The network was trained with 1000 iterations and with small weight decay. The function initialized the weights. The influence of the unknown initial conditions was also reduced. Next, validation of the trained network was performed. The following results were achieved (see Fig. 9 and 10).

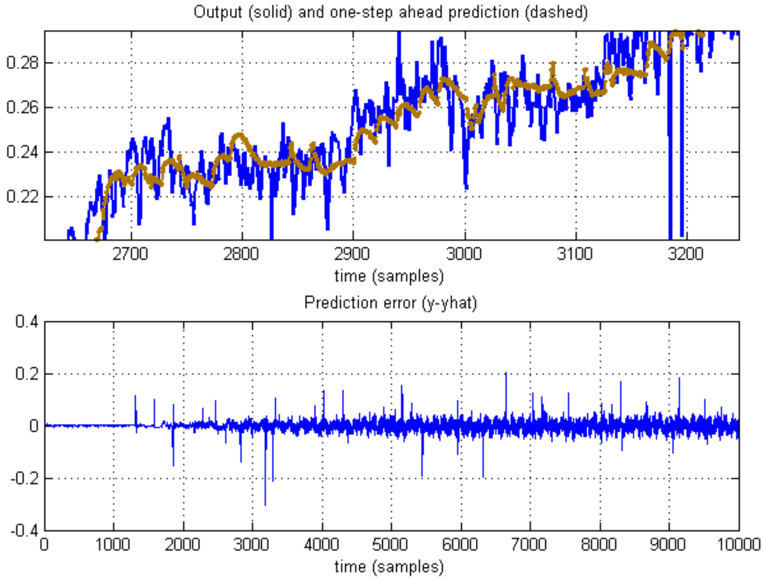


Fig. 9 Neural network model fit

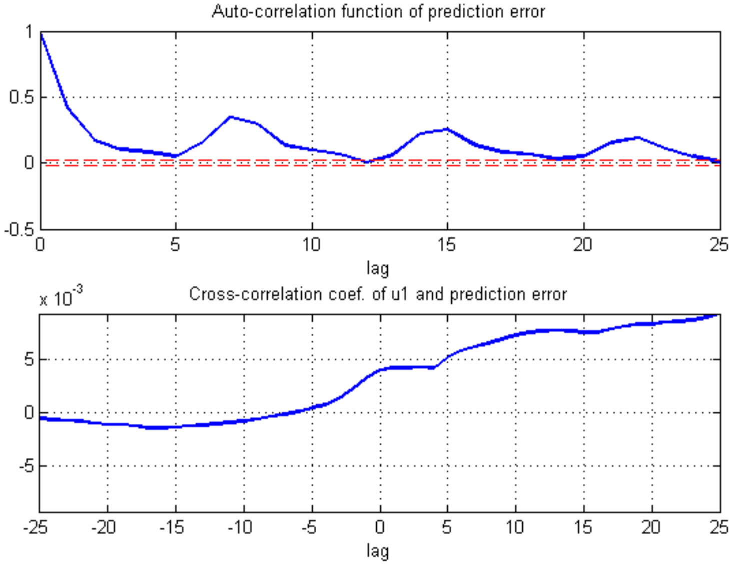


Fig. 10 Model correlation plot

The achieved results are definitely better than the Linear Model. The validation on the test data gave a less positive result (see Fig. 11 and 12).

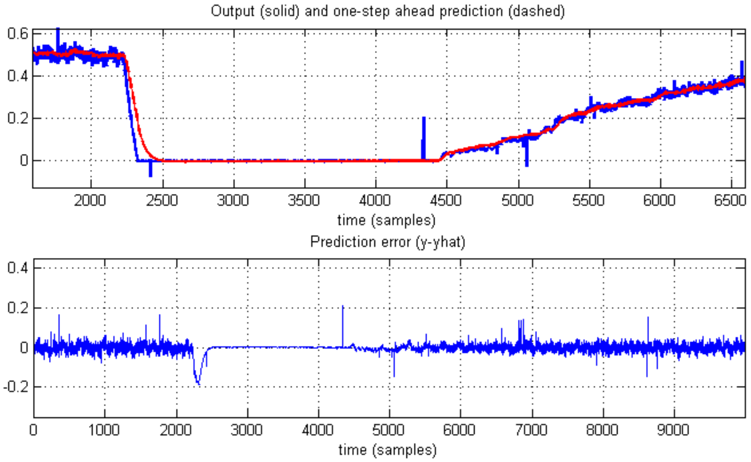


Fig. 11 Neural network model fit

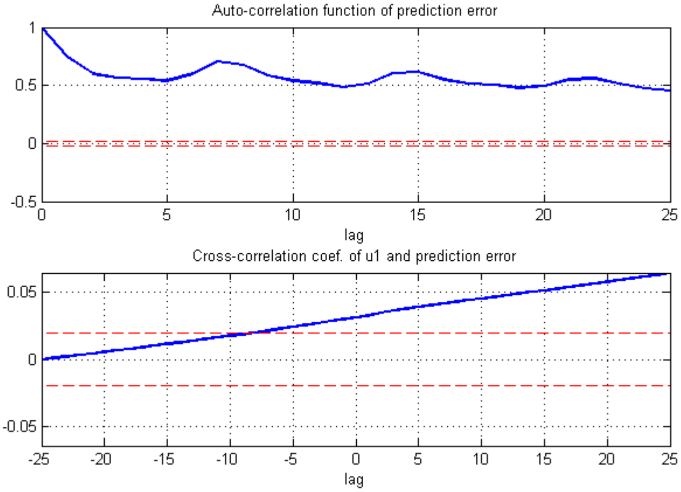


Fig. 12 Model correlation plot

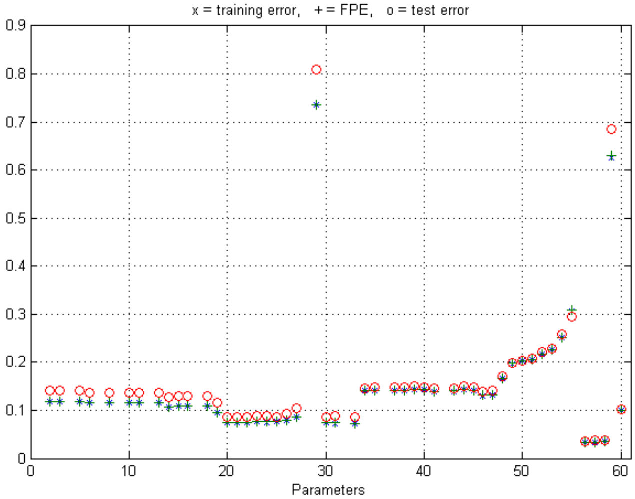


Fig. 13 Wright's analysis

By comparing the plots for training and test set, we can see that the network is over-fitting the data at a certain degree. Therefore, the selected model structure contains too many weights and the in future applications it should be decreased. In order to remove the unwanted weights and to determine the optimal network architecture, we used a network pruning technique. See Fig. 13 for more detailed results. Figure 13 displays training error, test error, and FPE estimate of the generalization error of each of the intermediate networks. This figure shows that the minimum of the test error occurs when there are only 56 weights left in the network. Ideally one would expect the training error to increase monotonically as the weights are pruned, while FPE estimate and test error should decrease until a certain minimum, and the start increasing as well. However, the final results depend on the initial local minimum and the initial weights.

3 Conclusions

We have analysed a complex problem of Radio telescope disk rotation predictive control. The rotation speed was determined by the small DC motor rotation speed indicator voltage and other statistical data was gathered from the control voltage from the converter. Data was used to predict 1-step ahead the voltage output, thus disk rotation speed. Two models were proposed and compared using the following Intel i5-2410M 2.3 GHz processor. Linear Model proved to be >90% effective with the predictions and the MLP neural network proved to be more precise in predictions. Therefore, computation speed was also compared. 8th order computation for the Linear Model took 8 minutes to complete, thus it is useless in the real life application. On the other hand, the newly trained neural network will perform much quicker by using less hidden neurons. During the experiment, different network structures were examined with a different number of hidden neurons. Results suggest that a network with the number of neurons >5 can perform much better than the Linear Model. Computation of this network takes only a few seconds. Thus it can be used in real life applications and in RT-16.

Acknowledgments. This work was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070) and by the Development of human resources in research and development of latest soft computing methods and their application in practice project (CZ.1.07/2.3.00/20.0072) funded by Operational Programme Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic, also this work has been 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.

References

1. Savran, A., Tasaltin, R., Becerikli, Y.: Intelligent adaptive nonlinear flight control for a high performance aircraft with neural networks. ISA Transactions 45(2), 225–247 (2006)
2. Vasickaninova, A., Bakosova, M., Meszaros, A., Klemes, J.J.: Neural network predictive control of a heat exchanger. Applied Thermal Engineering 31(13), 2094–2100 (2011)

3. Vahidinasab, V., Jadid, S., Kazemi, A.: Day-ahead price forecasting in restructured power systems using artificial neural networks. Electric Power Systems Research 78(8), 1332–1342 (2008)
4. Mukherjee, I., Routroy, S.: Comparing the performance of neural networks developed by using Levenberg–Marquardt and Quasi-Newton with the gradient descent algorithm for modelling a multiple response grinding process. Expert Systems with Applications 39(3), 397–2407 (2012)
5. Piotrowski, A.P., Napiorkowski, J.J.: Optimizing neural networks for river flow forecasting – Evolutionary Computation methods versus the Levenberg–Marquardt approach. Journal of Hydrology 407(1-4), 12–27 (2011)
6. Kermani, B.G., Schiffman, S.S., Troy Nagle, H.: Performance of the Levenberg–Marquardt neural network training method in electronic nose applications. Sensors and Actuators B: Chemical 110(1), 13–22 (2005)
7. Snášel, V., Húsek, D., Frolov, A.A., Řezanková, H., Moravec, P., Polyakov, P.: Bars Problem Solving - New Neural Network Method and Comparison. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) MICAI 2007. LNCS (LNAI), vol. 4827, pp. 671–682. Springer, Heidelberg (2007)

Modeling and Simulation of a Small Unmanned Aerial Vehicle

Ozan Eren Yuceol and Ahmet Akbulut

Abstract. This paper aims to demonstrate a design of low-cost, and Inertial Measurement Unit (IMU) based autopilot that is ArduPilot, and its applications. ArduPilot's control algorithm possesses two control loops, which are used to control the navigation and the control surfaces such as actuators. Due to using PID control and being able to tune and update the PID parameters (Kp, Ki, Kd) even during the flight, ArduPilot becomes very efficient autopilot also. It's also possible to export and analyze the data such as airspeed, air pressure, latitude-longitude-altitude, roll-pitch-yaw angles, accelerometer-gyro information, servo outputs, and moreover on Matlab®, after or during the flight. In order to go one step further, a model aircraft, which has nearly the same physical parameters as the one used for the real tests, designed on Plane Maker®, and simulated on Xplane® simulator. Using the simulator made this study easy to tune and test the whole flight parameters before real flight. This paper will lead the students who are interested in automatic control applications especially Unmanned Aerial Vehicles (UAV).

Keywords: UAV, ArduPilot, automatic control systems, PID control, IMU, direction cosine matrix, Xplane®.

1 Introduction

Unmanned aerial vehicles have been started to be used in several areas, and commercial ones are able to be had under $500. In our century, aviation became very important for several areas such as military, transportation, security, and scientific researches. Because of that reason, it has been required to have more efficient and safer aerial vehicles. This idea forced the manufacturers and developers to made autonomous subsystems, and made this subsystems depended on the sensors [2], [3].

By development of the microprocessors in last two decades, usage of unmanned aerial vehicles have been accelerated firstly for the military use, and then commercial even hobbyist use also. Although these are quite complex systems, the

Ozan Eren Yuceol · Ahmet Akbulut
Ankara University, Electrical & Electronics Engineering Department
Besevler, 06100 Ankara, Turkey
e-mail: ozanyuceol@hotmail.com, aakbulut@ankara.edu.tr

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 245–255.
DOI: 10.1007/978-3-319-00542-3_25

main idea depends on simple theories. One of the best theories is PID control, which is an error processing based application. PID control is used in wide range applications from chemistry to aerospace. Moreover, analysis results are quite easy to understand for human mind, because it's relied on following the input of the system. The faster you follow and stand on the input value, the more efficient system you will have. So, one of the most important goals of this study is making the system, which is already easy to find, much safer, faster, and lesser error.

2 Overview of PID Control

PID control is one of the most frequently used automatic control type for a regular system. The main reason is that it's possible to apply this method both on to the digital computers numerically, and the analog computers mathematically. By development of the digital computers, analog ones, such as Op-Amps, became less to be required. Hence, today it's quite easy to construct a PID controller in a few steps by using high speed computers. Basically the algorithm depends on some manipulations on the error or in other words the difference between input and output. In order to reach that aim, sensors are used while observing the output of the system, and then the output of the sensor is compared with the reference input. The comparison is exactly the same as error that is handled by the PID controller, as seen in Fig.1.

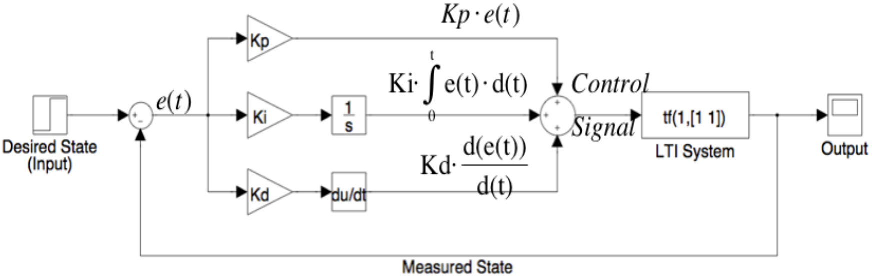


Fig. 1 Basic PID control loop scheme on Simulink®

While sensors and the other electronic devices are operating, less or much rounding error happens, because they operate in discrete time. Naturally, there will be errors during sampling and quantizing the analog signal. In same cases, rounding error can cause the system to be forced to unacceptable operating points, which may make the system unstable.

3 Direction Cosine Matrix (DCM)

Because body is moving by time, it's required to use one reference coordinate system while computing the orientation. In order to achieve this problem, a matrix, which is named as "Rotation Matrix" [1], [5], is used. It describes the orientation of one coordinate system with respect to another. A vector in one system can be transformed into the other system by multiplying it by the rotation matrix, as seen in Fig. 2 and Fig. 3. Because of its orthogonality, it's quite easy to calculate the inverse Rotation Matrix in order to turn back from second coordinate system to first one again. This matrix has 9 elements, and only 3 of them are independent, because of being antisymmetric.

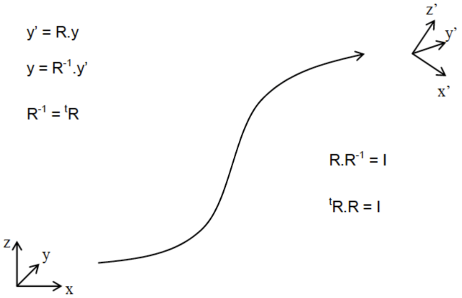


Fig. 2 Carrying a vector to another space by using the rotation matrix

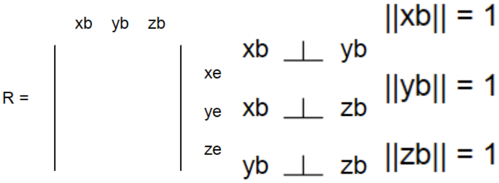


Fig. 3 Properties of the rotation matrix

Basically, R matrix is used to control the roll and pitch, navigate, find out if the aircraft is upside down, and find out the turning rate of the aircraft around the vertical axis, as seen in Fig. 4 and Fig. 5.

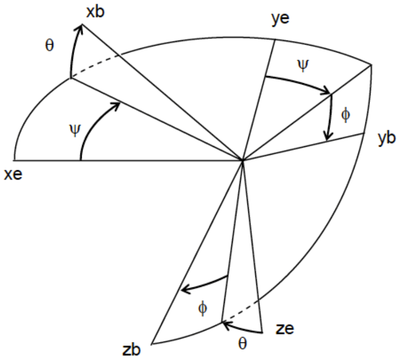


Fig. 4 Euler angles for the reference space (earth), and aircraft body

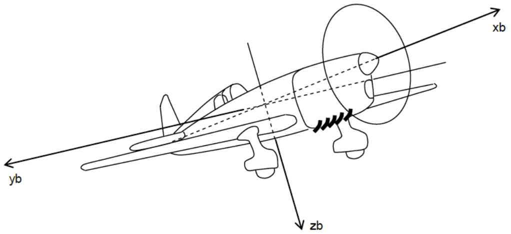


Fig. 5 Euler angles on the aircraft body (roll, pitch, yaw: xb, yb, zb)

3.1 Using DCM in Control and Navigation

Before explaining the attitudes of the axes, the rotations from one axis to another should be defined. In order to do this, the elements of the DCM can be expanded by using the right hand rule as shown following.

$$R = \begin{bmatrix} rxx & rxy & rxz \\ ryx & ryy & ryz \\ rzx & rzy & rzz \end{bmatrix} \tag{1}$$

A rotation can be found as the following expressions;

$$\begin{aligned} &\therefore rxx = \hat{x} \cdot \hat{x} = 1 \\ &\therefore rxz = \hat{x} \cdot \hat{z} = 0 \cong d\theta y \\ &\therefore ryz = \hat{y} \cdot \hat{z} = 0 \cong = d\theta z \end{aligned} \tag{2}$$

All elements in the DCM represent a rotation from one axis to another one. For instance, refers a rotation around the y-axis, because of the right hand rule in vector algebra. Thus, the rotation matrix changes to such a simple matrix;

$$R = \begin{bmatrix} 1 & -d\theta z & d\theta y \\ d\theta z & 1 & -d\theta z \\ -d\theta y & d\theta z & 1 \end{bmatrix} \tag{3}$$

3.2 Controlling the Pitch

To control the pitch of the aircraft, the pitch attitude is needed to be known. It can be found by taking the dot product of the roll axis of the aircraft with ground vertical, as seen in Fig. 6. So, the direction cosine for pitch attitude is r_{zx}=sin (θ). When the plane is leveled r_{zx} will be zero.

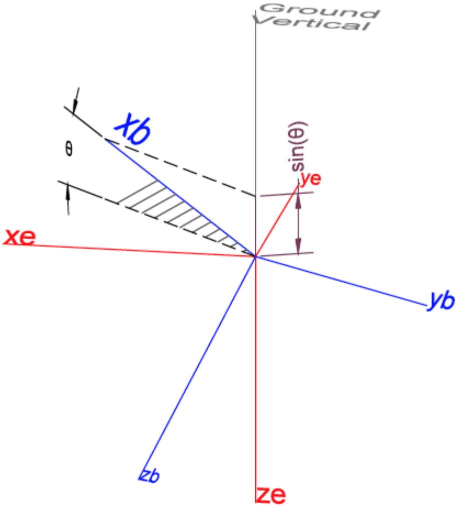


Fig. 6 Finding the pitch attitude

3.3 Controlling the Roll

To control the roll of the aircraft, the bank attitude, which can be found by taking the dot product of the pitch axis of the aircraft with ground vertical, as seen in Fig. 7. So, the direction cosine for pitch attitude is r_{zy}=sin (φ). When the plane is leveled r_{zy} will be zero.

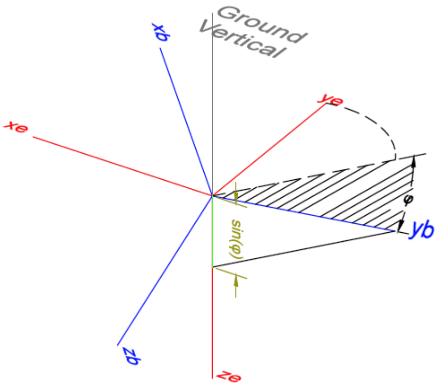


Fig. 7 Finding the roll attitude

3.4 Navigation

To navigate, the yaw attitude with the respect to the direction, which is wanted to be gone must be known. The yaw attitude can be found by taking the cross product of the roll axis of the aircraft with a vector in the direction that is wanted to be gone, as seen in Fig. 8.

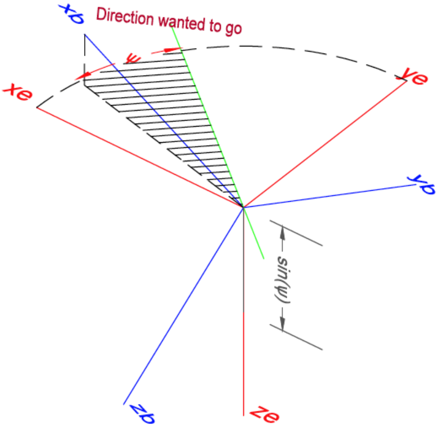


Fig. 8 Finding the yaw attitude

4 Simulations

Before simulating the autopilot and the aircraft that have been used a model which is available to be used on the Xplane®, had been created and simulated with the autopilot, as seen in Fig. 9 and Fig. 10. The main aim was that trying to create a drawing and tune the aerodynamic values are as the same as the airplane that would be used in the real world. The model one is a good copy of Bixler2 radio controlled airplane, which is available on www.hobbyking.com. During the simulations several tests were done, and Megabytes of data processed on Matlab® and Mission planning software, as seen in Fig. 11 and Fig. 12 [4], [5], [6], [7], [9].

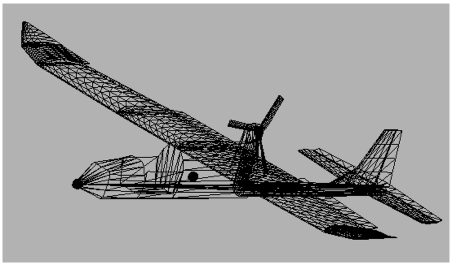


Fig. 9 The model drawen on Plane Maker®

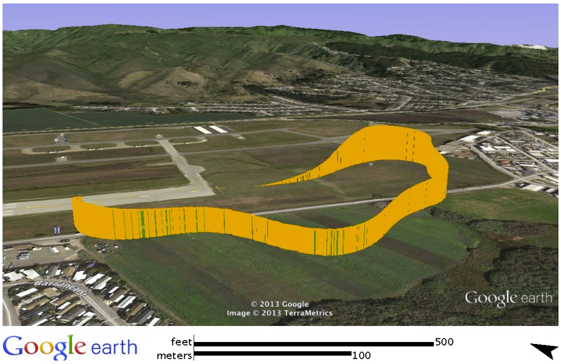


Fig. 10 Simulation's flight path on the Google Earth®

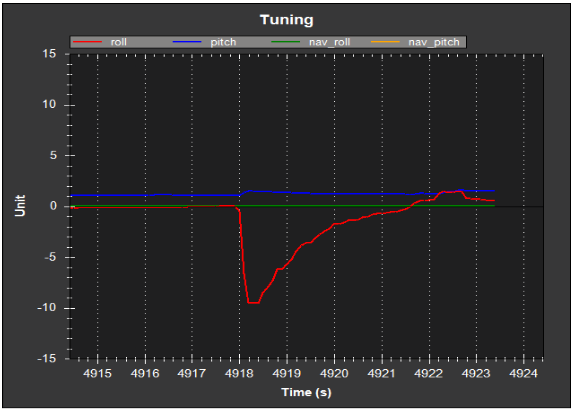
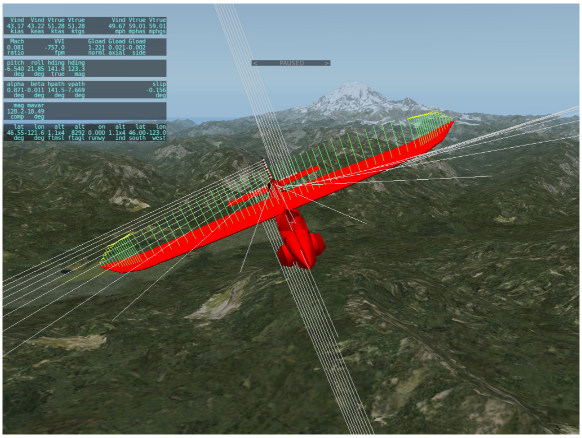


Fig. 11 The unit step response of the roll axis in the simulation

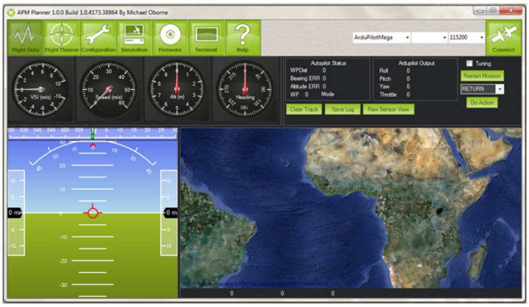


```
 Vind  Vind Vtrue Vtrue        Vind Vtrue Vtrue
43.17 43.22 51.28 51.28       49.67 59.01 59.01
 kias  keas  ktas  ktgs         mph mphas mphgs

 Mach         VVI       Gload Gload Gload
0.081      -757.0       1.221 0.021-0.002
ratio         fpm       norml axial  side

pitch  roll hding hding
-6.540 21.85 141.8 123.3
  deg   deg  true   mag

alpha  beta hpath vpath                   slip
0.871-0.011 141.5-7.669                 -0.156
  deg   deg   deg   deg                    deg

  mag mavar
128.2-18.49
 comp   deg

  lat   lon   alt   alt    on   alt   lat   lon
46.55-121.6 1.1x4  8292 0.000 1.1x4 46.00-123.0
  deg   deg ftmsl ftagl runwy   ind south  west
```

Fig. 12 The model in simulation on the Xplane®

4.1 Autopilot/Mission Planner

A low cost, open source and commercially available autopilot has been selected to use during the research, due to being easy to repair. ArduPilot is an autopilot, which is easy to configure, and update. Moreover, it's possible to send or receive new data, and plan a mission during the operation, even if the main station is thousands of meters away from the airplane by using the telemetry. There are several ground station softwares, which are compatible with ArduPilot, but two of them were used to test and simulate the system at this study. First of them is "ArduPilot Mega Mission Planner (APM)" , which is easy to simulate on Xplane®, plan & update the mission, change the current status of the autopilot by one clicking, and suitable to show exported data such as mission path on Google Earth®, but unfortunately it's not compatible with Mac OS®, as seen in Fig 13.



APM Planner 1.0.0 Build 1.0.4173.38864 By Michael Oborne

ArduPilotMega 115200

Autopilot Status
WPDist 0
Bearing ERR 0
Altitude ERR 0
WP 0 Mode

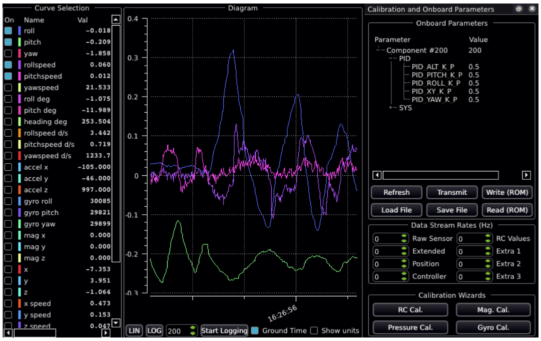
Ardupilot Output
Roll 0
Pitch 0
Yaw 0
Throttle 0

☐ Tuning

RETURN

Fig. 13 ArduPilot Mega Mission Planner Software

Second of them is "Q Ground Control", which is able to give more scientific data than APM to the user, show the values of such flight variables run on the program, and export the whole flight data with values such as roll, pitch, yaw and bearing error etc. to the Matlab®, as seen in Fig 14.



Curve Selection

On	Name	Val
☑	roll	-0.018
☑	pitch	-0.209
☐	yaw	-1.858
☑	rollspeed	0.060
☑	pitchspeed	0.012
☐	yawspeed	21.533
☐	roll deg	-1.075
☐	pitch deg	-11.989
☐	heading deg	253.504
☐	rollspeed d/s	3.442
☐	pitchspeed d/s	0.719
☐	yawspeed d/s	1233.7
☐	accel x	-105.000
☐	accel y	-46.000
☐	accel z	997.000
☐	gyro roll	30085
☐	gyro pitch	29821
☐	gyro yaw	29899
☐	mag x	0.000
☐	mag y	0.000
☐	mag z	0.000
☐	x	-7.353
☐	y	3.951
☐	z	-1.064
☐	x speed	0.473
☐	y speed	0.153
☐	z speed	0.047

Calibration and Onboard Parameters

Onboard Parameters

Parameter	Value
Component #200	200
PID	
PID_ALT_K_P	0.5
PID_PITCH_K_P	0.5
PID_ROLL_K_P	0.5
PID_XY_K_P	0.5
PID_YAW_K_P	0.5
SYS	

Data Stream Rates (Hz)

0	Raw Sensor	0	RC Values
0	Extended	0	Extra 1
0	Position	0	Extra 2
0	Controller	0	Extra 3

200 ☑ Ground Time ☐ Show units

Fig. 14 Qgroundcontrol Software

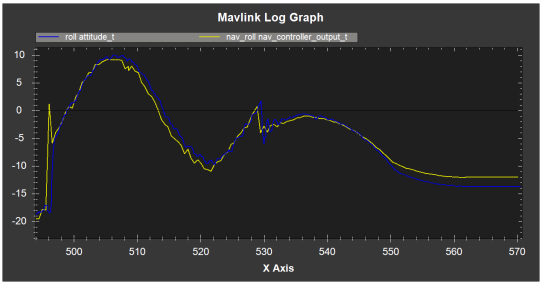


Fig. 15 The roll axis output of the autopilot on the Stewart Plate

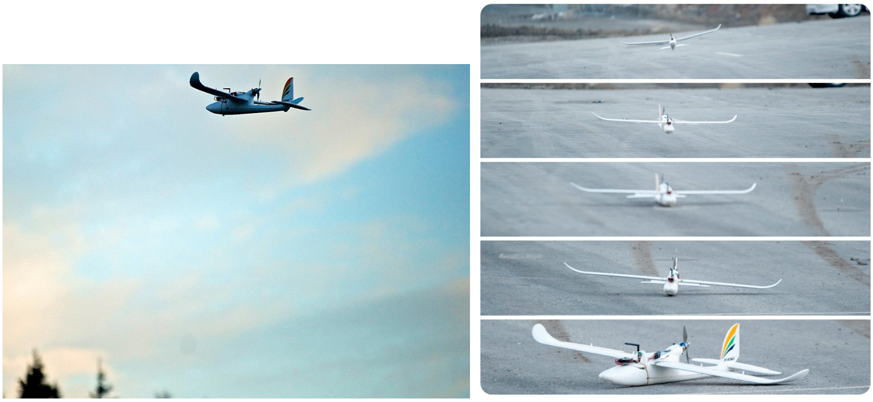


Fig. 16 a) An autonomous flight with Bixler2, b) While doing a successful autonomous landing without any external radio control

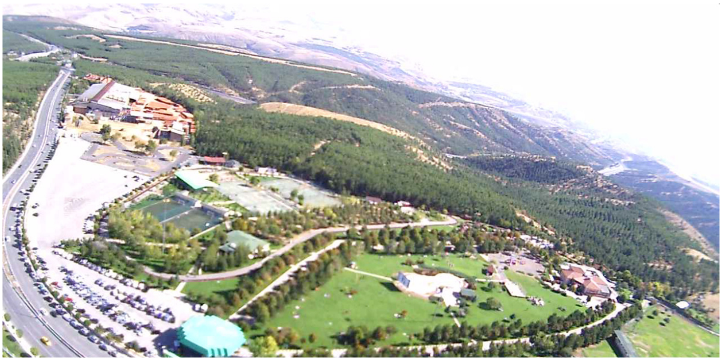


Fig. 17 A frame from the on-board CCD camera

In order to tune the controller to the most proper PID gains, a “Stewart Plate” was designed, and constructed for testing the autopilot in one axis [8]. This plate was quite helpful to understand the effects of the PID gains on the airplane. Moreover, this plate helped to figure out what the boundary values should be, as seen in Fig 15. While trying to approach the best PID gains, Ziegler-Nichols method was used on Matlab®.

By these studies, several test flights have been done with Bixler2 RC plane, as seen in Fig 16, and Fig. 17. The results of the test flights were imported to the Matlab®, and compared the simulation results from the same mission simulated on the Xplane®.

5 Conclusions

A similar model of the Bixler2 RC aircraft has been built to validate the PID tuning for the real tests by using the ArduPilot. A test platform (Stewart plate) for the Ardupilot controller was developed and tested to stabilize itself. According to real test data, the aerodynamic of the simulated model of the Bixler2 has been improved, and corrected. The main aim of that study has been approached via these methods.

It has been understood that modeling, and simulating a system before testing, is quite important to figure out the effects of the critical parameters on the system. Because all these methods had been used during the study, the total period, and cost were reduced.

According to the results of this research, next stages are thought that the number of aircrafts will be increased, and a number of aircrafts will be controlled by only one ground control system at the same time, and simultaneously.

References

1. Rachman, E., Razali, R.: A Mathematical Modeling For Design and Development of Control Laws for Unmanned Aerial Vehicle (UAV). International Journal of Applied Science and Technology 1(4), 66–75 (2011)
2. Noura, H., Bateman, F.: Control of an Unmanned Aerial Vehicle. In: Proceeding of the 7th International Symposium on Mechatronics and its Applications (ISMA 2010), Sharjah, UAE, April 20-22, pp. 1–6 (2010)
3. Nonami, K., Kendoul, F., Suzuki, S., Wang, W., Nakazawa, D.: Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles. Springer, Heidelberg (2010) ISBN 978-4-431-53855-4, e-ISBN 978-4-431-53856-1
4. Mueller, E.R.: Hardware-in-the-loop Simulation Design for Evaluation of Unmanned Aerial Vehicle Control Systems. In: AIAA Modeling and Simulation Technologies Conference and Exhibit, Hilton Head, South Carolina, pp. 6539–6553 (2007)
5. Abdunabi, T.A.M., El-Gelani, M., Nasr, N.: Modeling and Autonomous Flight Simulation of a small Unmanned Aerial Vehicle. In: 13th International Conference on Aerospace Sciences & Aviation Technology, ASAT-13 (2009)

6. Jung, D., Tsiotras, P.: Modelling and Hardware-in-the-Loop Simulation for a Small Unmanned Aerial Vehicle. In: AIAA Infotech at Aerospace, Rohnert Park, CA, AIAA Paper 07-2768, May 7-10 (2007)
7. Wul, J., Wang, W., Zhang, J., Wang, B.: Research of a kind of New UAV Training Simulator Based on Equipment Simulation. In: 2011 International Conference on Electronic & Mechanical Engineering and Information Technology, pp. 4912–4815 (August 2011)
8. Santos, S.R.B., Oliviera, N.M.F.: Test Platform to Pitch Angle using Hardware in Loop. In: 39th ASEE/IEEE Frontiers in Education Conference, San Antonio, TX, W1J1–W1J5 (October 2009)
9. Jin, G.-D., Gu, L.X., Lu, L.-B.: UAV Simulator-Based Simulation of Flight Control System. In: Intelligent Systems and Applications, ISA 2009, pp. 1–4 (2009)

Mathematical Models of Controlled Systems

Vladimír Jehlička

Abstract. The paper is focused on the build of a mathematical model of digital controlled systems. This article describes the creation of mathematical models of one-dimensional systems by the method of experimental identification. The created model is used for predicting the static and dynamic behavior of the controlled system in the closed loop. Dynamic properties of systems are described by the differential equations. In the experimental part are identified the parameters of the mathematical model of rectifying column. As an example, the one-dimensional controlled system, in this case is described the dependence of concentration distilled mixture on change the reflux.

Keywords: Mathematical models, experimental identification, recursive identification, rectification column.

1 Introduction

During the design of control algorithms of technological processes and in many other cases [1-5], it is necessary to know the static and dynamic characteristics of controlled systems. It is necessary to create a mathematical model of the controlled system, on the basis of which it would be possible to predict the behavior of the system in the closed loop. The mathematical model of the system can be created on the basis of mathematical-physical analysis or by experimental identification.

If model of system is created by the method of mathematical-physical analysis, then the system is mathematically described on the basis of knowledge of relevant physical laws [6]. The internal structure of the model is the same as the structure of the system. The created model describes very accurately controlled system. Possible to create valid models generally quite independently of whether the modeled system actually exists already or yet to be created.

A prerequisite for using this classical approach is the knowledge of all the processes that take place within the modeled system and the possibility of this happening exactly mathematically describe. To specify the parameter values of the model it is necessary that all the required variables are measurable. These

Vladimír Jehlička
University of Pardubice, Jan Perner Transport Faculty,
Department of Informatics in Transport, Studentská 573, 530 10 Pardubice, Czech Republic

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 257–264.
DOI: 10.1007/978-3-319-00542-3_26

assumptions are satisfied only for simple systems. Therefore, when modeling complex systems must be adopted certain simplifying assumptions that impair the quality of the final model [7].

The second group of approaches to the design of control systems are methods that are based on experimental identification systems [8]. They are used primarily for control of linear deterministic or stochastic systems, the model can not be created using mathematical and physical analysis.

Experimental identification of the values of the parameters of the model is based on the evaluation of the response of the system to precisely-defined waveforms of the values of the input variables of the system. The internal structure of the model should be chosen. If it has been previously made at least some mathematical-physical analysis of the controlled system, then it is possible to select the internal structure of the model based on the analysis.

Creation of mathematical models of systems by the method of experimental identification is based on the measurement and evaluation values of the input and output signals of the system. On the basis of the results of the performed identification is then possible to propose suitable algorithms of numerical control of the system. In the experimental identification of the measured data are evaluated, either "off-line", i.e. after the end of the experiment, or continuously, i.e. "online", where the data is processed during the measurement. Identification method of "off-line" is used to identify the system parameters, the static and dynamic properties, are not depending on the time change. On the contrary, for systems whose parameters are not constant and algorithms for adaptive control is needed ongoing identification, i.e., the type of "on-line", which takes place in real time, which is a mathematical model refined and modified according to the current state of the system.

2 Identification of One-Dimensional Systems

Assume one-dimensional linear or in restricted around the operating point linearized deterministic system with lumped constant parameters, which can be using the z-transform equation to describe

$$A\left(z^{-1}\right)Y\left(z\right)=z^{-q}B\left(z^{-1}\right)U\left(z\right) \tag{1}$$

where

Y(z) is the z-image of output signal *y(t)* dependent on the time *t*,

U(z) is the z-image of input signal u*(t)* dependent on the time *t*,

q represents the time delay q sampling interval,

$$\begin{aligned} A(z^{-1}) &= 1 + a_1 z^{-1} + \dots + a_n z^{-n}, \\ B(z^{-1}) &= b_1 z^{-1} + b_2 z^{-1} + \dots + b_m z^{-m}, \end{aligned} \tag{2}$$

n, m are the degrees of polynomials $A(z^{-1})$, $B(z^{-1})$.

The aim is to determine the identification parameters of polynomials $A(z^{-1})$ and $B(z^{-1})$ and possibly the model structure, ie the determination of degrees of polynomials $A(z^{-1})$, $B(z^{-1})$ and the size of the time delay.

For transfer $F(z)$ of the described system, which is defined as the ratio of the image-output variable to the image-input variable

$$F(z) = \frac{Y(z)}{U(z)} \tag{3}$$

then from equation (1) is valid

$$F(z) = \frac{z^{-q} B(z^{-1})}{A(z^{-1})} \tag{4}$$

It should also be noted that equation (1) represents the z-image differential equations

$$A(z^{-1})\, y(k) = z^{-q} B(z^{-1})\, u(k) \tag{5}$$

where

y(k) is the value of the output variable *y(t)* in the *k*-th step,

u(k) is the value of the input variable *u(t)* in the *k*-th step,

k represents the time value as a multiple of the sampling interval Δt

Writing *y(k)* means the value of the output variables *y(t)* at time $t = k\,\Delta t$ i.e. in *k*-th step. Symbolic notation $z^{-1}y(k)$ is to be understood as the value of the output variable *y(t)* in the previous step, i.e. shifted back in time by one sampling interval, which can be expressed by the equation.

$$z^{-1} y(k) = y(k-1) \tag{6}$$

Likewise

$$z^{-n} y(k) = y(k-n) \tag{7}$$

represents a shift in time on *n* sample intervals.

Another possible way to edit the model (5) lies in the introduction of the coefficient *s*, in which is included a non-zero mean value of the output and the input variables in a given working point.

$$A\left(z^{-1}\right) y(k) = z^{-q} B\left(z^{-1}\right) u(k) + s \tag{8}$$

Static characteristics of the model (8) does not pass beginning of u-y coordinate system, but intersects the y-axis at a non-zero distance from the beginning and is described by the equation

$$\overline{y} = \frac{\sum_{i=1}^{m} b_i}{1 + \sum_{i=1}^{n} a_i}\,\overline{u} + \frac{s}{1 + \sum_{i=1}^{n} a_i} \tag{9}$$

These types of models more or less well approximate the identified system and for the measured data are valid for a certain error er(k), called the error equation.

$$A\left(z^{-1}\right) y(k)=z^{-q}B\left(z^{-1}\right)u(k)+s+e_r(k) \tag{10}$$

3 Non-recursive Identification Methods

For the estimation of the parameters of the models in the form of differential equations was proposed many different non-recursive identification methods, but practice has shown that usually entirely sufficient basic method that minimizes the sum of squares of errors equation

$$\sum_{k=n+q+1}^{N} e_r^2(k) \rightarrow \min. \tag{11}$$

i.e. the method of least squares.

Assume: a system is described by the model (10), the structure of the model is known, degree polynomials A(z-1) and B(z-1) are the same, i.e. n = m.

Then for N measured values the input and output variables, you can write

$$\boldsymbol{y}=\boldsymbol{F\gamma}+\boldsymbol{e}_r \tag{12}$$

where

$$\boldsymbol{y}=\left(y(n+q+1) \quad y(n+q+2) \quad \dots \quad y(N)\right)^{\mathrm{T}}$$

$$\boldsymbol{e}_r=\left(e_r(n+q+1) \quad e_r(n+q+2) \quad \dots \quad e_r(N)\right)^{\mathrm{T}}$$

$$\boldsymbol{\gamma}=\left(a_1 \quad a_2 \quad \dots \quad a_n \quad b_1 \quad b_2 \quad \dots \quad b_n \quad s\right)^{\mathrm{T}}$$

There are column vectors, and F is a matrix type (N-n-q, 2n+ 1).

$$\boldsymbol{F}=\begin{pmatrix} -y(n+q) & \dots & -y(q+1) & u(n) & \dots & u(1) & 1 \\ -y(n+q+1) & \dots & -y(q+2) & u(n+1) & \dots & u(2) & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -y(N-1) & \dots & -y(N-n) & u(N-q-1) & \dots & u(N-q-n) & 1 \end{pmatrix}$$

After substituting equation error $\boldsymbol{e}_r$ from (12) to (11) we obtain the relation, the minimization of

$$(\boldsymbol{y}-\boldsymbol{F\gamma})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{F\gamma}) \rightarrow \min. \tag{13}$$

We obtain a condition for the existence of a minimum of the sum of squared errors of the equation $\boldsymbol{e}_r$

$$\boldsymbol{F}^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{F\gamma})=\boldsymbol{0} \tag{14}$$

After the treatment we receive the relationship for the estimation of $\hat{\boldsymbol{\gamma}}$ the parameters identified by the method of least squares

$$\hat{\boldsymbol{\gamma}} = \left(\boldsymbol{F}^{\mathrm{T}}\boldsymbol{F}\right)^{-1}\boldsymbol{F}^{\mathrm{T}}\boldsymbol{y} \tag{15}$$

This equation is solvable if and only if the matrix $\boldsymbol{F}^{\mathrm{T}}\boldsymbol{F}$ is a regular.

4 Recursive Identification Methods

Recursive identification methods do not process all the measured data at once, but gradually. Unlike from non-recursive methods that can be used only in the mode of "off-line" can be a recursive method used in mode "on-line", when the parameters of the system are identified continuously in real time. New estimates $\hat{\boldsymbol{\gamma}}_{N+1}$ of the parameters in step $N+1$ are calculated from previous estimates $\hat{\boldsymbol{\gamma}}_N$ calculated in step N by adding corrections $\boldsymbol{\Gamma}_{N+1}$.

$$\hat{\boldsymbol{\gamma}}_{N+1} = \hat{\boldsymbol{\gamma}}_N + \boldsymbol{\Gamma}_{N+1} \tag{16}$$

The calculation of the correction must be simple and is executed before to measure the next pair of values in the input u(N+ 1) and the output y(N+ 1). In identifying the chemical technological processes this condition is met. The time required for the calculation of the correction is usually negligible compared with the time constants of identified processes, which are usually expressed in the order of minutes to tens of minutes.

Recursive identification methods were derived many, but in practice it turned out that the greatest application is again the least squares method.

In step N+ 1 is measured the values of the output $y(N+1)$ and the input $u(N+1)$ variables and increases the dimensions of the vectors **y** and $\mathbf{e}_r$ and the matrixs $\boldsymbol{F}$ expands on the line.

$$\boldsymbol{f}_{N+1} = \begin{pmatrix}-y(N) & \dots & -y(N-1) & u(N-q) & \dots & u(N+1-q-n) & 1\end{pmatrix} \tag{17}$$

A simple way can be derived from equation (15) the least-squares methods algorithm for continuous parameter estimation, which is described by equations (18) to (20)

$$\hat{\boldsymbol{\gamma}}_{N+1} = \hat{\boldsymbol{\gamma}}_N + \boldsymbol{m}_N\left\lfloor y(N+1) - \boldsymbol{f}_{N+1}^{\mathrm{T}}\hat{\boldsymbol{\gamma}}_N\right\rfloor \tag{18}$$

where

$$\boldsymbol{m}_N = \boldsymbol{P}_N\boldsymbol{f}_{N+1}\left(1 + \boldsymbol{f}_{N+1}^{\mathrm{T}}\boldsymbol{P}_N\boldsymbol{f}_{N+1}\right)^{-1} \tag{19}$$

To calculate the matrix pays recurrent relationship

$$\boldsymbol{P}_{N+1} = \boldsymbol{P}_N - \boldsymbol{m}_N\boldsymbol{f}_{N+1}^{\mathrm{T}}\boldsymbol{P}_N \tag{20}$$

In the application of the algorithm, it is necessary to solve these problems:

- enter the starting values $\hat{\boldsymbol{\gamma}}_0$ and $\boldsymbol{P}_0$,
- the numeric stability of the algorithm,

- the adaptation of the model when changes to the properties identified by the system,
- the stability of the algorithm for the case that the system is not sufficiently excited, i.e. if do not change the values of the input and output variables. This condition can occur in a closed loop when the controlled variables is fixing.

5 Choice of the Parameters of the Input Signal

If this technological conditions allow, it is desirable to perform experimental identification of the input identified by the system is fed a signal with a precisely defined parameters. Typically, this is the sequence of the pseudorandom binary signal (PRBS), for which it is necessary to define the length of the period, amplitude and the size of the sampling interval.

5.1 The Choice of the Period and the Sampling Interval PRBS

In an experimental identification is necessary to choose PRBS with a frequency spectrum that exceeds the frequency spectrum identified by the system, so it must be:

$$f_{\text{low}} \leq f_{\text{min}} \wedge f_{\text{max}} \leq f_{\text{top}} \tag{21}$$

where

f_{low} = lower frequency of the PRBS

f_{min} = lower frequency of the identified system

f_{max} = upper frequency of the identified system

f_{top} = upper frequency of the PRBS

For the lower and upper frequency of the PRBS is true

$$f_{\text{lower}} = \frac{1}{P_T \, \Delta t}; \quad f_{\text{upper}} = \frac{0{,}44}{\Delta t} \tag{22}$$

where

P_T is the length of the period of the PRBS,

Δt is the sampling interval.

When estimating the frequency spectrum of an identified system, we assume that the system can be approximated by the transmission 1. order with time delay. We estime the value τ_{max} of the time constants of the transfer. Then the minimum and maximum frequency of the system is calculated from the relations

$$f_{\text{min}} \approx \frac{0{,}5}{\tau_{\text{max}} 2\pi}; \quad f_{\text{max}} \approx 20 f_{\text{min}} \tag{23}$$

5 Experimental Part

Recursive algorithm of parameter identification of one-dimensional system was tested to identify the parameter 7 floor rectifying column, which was a mixture of methanol-water distilled. The input variable was reflux, the output variable was the concentration of the liquid at the 7th floor column.

The identification was carried out in the vicinity of the working point with parameters:

- x_F = 0.25 mol. div. (concentration of feed flow),
- F = 19.3 mol min^{-1} (flow of feed),
- R = 6.5 mol min^{-1} (flow of reflux),
- W = 11.6 mol min^{-1} (flow of vapor).

The value of the reflux R has been changed according to the parameters of the PRBS with amplitude of 0.5 mol min^{-1}. This means that the flow of the reflux acquired values of R = 6 mol min^{-1}, respectively. R = 7 mol min^{-1}.

In figure 1 is a recording during measurement of input and output variables for one period PRBS.

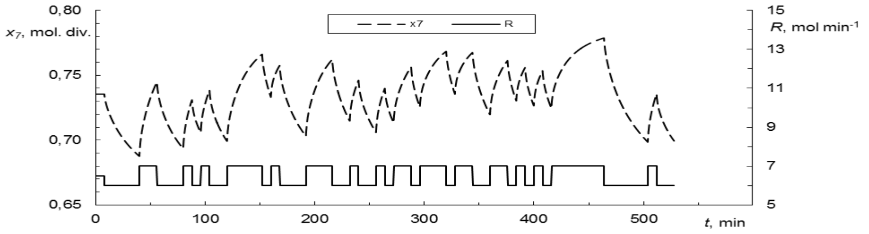


Fig. 1 Input and output variables

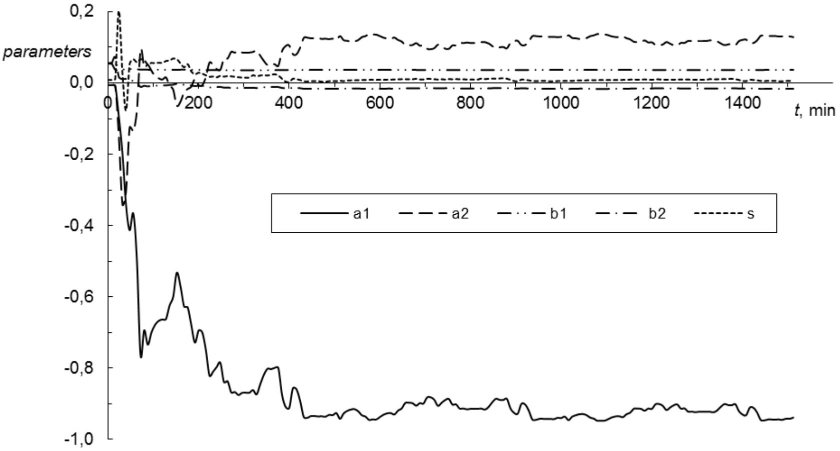


Fig. 2 Identification of parameters system

In figure 2 is recorded the identification of parameters of the mathematical model of the system. The model is in the form of differential equations of second order. The graph shows good convergence of values of identified parameters.

6 Conclusion

For proper design of algorithms of digital control systems is necessary to know the mathematical model of the controlled system. The experiment, which was presented in experimental part, demonstrated that the experimental identification is very easy to use and reliable for identification of controlled systems.

References

[1] Torkamani, S., Butcher, E.A., Khasawneh, F.A.: Parameter identification in periodic delay differential equations with distributed delay. Communications in Nonlinear Science and Numerical Simulation 18(4), 1016–1026 (2013), ISSN 10075704, doi:10.1016/j.cnsns.2012.09.001, http://linkinghub.elsevier.com/retrieve/pii/S1007570412004182

[2] Abbassi, F., Belhadj, T., Mistou, S., Zghal, A.: Parameter identification of a mechanical ductile damage using Artificial Neural Networks in sheet metal forming. Materials 45, 605–615 (2013), ISSN 02613069, doi:10.1016/j.matdes.2012.09.032, http://linkinghub.elsevier.com/retrieve/pii/S0261306912006632

[3] Victor, S., Malti, R., Garnier, H., Oustaloup, A.: Parameter and differentiation order estimation in fractional models. Automatica 26(1) (2013), ISSN 00051098, doi:10.1016/j.automatica.2013.01.026, http://linkinghub.elsevier.com/retrieve/pii/S0005109813000277

[4] Valderrama, J.O., Faã°ndez, C.A., Toselli, L.A.: Advances on modeling and simulation of alcoholic distillation. Part 1: Thermodynamic modeling. Food and Bioproducts Processing 90(4), 819–831 (2012), ISSN 09603085, doi:10.1016/j.fbp.2012.04.004, http://linkinghub.elsevier.com/retrieve/pii/S0960308512000302

[5] Valderrama, J.O., Toselli, L.A., Faundez, C.A.: Advances on modeling and simulation of alcoholic distillation. Part 2: Process simulation. Food and Bioproducts Processing 90(4), 832–840 (2012), ISSN 09603085, doi:10.1016/j.fbp.2012.04.003, http://linkinghub.elsevier.com/retrieve/pii/S0960308512000296

[6] Isermann, R.: Digitale Regelsysteme: Band II, Stochastische Regelungen, Mhrgrössenregelungen, Adaptive Regelunungen, Anwendungen. 2., überarb. u. erw. Aufl. Springer, Berlin (1987) ISBN 35-401-6597-5

[7] Mahapatra, P., Bequette, B.W.: Process Design and Control Studies of an Elevated-Pressure Air Separations Unit for IGCC Power Plants. In: Proceedings of the.. American Control Conference, pp. 2003–2008. IEEE, New York (2010) ISBN 978-1-4244-7427-1ISSN 0743-1619

[8] Isermann, R.: Digitale Regelsysteme: Band I, Grundlagen Deterministische Regelungen. 2., überarb. und erw. Aufl. Springer, Berlin (1987) ISBN 35-401-6596-7

Effect of Weighting Factors in Adaptive LQ Control

Jiri Vojtesek and Petr Dostal

Abstract. An adaptive control is a technique with strong theoretical background and lots of applications to the abstract and real systems. The big advantage can be found in usability of this control method for systems with negative control properties such as nonlinearity, time-delay, non-minimum behavior etc. The adaptive approach here is based on the choice of the external linear model of the originally nonlinear system parameters of which are updated in defined time moments via recursive identification. The control synthesis employs polynomial approach with linear-quadratic approach and spectral factorization. Resulted controller has two weighting factors as tuning parameters. This paper explores the effect these factors to the control. All proposed approaches were tested by simulations on the mathematical model of the continuous stirred-tank reactor as a typical member of the nonlinear lumped-parameters systems.

1 Introduction

It is known that the control of nonlinear processes is not simple and sometimes even very hard. Unfortunately, a major group of systems from industry is nonlinear and control of such processes with the conventional controllers with fixed parameters could lead to the unstable, inaccurate or unwanted output response when the state of the system change or the disturbance occurs. The adaptive control [1] is one way how we can solve these problems. This control method uses idea from the nature where plants or animals "adapt" their behavior to the actual state or environmental conditions. The adaptive controller adapts parameters or the structure to parameters of the controlled plant according to he selected criterion [2].

The adaptive approach here is based on the choice of the External Linear Model (ELM) as a linear approximation of the originally nonlinear system, parameters of which are identified recursively and parameters of the controller are recomputed according to identified ones. The dynamic analysis was used for the choice and the

Jiri Vojtesek · Petr Dostal
Tomas Bata University in Zlín, Faculty of Applied Informatics, Nam T.G. Masaryka 5555,
760 01 Zlin, Czech Republic
e-mail: {vojtesek,dostalp}@fai.utb.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 265–274.
DOI: 10.1007/978-3-319-00542-3_27

ELM‘s order. The delta-models [3] used here for identification are special type of discrete-time (DT) models parameters of which are recomputed to the sampling period and approaches to the continuous-time ones.

The continuous-time controller was designed via the polynomial synthesis [4], spectral factorization and the Linear-Quadratic (LQ) approach. These methods satisfy basic control requirements such as the stability, the reference signal tracking and the disturbance attenuation [4]. The resulted controller is called “hybrid” because it works in continuous-time but its parameters are recomputed in discrete time intervals together with the ELM delta-model identification.

The control procedures were tested on the mathematical model of the isothermal Continuous Stirred-Tank Reactor (CSTR). The mathematical model of this reactor is described by the set of five ordinary differential equations (ODE) [5] and the mathematical simulation software Matlab was used as a simulation tool.

2 Adaptive LQ Control

The adaptivity, or let us say sensitivity, to the change of the state or conditions of the control could be satisfied for example by the recursive identification during the control. The most of the processes has nonlinear behavior and identification of such processes is very complex. We can overcome this inconvenience by the choice of the External Linear Model (ELM) as a linear representation of the originally nonlinear system. Two major groups of the ELM are continuous-time (CT) and discrete-time (DT). Each ELM has its pros and cons. The method used here employs so called delta models as special type of DT models parameters of which are related to the sampling period and it means that they are close to the more accurate CT ones [6].

The polynomial theory is used in the controller design here. The scheme of the one degree-of-freedom (1DOF) control scheme is in Fig. 1.

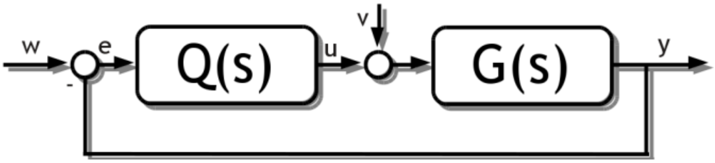


Fig. 1 1DOF control scheme

where *G*(*s*) represents the transfer function (i.e. ELM) of the controlled output and *Q*(*s*) denotes the transfer function of the controller, generally:

$$G(s)=\frac{b(s)}{a(s)};\quad Q(s)=\frac{q(s)}{p(s)} \tag{1}$$

where *q*(*s*) and *p*(*s*) are again commensurable polynomials with the properness condition $\deg p(s) \geq \deg q(s)$.

The Laplace transform of the transfer function *G*(*s*) is

$$G(s)=\frac{Y(s)}{U(s)}\Rightarrow Y(s)=G(s)\cdot U(s) \tag{2}$$

where Laplace transform of the input signal *u* is from Fig. 1.

$$U(s)=Q(s)\cdot E(s)+V(s)=Q(s)\cdot\left[W(s)-Y(s)\right]+V(s) \tag{3}$$

If we put polynomials *a*(*s*), *b*(*s*), *p*(*s*) and *q*(*s*) into (3) instead of Laplace transforms *G*(*s*) and *Q*(*s*), the equation (2) has form

$$Y(s)=\frac{b(s)q(s)}{a(s)p(s)+b(s)q(s)}\cdot W(s)+\frac{a(s)p(s)}{a(s)p(s)+b(s)q(s)}\cdot V(s) \tag{4}$$

Both fractions in (4) has the same denominators which are called *a characteristic polynomial of the closed loop* and this polynomial can be rewritten to the form

$$a(s)\cdot p(s)+b(s)\cdot q(s)=d(s) \tag{5}$$

where *d*(*s*) is a stable optional polynomial. The whole equation (5) is called Diophantine equation [4]. The stability of the control system is fulfilled for the stable polynomial *d*(*s*) on the left side of the Diophantine equation (5). Asymptotic tracking of the reference signal and disturbance attenuation is attained if the polynomial *p*(*s*) includes the least common divisor of denominators of transfer functions of the reference *w* and disturbance *v*:

$$p(s)=f(s)\cdot\tilde{p}(s) \tag{6}$$

If we expect both these signals from the range of the step functions, the polynomial *f*(*s*) = *s* and the Diophantine equation (5) is then

$$a(s)\cdot s\cdot\tilde{p}(s)+b(s)\cdot q(s)=d(s) \tag{7}$$

and the transfer function of the feedback controller is

$$\tilde{Q}(s)=\frac{q(s)}{s\cdot\tilde{p}(s)} \tag{8}$$

As it is written above, the polynomial *d*(*s*) on the right side of the Diophantine equation (7) is the stable optional polynomial. There are several ways how we can construct this polynomial. The simples one is the based on pole-placement method where *d*(*s*) is divided into one or more parts with double, triple, etc. roots, e.g.

$$d(s)=(s+\alpha)^{m};d(s)=(s+\alpha_1)^{m/2}\cdot(s+\alpha_2)^{m/2},\ldots \tag{9}$$

where the only condition is that $\alpha > 0$. The disadvantage of this method can be found in the uncertainty. There is no general rule which can help us with the choice of roots which are, of course, various for different controlled processes. One way how we can overcome this unpleasant feature is to use spectral factorization of the polynomial $a(s)$ in the denominator of the transfer function $G(s)$. Big advantage of this method is that it can make stable roots from every polynomial, even if it is unstable. The polynomial $d(s)$ is in this case

$$d(s) = n(s) \cdot g(s) \tag{10}$$

where parameters of the polynomial $n(s)$ are computed from the spectral factorization of the polynomial $a(s)$, i.e.

$$n^*(s) \cdot n(s) = a^*(s) \cdot a(s) \tag{11}$$

The second part, polynomial $g(s)$, could be designed via pole-placement method similarly as in (9) or we can use the Linear Quadratic (LQ) [7] which is based on the minimizing of the cost function in the complex domain

$$J_{LQ} = \int_0^\infty \left\{ \mu_{LQ} \cdot e^2(t) + \varphi_{LQ} \cdot \dot{u}^2(t) \right\} dt \tag{12}$$

where $\varphi_{LQ} > 0$ and $\mu_{LQ} \geq 0$ are weighting coefficients, $e(t)$ is the control error and $\dot{u}(t)$ denotes the difference of the input variable. It practically means, that parameters of the polynomial $g(s)$ are computed from the spectral factorization

$$\left(a(s) \cdot f(s)\right)^* \cdot \varphi_{LQ} \cdot a(s) \cdot f(s) + b^*(s) \cdot \mu_{LQ} \cdot b(s) = g^*(s) \cdot g(s) \tag{13}$$

Degrees of unknown polynomials $\tilde{p}(s)$, $q(s)$ and $d(s)$ are for the fulfilled properness condition generally:

$$\begin{array}{ll} \deg \tilde{p}(s) \geq \deg a(s) - 1 & \deg n(s) = \deg a(s) \\ \deg q(s) = \deg a(s) + \deg f(s) - 1 & \deg g(s) = \deg d(s) - \deg n(s) \\ \deg d(s) = 2\deg a + 1 & \end{array} \tag{14}$$

It is good if the controlled could be tuned somehow. In our case, the weighting coefficients φ_{LQ} and μ_{LQ} are tuning parameters which could affect the speed of the control, overshoots of the output variable or the course of the input variable. The advantage could be found also in the opportunity to choose what is more critical for the control – to have minimal control error $e(t)$ or minimal changes of the input variable $u(t)$. The control error could be affected by choice of the weighting coeficinet μ_{LQ} and the course of the input variable is tuned by the choice of the second weighting coefficient φ_{LQ} in equation (12).

3 Simulation Experiment

The nonlinear system under the consideration is an Isothermal Continuous Stirred Tank Reactor with complex reaction [5]. The schematic representation of this reactor is in Fig. 2.

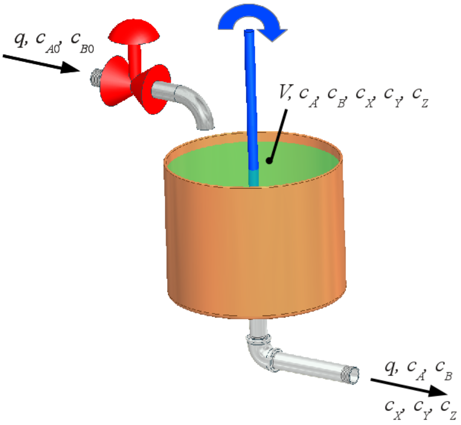


Fig. 2 Scheme of the Isothermic Continuous Stirred Tank Reactor

The reactions inside the reactor could be described by the scheme:

$$\mathrm{A} + \mathrm{B} \xrightarrow{k_1} \mathrm{X}; \quad \mathrm{B} + \mathrm{X} \xrightarrow{k_2} \mathrm{Y}; \quad \mathrm{B} + \mathrm{Y} \xrightarrow{k_3} \mathrm{Z} \tag{15}$$

We assume that the reactant inside is perfectly mixed and the volume is constant during experiments. We can now introduce the mathematical model of the system which comes from material balances inside the reactor:

$$\begin{aligned}
\frac{dc_A}{dt} &= \frac{q}{V}\left(c_{A0} - c_A\right) - k_1 \cdot c_A \cdot c_B \\
\frac{dc_B}{dt} &= \frac{q}{V}\left(c_{B0} - c_B\right) - k_1 \cdot c_A \cdot c_B - k_2 \cdot c_B \cdot c_X - k_3 \cdot c_B \cdot c_Y \\
\frac{dc_X}{dt} &= \frac{q}{V}\left(c_{X0} - c_X\right) + k_1 \cdot c_A \cdot c_B - k_2 \cdot c_B \cdot c_X \\
\frac{dc_Y}{dt} &= \frac{q}{V}\left(c_{Y0} - c_Y\right) + k_2 \cdot c_B \cdot c_X - k_3 \cdot c_B \cdot c_Y \\
\frac{dc_Z}{dt} &= \frac{q}{V}\left(c_{Z0} - c_Z\right) + k_3 \cdot c_B \cdot c_Y
\end{aligned} \tag{16}$$

where q denotes volumetric flow rate, V is used for volume of the reactant, c_A, c_B, c_X, c_Y and c_Z are concentrations, $k_{1\text{-}3}$ are rate constants and t is time. The fixed parameters are in Table 1[5].

Table 1 Parameters of the reactor

Rate constant of the reaction 1	$k_1 = 5\times10^{-4}\ m^3.kmol^{-1}.s^{-1}$
Rate constant of the reaction 2	$k_2 = 5\times10^{-2}\ m^3.kmol^{-1}.s^{-1}$
Rate constant of the reaction 3	$k_3 = 2\times10^{-2}\ m^3.kmol^{-1}.s^{-1}$
Volume of the reactant	$V = 1\ m^3$
Input concentration of the compound A	$c_{A0} = 0.4\ kmol.m^{-3}$
Input concentration of the compound B	$c_{B0} = 0.6\ kmol.m^{-3}$
Input concentration of the compound X	$c_{X0} = 0\ kmol.m^{-3}$
Input concentration of the compound Y	$c_{Y0} = 0\ kmol.m^{-3}$
Input concentration of the compound Z	$c_{Z0} = 0\ kmol.m^{-3}$

The system has theoretically six input variables – the volumetric flow rate q, input concentrations of the compounds A, B, X, Y, Z – c_{A0}, c_{B0}, c_{X0}, c_{Y0}, c_{Z0} and five state variables – final concentrations of the compounds A, B, X, Y and Z – c_A, c_B, c_X, c_Y and c_Z.

The mathematical model in (16) shows that this system belongs to the class of *nonlinear lumped-parameters systems*. Nonlinearity comes from the multiplication of the state variables and the set of ODE (16) is typical for systems with lumped parameters unlike system with distributed parameters which are described by the partial differential equations (PDE).

3.1 External Linear Model

As it written above, the choice of the external linear model comes from the static and dynamic analyses inside the reactor. The previous experiments [8] have shown that the optimal working point for the concentration of the compound B as a controlled variable is around the volumetric flow rate $q^s = 1\times10^{-4}\ m^3.s^{-1}$. The change of the volumetric flow rate q was chosen as a manipulated (input) variable from the practical point of view and control variable was the change of the concentration of the product B. The input and output variables are then:

$$u(t) = \frac{q(t) - q^s}{q^s}\cdot 100\,[\%];\quad y(t) = c_B(t) - c_B^s\left[kmol.m^{-3}\right] \tag{17}$$

Dynamic analyses in [8] indicates, that the controlled output $y(t)$ could be described by the second order continuous-time transfer function (ELM), i.e.

$$G(s) = \frac{b(s)}{a(s)} = \frac{b_1 s + b_0}{a_2 s^2 + a_1 s + a_0} \tag{18}$$

Degrees of the polynomials in (14) are then

$$\begin{gathered}
\deg q(s)=2 \Rightarrow q(s)=q_2 s^2+q_1 s+q_0 \\
\deg \tilde{p}(s)=2 \Rightarrow \tilde{p}(s)=s^2+p_1 s+p_0 \\
\deg g(s)=3 \Rightarrow g(s)=s^3+g_2 s^2+g_1 s+g_0 \\
\deg n(s)=2 \Rightarrow n(s)=s^2+n_1 s+n_0 \\
\deg d(s)=5 \Rightarrow d(s)=s^5+d_4 s^4+d_3 s^3+d_2 s^2+d_1 s+d_0
\end{gathered} \tag{19}$$

Polynomials *n*(*s*) and *g*(*s*) are computed as a results of spectral factorizations (11) and (13):

$$\begin{gathered}
g_0=\sqrt{\mu_{LQ} b_0^2},\, g_1=\sqrt{2g_0 g_2+\varphi_{LQ} a_0^2+\mu b_1^2},\, g_2=\sqrt{2g_1 g_3+\varphi_{LQ}\left(a_1^2-2a_0\right)}, \\
g_3=\sqrt{\varphi_{LQ}},\quad n_0=\sqrt{a_0^2},n_1=\sqrt{2n_0+a_1^2-2a_0}
\end{gathered} \tag{20}$$

Polynomials *a*(*s*) and *b*(*s*) of the ELM (18) are computed by the well-known and easily programmable recursive least-squares method [9]. As it was already mentioned, the delta-models used for the ELM here have parameters estimated in the discrete time intervals but as they are related to the sampling period T_v, we assume that they are close to the continuous ones. This assumption allows the usage of these parameters in computations of the controller's parameters. The vector of parameters and the data vector are in this case:

$$\begin{gathered}
\boldsymbol{\varphi}_\delta(k-1)=\left[-y_\delta(k-1),-y_\delta(k-2),u_\delta(k-1),u_\delta(k-2)\right]^T \\
\boldsymbol{\theta}_\delta(k)=\left[a_1^\delta,a_0^\delta,b_1^\delta,b_0^\delta\right]^T
\end{gathered} \tag{21}$$

where y_δ and u_δ denotes the recomputed output and input variables to the delta-model:

$$\begin{gathered}
y_\delta(k)=\frac{y(k)-2y(k-1)+y(k-2)}{T_v^2} \\
y_\delta(k-1)=\frac{y(k-1)-y(k-2)}{T_v} \qquad y_\delta(k-2)=y(k-2) \\
u_\delta(k-1)=\frac{u(k-1)-u(k-2)}{T_v} \qquad u_\delta(k-2)=u(k-2)
\end{gathered} \tag{22}$$

The vector of parameters is computed from:

$$y_\delta(k)=\boldsymbol{\theta}_\delta^T(k)\cdot\boldsymbol{\varphi}_\delta(k-1)+e(k) \tag{23}$$

where *e*(*k*) is a general random immeasurable error.

3.2 Simulation Results

The goal of simulations is to explore effect of the weighting factors φ_{LQ} and μ_{LQ} as a tuning parameters and find optimal combinations of these parameters.

All simulations were performed with the same conditions. The simulation time was $T_f = 20\ 000\ s$, five different step changes of the reference signal were done during this time, sampling period was $T_v = 10\ s$. The input variable was limited inside the bounds $u(t) = <-100\%;\ 100\%>$.

The first simulation analysis presented in Fig. 3 shows impact of the weighting factor φ_{LQ} to the course of the output variable. The second parameter was fixed to the value $\mu_{LQ} = 2$.

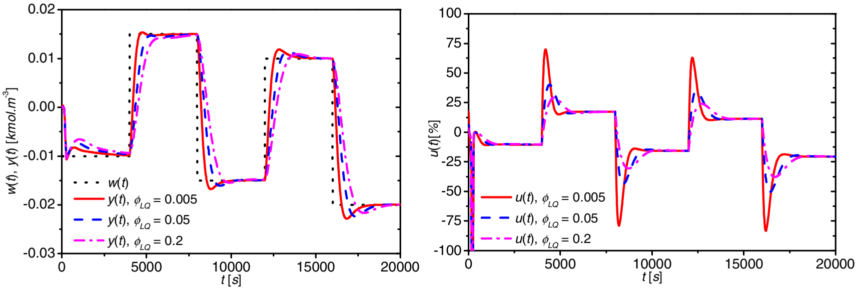


Fig. 3 Courses of the reference signal, $w(t)$, output signal $y(t)$ and input signal $u(t)$ for various values of weighting factor φ_{LQ}

It is clear, that increasing value of φ_{LQ} results in slower, but smoother course of the output and also input variable which is also good from the practical point of view where the input variable, in our case volumetric flow rate, is represented by twist of the valve on the input pipe and shocking or quick changes of this variable could affect service life of the valve.

Simulations for the different μ_{LQ} and fixed value of $\varphi_{LQ} = 0.1$ were performed in the second study.

Effect of the parameter μ_{LQ} in Fig. 4 is opposite to the previous one. In this case, increasing value of μ_{LQ} results in the quicker output response but with bigger overshoots.

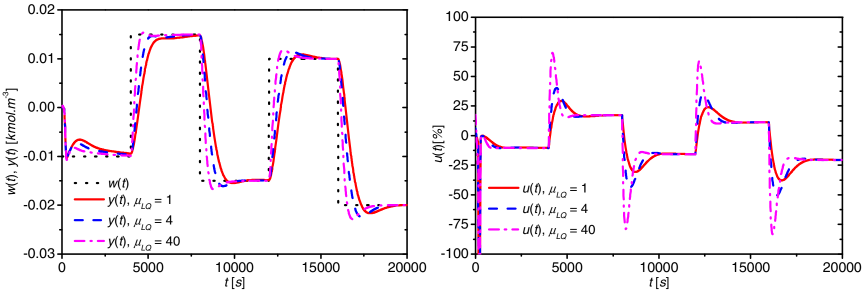


Fig. 4 Courses of the reference signal, $w(t)$, output signal $y(t)$ and input signal $u(t)$ for various values of weighting factor μ_{LQ}

The only problem of both simulation studies can be found at the very beginning of the control where the course of the input variable and of course the output variable is very inaccurate. This is caused by the recursive identification which purposely starts from the general point $\boldsymbol{\theta}_\delta(0) = [0.1, 0.1, 0.1, 0.1]^T$ and it takes time 50-100 s to get the right vector of parameters.

Both responses in graphs Fig. 3 and 4 have similar courses which lead us to explore the effect of the ratio $\varphi_{LQ} : \mu_{LQ}$ to the results of the control. Quantitative criteria besides visual comparisons were simple quality criteria S_u and S_y computed as

$$S_u = \sum_{i=300}^{N} \left(u(i) - u(i-1)\right)^2 [-]; \quad S_y = \sum_{i=300}^{N} \left(w(i) - y(i)\right)^2 [-], \text{ for } N = \frac{T_f}{T_v} \tag{24}$$

These criteria were computed not for all simulation time but it starts after the first step change and last to the end of the simulation. This was because of the objectivity of the results as the courses of both input and output variables are suboptimal at the very beginning due to inaccurate recursive identification as it was mentioned above. The results are shown in Table 2 where the left side of the table is related to the first simulation analysis (effect of the parameter φ_{LQ}) and the right side presents results of the second study for the different μ_{LQ}.

Table 2 Qualitative results of the control

φ_{LO}	μ_{LO}	$\varphi_{LO} : \mu_{LO}$	S_u	S_y	φ_{LO}	μ_{LO}	$\varphi_{LO} : \mu_{LO}$	S_u	S_y
0.005	2	1:400	12876.71	0.0571	0.1	1	1:10	13296.56	0.1381
0.05	2	1:40	12435.91	0.0984	0.1	4	1:40	12435.91	0.0984
0.2	2	1:10	13296.56	0.1381	0.1	40	1:400	12876.71	0.0571

Now it is clear, that results of both studies are comparable and ratio $\varphi_{LQ} : \mu_{LQ}$ could be used the qualitative criteria – bigger ratio produces quicker output response but with the possibility of overshoots.

4 Conclusion

The paper deals with hybrid adaptive LQ control of the nonlinear system. The polynomial approach, spectral factorization and LQ approach were used for constructing of the continuous-time controller. The delta ELM belongs to the class of discrete-time models but its parameters approaches to the continuous ones. That is why we call this controller hybrid – it runs in continuous-time, but parameters of the delta ELM are identified recursively in the defined sampling intervals. This adaptive controller has two tuning parameters – weighting factors which give attention to the output error or the change of the input variable. It was proofed, that are not important values of these weighting factors separately, but ratio of them, e.g. $\varphi_{LQ} : \mu_{LQ}$. Decreasing value of this ratio produces slower speed of the control but smoother course of both the input and the output variables. Simulation tests were done nonlinear on the mathematical model of the CSTR but the proposed controller could be used also on similar types of abstract or real models.

References

[1] Åström, K.J., Wittenmark, B.: Adaptive Control. Addison Wesley, Reading (1989) ISBN 0-201-09720-6

[2] Bobal, V., Böhm, J., Fessl, J., Machacek, J.: Digital Self-tuning Controllers: Algorithms. Implementation and Applications. In: Advanced Textbooks in Control and Signal Processing. Springer-Verlag London Limited (2005) ISBN 1-85233-980-2

[3] Middleton, H., Goodwin, G.C.: Digital Control and Estimation - A Unified Approach. Prentice Hall, Englewood Cliffs (2004) ISBN 0-13-211798-3

[4] Kucera, V.: Diophantine equations in control – A survey. Automatica 29, 1361–1375 (1993)

[5] Ingham, J., Dunn, I.J., Heinzle, E., Prenosil, J.E.: Chemical Engineering Dynamics. An Introduction to Modeling and Computer Simulation, 2nd Completely Revised edn. VCH Verlagsgesellshaft, Weinheim (2000) ISBN 3-527-29776- 6

[6] Stericker, D.L., Sinha, N.K.: Identification of continuous-time systems from samples of input-output data using the δ-operator. Control-Theory and Advanced Technology 9, 113–125 (1993)

[7] Hunt, K.J., Kucera, V., Sebek, M.: Optimal regulation using measurement feedback. A polynomial approach. IEEE Transactions on Automation Control 37(5), 682–685 (1992)

[8] Zelinka, I., Vojtesek, J., Oplatkova, Z.: Simulation Study of the CSTR Reactor for Control Purposes. In: Proc. of 20th European Conference on Modelling and Simulation, ESCM 2006, Bonn, Germany, pp. 479–482 (2006)

[9] Rao, G.P., Unbehauen, H.: Identification of continuous-time systems. IEEE Process-Control Theory Application 152, 185–220 (2005)

Unstable Systems Database: A New Tool for Students, Teachers and Scientists

František Gazdoš, Jiří Marholt, and Jaroslav Kolařík

Abstract. The contribution presents a starting project of a site focused on unstable systems. It is a web-based database in the bilingual version (ENG/CZ) which can be used as an information database for models of unstable processes. The site contains mathematical models of such systems including their simulation files together with basic information about stability of dynamical systems. The paper outlines motivation for development of this database, presents its basic structure and discusses several models from the site. Areas of prospective usage are also suggested together with possible directions of further development of this project.

1 Introduction

Many processes in industrial practice are unstable. These can be e.g. various types of reactors, distillation columns, combustion systems, etc. [1], [2]. There are also lots of systems in the environmental and social fields that are naturally unstable. Besides this, some systems in the military and aviation areas are deliberately designed to be unstable in order to gain better maneuverability and increase speed of command responses. Consequently all these systems are more difficult to control than the stable ones and are of special interest of designers and control engineers. They have to pay extra attention in order to implement safe control systems; if they fail the consequences can be catastrophic [3]. Designers and control engineers have to understand basic limitations that stem from the process instability [4], [5]. As real experiments with unstable systems can be hazardous, modeling and simulation play important roles in designing safe control systems for such processes nowadays. These tools enable safe experiments, analysis of dangerous states of the systems, design of convenient control and much more.

The presented site has been developed to enable students, teachers, scientist and many others interested in unstable processes easy access to mathematical models of such systems. All these people can easily use presented models for their own simulation experiments, testing control algorithms, etc. This will broaden awareness about unstable processes and problems they cause. Due to the fact that

František Gazdoš · Jiří Marholt · Jaroslav Kolařík
Tomas Bata University in Zlin, Faculty of Applied Informatics, nam. T. G. Masaryka 5555, 760 01 Zlin, Czech Republic
e-mail: gazdos@fai.utb.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 275–284.
DOI: 10.1007/978-3-319-00542-3_28

the database is easily accessible via the Internet [18] it can be used by wide range of users for various purposes, e.g. pedagogical, scientific and others.

The paper is divided into these sections: after a brief introduction into the stability of dynamical systems the contribution continues by examples of unstable processes form the site including their simplified mathematical models together with a brief discussion on their behaviour. Further, a basic structure of the developed site is outlined and explained and the paper concludes suggesting possible further development of the project.

2 Stability of Dynamical Systems

Stability is the fundamental property of control systems. Therefore a great deal of effort has been focused towards proper definition, testing and providing for system stability.

2.1 Defining Stability

Although all people naturally understand the concept of stability and are able to describe what stable behaviour is and what is not, a proper mathematical definition is not so straightforward. Generally, stability can be formulated as ability to recover from perturbations – short-time disturbances or non-zero initial conditions. One of the definitions says that a system is stable if bounded input into the system produces a bounded output from the system. This is so called *BIBO* (Bounded Input – Bounded Output) *stability*, e.g. [6], [7]. Another recognized and more general definition is the *Lyapunov stability*, e.g. [6], [8], [7], [9]. It states, simply speaking, that a system is stable if its output and all states are bounded and converge asymptotically to zero from sufficiently small initial conditions.

2.2 Testing Stability

During the past decades many methods of stability testing have been developed. Usage of a particular method depends on the properties of the system to be tested – e.g. if it is linear or nonlinear, continuous-time or discrete-time, time-variant or time-invariant, etc. The methods can be both numerical and graphical. An interested reader can find details in books focused on systems theory or control engineering, e.g. [6], [7], [9], [10].

2.3 Stabilization

An unstable system can be stabilized by feedback. There are many sources focused on the control system design for unstable processes, e.g. [1], [11], [12], [13], [2], [14], [15]. Many of these works solve the control system design problem connected also with delayed and non-minimum-phase systems which are also problematic to control.

Besides testing and attainment of stability it is often important to test and ensure certain measure of stability, i.e. *relative stability* which gives answer to the question how far the system is from instability. For control systems design, so called *gain* and *phase margins* are frequently used, for details see e.g. [7], [9], [10].

The so-called *robust stability* is next important term in control engineering. It is used for the case we want to test/achieve stability not only for one system but for a certain class of systems, typically a nominal system and some its neighborhood, which is useful in the case of uncertain models. An interested reader is referred to books devoted to the robust systems design, e.g. [16], [17].

3 Examples of Unstable Systems

As explained in the introduction section unstable processes are common in many areas of our daily lives. Several such systems are briefly described in this section. Mathematical models together with simulation files and original sources are also available online from the developed site [18].

3.1 Non-ideal CSTR

This process is represented by a continuous stirred-tank reactor (CSTR) with non-ideal mixing. The process can be sketched as illustrated in Figure 1. A simplified mathematical model of the process dynamics can be described by the following nonlinear differential formulas [19], [2]:

$$\begin{gathered}\frac{dc(t)}{dt}=\frac{nQ}{mV}\left[c_f(t)-c(t)\right]-\frac{k_1c(t)}{\left[1+k_2c(t)\right]^2}, \\ nc(t)+(1-n)c_f(t)=c_e(t).\end{gathered} \tag{1}$$

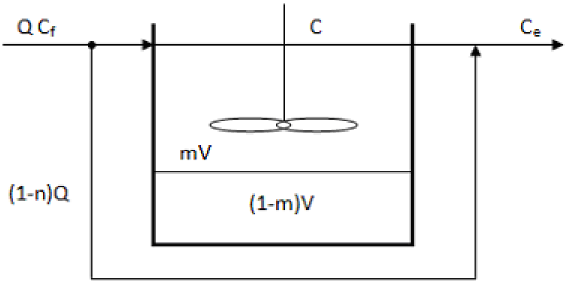


Fig. 1 Non-ideal CSTR

In the picture and equations, $c(t)$ is the concentration of the reactant in the well mixed zone, $c_e(t)$ is the concentration in the exit stream (controlled variable) and $c_f(t)$ is the feed concentration (manipulated variable). Further, n is the fraction of the reactant feed that enters the zone of perfect mixing and m is the fraction of the reactor total volume V where the reaction occurs. Constants k_1, k_2 describe the reaction rate and Q is the inlet flow rate. Parameters of the model are defined as: $n = m = 0.75$ [-], $V = 1$ [l], $Q = 0.033$ [l/s], $k_1 = 10$ [s^{-1}], $k_2 = 10$ [l/mol] and a nominal operating point is described as: $c_f = 6.484$ [mol/l], $c_e = 1.8$ [mol/l] and $c = 0.2387$ [mol/l]. Linearization of the nonlinear model (1) around the given nominal operating point gives the transfer function model as:

$$\frac{\Delta c_e(s)}{\Delta c_f(s)} = \frac{0.7725s - 0.1727}{3.1s - 1}. \tag{2}$$

A state-space representation of the linearized model in the general form:

$$\mathbf{x}'(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t); \quad \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \tag{3}$$

where $\mathbf{x}(t)$ defines a vector of state variables, $\mathbf{y}(t)$ a vector of output variables and $\mathbf{u}(t)$ a vector of input variables can be obtained e.g. in this form:

$$\mathbf{A} = [0.3226], \mathbf{B} = [0.1250], \mathbf{C} = [0.1974], \mathbf{D} = [0.2492]. \tag{4}$$

In the case of the presented reactor the variables $\mathbf{x}(t)$, $\mathbf{y}(t)$ and $\mathbf{u}(t)$ are only scalar and correspond to the reactor variables $c(t)$, $c_e(t)$ and $c_f(t)$ respectively (consequently the matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ are also only scalars). From the control theory point of view, the models (2)-(4) represent a first-order proper system which is unstable (one positive pole, i.e. denominator root, located at $p_1 = 0.3226$) with non-minimum-phase behaviour (one positive zero, i.e. numerator root, located at $z_1 = 0.2236$), and with gain $k = 0.1727$ [-]. Such systems which are both unstable and non-minimum-phase are not so easy to control. The step-response of the model recorded in Figure 2 clearly demonstrates instability of the system.

3.2 Ballistic Missile

A ballistic missile can represent another unstable system from military industry. Although it is completely different from the previous one, it shares the property of instability and consequently problematic control. When controlling altitude of the ballistic missile, the transfer function relating the altitude $y(t)$ to the thrust chamber deflection $\delta(t)$ has the following form [20], [2]:

$$\frac{\Delta y(s)}{\Delta \delta(s)} = \frac{7.21(s + 0.0526)}{(s + 1.6)(s - 1.48)(s - 0.023)} \tag{5}$$

Its state-space representation in the general form (3) can be obtained e.g. as:

$$\mathbf{A}=\begin{bmatrix}-0.0970 & 1.1854 & -0.0545\\ 2 & 0 & 0\\ 0 & 0.5 & 0\end{bmatrix},\mathbf{B}=\begin{bmatrix}2\\0\\0\end{bmatrix},\mathbf{C}=[0\quad 1.8025\quad 0.1896],\mathbf{D}=[0]. \qquad (6)$$

From the control theory point of view the missile represents a strictly proper unstable system of 3rd order. The instability is given by the two poles located in the right half of the complex plane (p_1=0.023, p_2=1.48) as illustrated in Figure 3. The system also has relatively fast dynamics with time-constants in seconds.

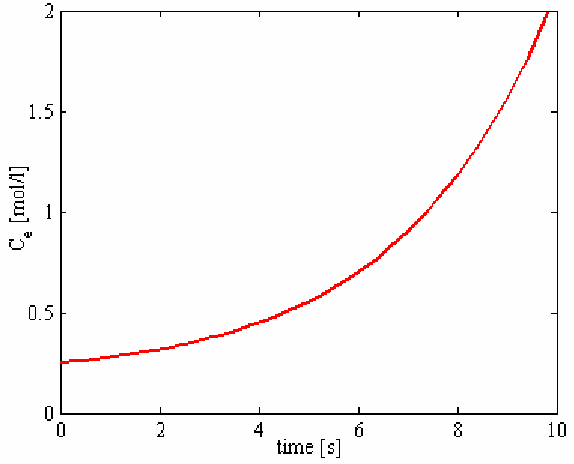


Fig. 2 Non-ideal CSTR step-response

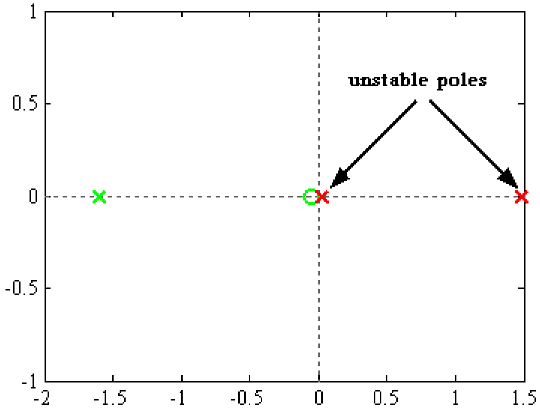


Fig. 3 Pole-zero map of the ballistic missile

3.3 X-29 Aircraft

The X-29 illustrated in Figure 4 was an experimental aircraft that tested forward-swept wing, canard control surfaces, and other novel aircraft technologies. It was deliberately designed with static instability to increase its maneuverability and speeds of command response. However, consequently it was impossible to pilot this airplane conventionally by manual flight controls and it required the use of so-called fly-by-wire (computerized) control system. In addition special hardware (sensors, control processors and actuators) had to be used to stabilize the system over all flight regimes and all loading conditions. Considerable effort has been devoted to the design of flight control system for this airplane, e.g. [21], [22], [3].

Fig. 4 X-29 aircraft

The benefits of instability (better maneuverability and faster reaction) were desired in the transonic and supersonic flight regimes, so the airplane was designed to be modestly unstable in those regimes. However, due to a basic aerodynamic phenomenon the X-29's slight instability at supersonic speeds turned into a much more dramatic instability at subsonic speeds. A simplified linearized model at one such flight condition given by a transfer function has the form:

$$G(s)=\frac{s-26}{s-6} \tag{7}$$

As can be clearly seen, the airplane's real pole (denominator root) is as large as +6 rad/s which makes this system nearly impossible to control manually – it can be compared, simply speaking, to balancing a 1-ft-long stick [3]. Besides the unstable pole the systems has also strong non-minimum-phase behaviour, i.e. inverse response (a zero – numerator root, located at z_1 = 26). These facts make this system very difficult to control. State-space realization of the model (7) in the general form (3) can be obtained e.g. as:

$$\mathbf{A}=[6], \mathbf{B}=[4], \mathbf{C}=[-5], \mathbf{D}=[1]. \tag{8}$$

Next figure (Figure 5) shows sensitivity function $S(\omega)$ of a X29 prototype.

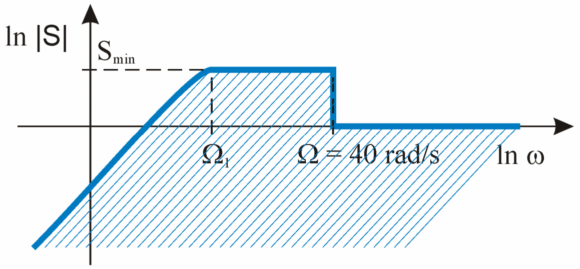


Fig. 5 X-29 sensitivity function

From the plot it can be seen that the system has considerably limited bandwidth (up to 40 rad/s) which is given by the used HW components (sensors, control processors, actuators), airplane mechanical structure and aerodynamics conditions. Consequently it narrows possibilities of convenient control system design.

All here discussed models are available online at the presented site http://www.unstable-systems.cz [18] with short description, downloadable model in the MATLAB/Simulink environment and selected references related to modeling, simulation, control system analysis and design of the systems. Next section briefly presents main structure of the developed site devoted to unstable systems and outlines main possibilities it offers for prospective users.

4 Site Structure and Choices

The database of unstable system has been developed as an open, easily extensible system in the bilingual version (English/Czech). It is easily accessible on-line via the Internet at the web-address http://www.unstable-systems.cz. A starting version of the site was implemented within the final work [23] using a free and open source content management system Joomla! of the 2.5 version (available online at http://www.joomla.org and described in detail in e.g. [24]). This makes administration of the site very easy. Basic structure of the web presentation is divided into several bookmarks – *HOME*, *UNSTABLE SYSTEMS*, *(IN)STABILITY*, *CONTACT US* and *ABOUT PROJECT*. Apart from this basic structure there are also user-related services such as registration, profile editing and web administration.

The *HOME* bookmark is the starting point of the site. It introduces a purpose of the site, enables registration, shows latest news and recent posts together with simple statistics and some useful links. Registered users have the following possibilities:

- access to files with simulation models of the systems
- access to latest news via a newsletter
- possibility of articles rating
- possibility to add comments to the systems models

It is also possible to search within the site and change language (ENG/CZ) here.

The *UNSTABLE SYSTEMS* bookmark is the main part of the site which contains mathematical models of unstable processes. So far (March 2013) it contains following models:

- ballistic missile
- fluidized bed reactor
- inverted pendulum
- magnetic levitation system
- non-ideal continuous stirred-tank reactor
- X-29 aircraft

Every model has the following information:

- brief description of the system
- scheme of the process or picture of the system
- simplified mathematical model
- definition of used variables and parameters
- downloadable model in e.g. MATLAB/Simulink environment
- sources of further information

It is also possible to generate a printable version of the models or send a link of a model by e-mail. Registered users can download the simulation models, add comments and rate them. Besides this it is also possible to search within the models and sort them according to various criteria.

The (IN)STABILITY bookmark explains basics about stability of systems – general understanding of the term, several definitions, such as *BIBO* (Bounded Input – Bounded Output) and *Lyapunov* stability. It also offers further reading on this subject.

Next bookmark *CONTACT US* enables to send questions, suggestions and remarks, etc. on the site and models to the authors. The final bookmark *ABOUT PROJECT* briefly introduces basic information about this project including main authors, brief description and terms of use.

5 Conclusions

Modeling and simulation tools play important roles in our lives nowadays. In the case of unstable systems analysis and control design, they role is crucial. Experiments with such systems without the proper knowledge about possible consequences can be very hazardous [3]. The goal of this contribution was to present a starting project of the web-based database of unstable systems. The site can help students, teachers, designers, scientists and many others to understand basic properties of unstable systems. This is done via available models of the systems/processes and suggested further readings. The developed web-site is an open, constantly developing system which is still “under construction”. Therefore experiences of users, their suggestions, remarks and comments are welcome.

Further development of the site will be focused on the extension of the given information concerning the systems stability, description of the models and suggested further readings. The number of available unstable systems models will of course grow as well as the number of simulation files (not necessarily limited to the MATLAB/Simulink environment). Another attractive extension can be seen in the possibility to implement simulation directly into the site, e.g. using the popular *Easy Java Simulations* (EJS) open-source software tool. At the moment only the site administrator can add articles and models into the system. The possibility to do this by registered users is also being considered, which would certainly help to develop further the database. Besides simulation models of the unstable systems, also simulation files related to control system design for such processes can be added gradually.

References

1. Chidambaram, M.: Control of unstable systems: a review. Journal of Energy, Heat and Mass Transfer 19, 49–57 (1997)
2. Padma Sree, R., Chidambaram, M.: Control of unstable systems. Alpha science Int. Ltd., Oxford (2006)
3. Stein, G.: Respect the unstable. IEEE Control System Magazine 23, 12–25 (2003)
4. Middleton, R.H.: Trade-offs in linear control system design. Automatica 27, 281–292 (1991)
5. Skogestad, S., Havre, K., Larsson, T.: Control limitations for unstable plants. In: Proceedings of the 15th Triennial World Congress, pp. 485–490 (2002)
6. Willems, J.L.: Stability Theory of Dynamical Systems. Wiley, New York (1970)
7. Skogestad, S., Postlethwaite, I.: Multivariable Feedback Control: Analysis and Design. Wiley, Chichester (2005)
8. Parks, P.C.: A.M. Lyapunov's stability theory - 100 years on. IMA Journal of Mathematical Control & Information 4(9), 275–303 (1992)
9. Åström, K.J., Murray, R.M.: Feedback Systems: An Introduction for Scientist and Engineers. Princeton University Press (2008)
10. Doyle, J.C., Francis, B.A., Tannenbaum, A.R.: Feedback Control Theory. Dover Publications (2009)
11. Park, J.H., Sung, S.W., Lee, I.B.: An enhanced PID control strategy for unstable processes. Automatica 34, 751–756 (1998)
12. Marchetti, G., Scali, C., Lewin, D.R.: Identification and control of open-loop unstable processes by relay methods. Automatica 37, 2049–2055 (2001)
13. Lozano, R., Castillo, P., Garcia, P., Dzul, A.: Robust prediction-based control for unstable delay systems: application to the yaw control of a mini-helicopter. Automatica 40, 603–612 (2004)
14. García, P., Albertos, P., Hägglund, T.: Control of unstable non-minimum-phase delayed systems. Journal of Process Control 16, 1099–1111 (2006)
15. Dostál, P., Gazdoš, F., Bobál, V.: Design of controllers for time delay systems - part II: integrating and unstable systems. Journal of Electrical Engineering 59, 3–8 (2008)
16. Barmish, B.R.: New Tools for Robustness of Linear Systems. Macmillan (1994)

17. Bhattacharyya, S.P., Chapellat, H., Keel, L.H.: Robust Control - The Parametric Approach. Prentice-Hall (1995)
18. Gazdoš, F., Kolařík, J.: Database of unstable systems (2012), `http://www.unstable-systems.cz`
19. Liou, C.T., Chien, Y.S.: The effect of nonideal mixing on input multiplicities in a CSTR. Chem. Eng. Sci 46, 2113–2116 (1991)
20. Blakelock, J.H.: Automatic Control of Aircraft and Missiles. John Wiley, New York (1991)
21. Rogers, W.L., Collins, D.J.: X-29 $H_($ controller synthesis. J. Guidance Control and Dynamics 4(15), 962–967 (1992)
22. Clarke, R., Burken, J.J., Bosworth, J.T., Bauer, J.E.: X-29 flight control system – lessons learned. Int. J. Control 1(59), 199–219 (1994)
23. Kolařík, J.: Web-based Database of Unstable Systems. Bachelor's thesis. Tomas Bata University in Zlín, Faculty of Applied Informatics, CZ (2012)
24. Marriott, J., Waring, E.: The official Joomla! Book. Addison-Wesley Professional (2013)

Nonlinear State Estimation and Predictive Control of pH Neutralization Process

Jakub Novák and Petr Chalupa

Abstract. In the paper the fuzzy Kalman filter (KF) is proposed to allow for adaptation to changing properties of the controlled process. The fuzzy KF is used to estimate both states and unmeasured disturbances of the nonlinear process. Further, a Model Predictive Control (MPC) based on the fuzzy representation of the nonlinear process is formulated. The performance of the proposed estimation fuzzy scheme and predictive controller is evaluated through computer simulations of the pH neutralization process. The pH neutralization process is widely recognized as a difficult control problem due to the strong nonlinearity of the process.

1 Introduction

The control of pH level is often performed in many industrial and chemical processes. High performance control is often difficult to achieve due to severe process nonlinearities and time-varying parameters. This nonlinearity together with frequent and rapid load changes makes the pH control in the wastewater treatment facilities very challenging. Generally, two main control approaches can be distinguished to control pH processes. The first group is based on the first principle models and use linear, nonlinear or adaptive control strategies to cope with the process nonlinearity. Henson and Seborg [1] have developed an adaptive nonlinear controller which combines the input-output linearizing controller with reduced-order, open-loop observer. In [2] the process is modeled by a set of linear models constructed by velocity-based linearization of the first principle model. To avoid computational burden, the pH value is controlled using a neural network controller which approximates the optimal model predictive control strategy.

The other group considers the process as a black-box model and the parameters of this model are identified through off-line identification. In [3] Wiener model identification from input-output data obtained from a pH neutralization is

Jakub Novák · Petr Chalupa
Tomas Bata University in Zlin, Faculty of Applied Informatics, T.G. Masaryka 5555,
760 01 Zlin, Czech Republic
e-mail: {jnovak,chalupa}@fai.utb.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 285 – 294.
DOI: 10.1007/978-3-319-00542-3_29

presented. The identified Wiener model is used as an internal model in a model predictive controller. Oblak and Skrjanc in [4] approximated the nonlinear output mapping of the continuous Wiener model of pH neutralization process by a fuzzy system whose apexes were determined by c-means clustering. Mahmoodi et al. used Laquerre filters and simple polynomials as linear and nonlinear parts of Wiener structure [5]. The model is then used within model predictive framework. Many authors approximated the nonlinear dynamics of the process by multiple models. Switching among multiple linear local models is studied in [6]. The active model is then used in model predictive control. Due to the nonlinearity of the process the weighting matrices of the cost function are also adapted.

In the presented work the problem of estimation of the states of the nonlinear process is overcome with the help of multiple local estimators. Multiple estimates of the Kalman Filter bank are weighted using the set of fuzzy rules. Thus, the derivation necessary for computation of Jacobians in the Extended Kalman Filter strategy can be avoided. The obtained estimates of states and disturbances are then used in state-space predictive controller. The proposed control strategy is tested on simulated pH neutralization process.

2 State Estimation with Extended Kalman Filter

The Extended Kalman filter (EKF) is the most widely applied state estimation algorithm for nonlinear systems. It is a predictor-corrector type linear estimator obtained by linearization of a nonlinear model at each sampling interval. The states of the pH neutralization process are the acid, base and buffer concentrations whose measurements are usually not available and must be estimated. In the extended Kalman filter the state transition and observation models must be differentiable functions:

$$\begin{aligned} \boldsymbol{x}(k+1) &= f(\boldsymbol{x}(k), \boldsymbol{u}(k)) + \boldsymbol{w}(k) \\ \boldsymbol{y}(k) &= g(\boldsymbol{x}(k)) + \boldsymbol{v}(k) \end{aligned} \tag{1}$$

where $\boldsymbol{x}(k) \in R^n$ is the state vector, $\boldsymbol{u}(k) \in R^m$ is the deterministic input, $\boldsymbol{y}(k) \in R^r$ is the measured output and $\boldsymbol{w}(k-1), \boldsymbol{v}(k)$ are the process and measurement noises which are both assumed to be zero mean Gaussian noises with covariance $\boldsymbol{Q}_{k-1}$ and $\boldsymbol{R}_k$, respectively. The time update and measurement update equations of the EKF are given as:

$$\hat{\boldsymbol{x}}(k|k-1) = f(\hat{\boldsymbol{x}}(k-1), \boldsymbol{u}(k-1)) \tag{2}$$

$$\hat{\boldsymbol{x}}(k|k) = \hat{\boldsymbol{x}}(k|k-1) + \boldsymbol{K}(k)[\boldsymbol{y}(k) - g(\hat{x}(k|k-1))] \tag{3}$$

$$\boldsymbol{P}(k|k-1) = \boldsymbol{F}(k-1)\boldsymbol{P}(k-1|k-1)\boldsymbol{F}^T(k-1) + \boldsymbol{Q}_{\boldsymbol{k-1}} \tag{4}$$

$$\boldsymbol{K}(k) = \boldsymbol{P}(k|k-1)\boldsymbol{H}^T(k)(\boldsymbol{H}(k)\boldsymbol{P}(k|k-1)\boldsymbol{H}^T(k) + \boldsymbol{R}(k))^{-1} \tag{5}$$

$$\boldsymbol{P}(k|k) = (\boldsymbol{I} - \boldsymbol{K}(k)\boldsymbol{H}(k))\boldsymbol{P}(k|k-1) \tag{6}$$

where $\boldsymbol{P}(k|k)$ is the estimation covariance error matrix, $\boldsymbol{K}(k)$is the Kalman gain, $\boldsymbol{I}$ is the appropriately dimensioned identity matrix and Jacobians $\boldsymbol{F}(k), \boldsymbol{H}(k)$ are defined as

$$\boldsymbol{F}(k) = \frac{\delta f}{\delta x}\Big|_{x(k)=\hat{x}(k|k)} \tag{7}$$

$$\boldsymbol{H}(k) = \frac{\delta g}{\delta x}\Big|_{x(k)=\hat{x}(k|k)} \tag{8}$$

A combination of local linear filters is assumed to reduce the computation burden during the numerical derivation of Jacobians. Each of the *M* local filters is based on the local approximation of the process at steady-state operating conditions. The linear state-space model is obtained through Taylor expansion for different operating conditions and has the form of:

$$\begin{aligned} \boldsymbol{x}_i(k+1) &= \boldsymbol{A}_i\boldsymbol{x}_i(k) + \boldsymbol{B}_i\boldsymbol{u}(k) + \boldsymbol{E}_i \\ \boldsymbol{y}_i(k) &= \boldsymbol{C}_i\boldsymbol{x}_i(k) + \boldsymbol{D}_i \end{aligned} \tag{9}$$

The affine parts $\boldsymbol{E}_i, \boldsymbol{D}_i$ result from nonzero steady-state operating conditions and the global model is then a fuzzy combination of these local models

$$\begin{aligned} \boldsymbol{x}(k+1) &= \sum_{i=1}^{M} p_i(\,z(k))(\boldsymbol{A}_i\boldsymbol{x}_i(k) + \boldsymbol{B}_i\boldsymbol{u}(k) + \boldsymbol{E}_i\,) \\ \boldsymbol{y}(k) &= \sum_{i=1}^{M} p_i(z(k))(\boldsymbol{C}_i\boldsymbol{x}_i(k) + \boldsymbol{D}_i\,) \end{aligned} \tag{10}$$

where the membership degrees p_i are based on the actual measurement *z(k)*. For linear model the Kalman time update and measurement update are simplified to:

$$\hat{\boldsymbol{x}}_i(k|k-1) = \mathbf{A}_i\hat{\boldsymbol{x}}_i(k|k-1) + \boldsymbol{B}_i\boldsymbol{u}(k) + \boldsymbol{E}_i \tag{11}$$

$$\hat{\boldsymbol{x}}(k|k) = \hat{\boldsymbol{x}}(k|k-1) + \boldsymbol{K}(k)[z(k) - \boldsymbol{C}_i\hat{\boldsymbol{x}}(k|k-1) - \boldsymbol{D}_i] \tag{12}$$

The linearization of the functions $f(\boldsymbol{x}(k-1), \boldsymbol{u}(k-1))$ and $g(\boldsymbol{x}(k), \boldsymbol{u}(k))$ is accomplished off-line during the modeling process and the Jacobians can be computed using the linearized model as:

$$\boldsymbol{F}(k) = \boldsymbol{A}_i, \boldsymbol{H}(k) = \boldsymbol{C}_i \tag{13}$$

Using the linear model the model predictive controller would exhibit steady – state offset in the presence of plant/model mismatch or unmeasured disturbance due to lack of integral action. In order to introduce integral behavior, the discrete model is augmented with an input disturbance model in order to capture the mismatch between the process and model of the plant [7]:

$$\begin{aligned} \boldsymbol{x}_i(k+1) &= \boldsymbol{A}_i\boldsymbol{x}_i(k+1) + \boldsymbol{B}_i\boldsymbol{u}(k) + \boldsymbol{E}_i + \boldsymbol{B}_d\boldsymbol{d}(k) \\ \boldsymbol{d}_i(k+1) &= \boldsymbol{d}_i(k) \\ \boldsymbol{y}_i(k) &= \boldsymbol{C}_i\boldsymbol{x}_i(k) + \boldsymbol{D}_i \end{aligned} \tag{14}$$

New state vector $\boldsymbol{\xi}_i$ is created and the system matrices are augmented with the input disturbance model:

$$\begin{aligned}
&\boldsymbol{\xi}_i(k+1)=\tilde{\boldsymbol{A}}_i\boldsymbol{\xi}_i(k+1)+\tilde{\boldsymbol{B}}_i u+\tilde{\boldsymbol{E}}_i\\
&\boldsymbol{y}_i(k)=\tilde{\boldsymbol{C}}_i\boldsymbol{\xi}(k+1)+\tilde{\boldsymbol{D}}_i\\
&\boldsymbol{\xi}_i(k)=[\boldsymbol{x}_i^T(k)\,\boldsymbol{d}_i(k)]^T\\
&\tilde{\boldsymbol{A}}_i=\begin{bmatrix}\boldsymbol{A}_i & \boldsymbol{B}_{id}\\ 0 & \boldsymbol{I}\end{bmatrix},\tilde{\boldsymbol{B}}_i=\begin{bmatrix}\boldsymbol{B}_i\\ 0\end{bmatrix},\tilde{\boldsymbol{E}}_i=\begin{bmatrix}\boldsymbol{E}_i\\ 0\end{bmatrix}\\
&\tilde{\boldsymbol{C}}_i=\begin{bmatrix}\boldsymbol{C}_i & 0\end{bmatrix},\tilde{\boldsymbol{D}}_i=\boldsymbol{D}_i
\end{aligned} \tag{15}$$

3 Multiple Model Predictive Control

The control structure for the model predictive control [8] with multiple local models is shown in the block diagram in Fig. 1. Estimates from multiple KF are weighted using fuzzy membership functions and current measured output in the Fuzzy State Estimation block. The control signal optimization is accomplished in the MPC block and the first control signal of the optimized control sequence is applied to the plant.

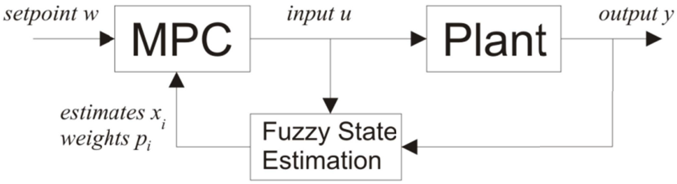


Fig. 1 Multiple model control scheme

The state-space model based predictive control is based on the linear time-invariant model that was obtain using the weights p_i, augmented state-space model parameters $\boldsymbol{A}_i, \boldsymbol{B}_i, \boldsymbol{C}_i, \boldsymbol{D}_i, \boldsymbol{E}_i$ and augmented state vectors $\boldsymbol{\xi}_i$:

$$\begin{aligned}
&\boldsymbol{\xi}(k+1)=\tilde{\boldsymbol{A}}\boldsymbol{\xi}(k+1)+\tilde{\boldsymbol{B}}\boldsymbol{u}+\tilde{\boldsymbol{E}}\\
&\boldsymbol{y}(k)=\tilde{\boldsymbol{C}}\boldsymbol{\xi}(k+1)+\tilde{\boldsymbol{D}}
\end{aligned} \tag{16}$$

The model of the process is obtained at every sampling interval and its parameters are used for the entire prediction horizon Hp. The N_p-steps ahead output prediction can be formulated based on the knowledge of the actual state $\boldsymbol{x}(k)$ and the vector future control signal $\boldsymbol{u}$. The computation of a control signal of model predictive controller is based on minimization of the quadratic criterion:

$$J_{MPC}=\left(\hat{\boldsymbol{y}}-\boldsymbol{w}\right)^T\left(\hat{\boldsymbol{y}}-\boldsymbol{w}\right)+\lambda\triangle\boldsymbol{u}^T\triangle\boldsymbol{u} \tag{17}$$

subjected to constraints:

$$\boldsymbol{A}\boldsymbol{u}\le\boldsymbol{b} \tag{18}$$

where $\hat{y}(k+j)$ is a j steps ahead prediction of the systems output, $w(k+j)$ is a future reference trajectory and λ is the weighting factor. The minimization of the criterion can be transformed into a quadratic programming problem:

$$J_{MPC} = \boldsymbol{u}^T \boldsymbol{H} \boldsymbol{u} + \boldsymbol{f} \boldsymbol{u} \tag{19}$$

and solved numerically at each sampling instant.

4 Simulation Example

The efficiency of the proposed algorithm is illustrated on the model of the pH neutralization process. The simulated system consists of a continuous stirred tank reactor (CSTR) in which neutralization reaction between a strong acid (HA) and a strong base (BOH) takes place in the presence of a buffer (BX). The system has three states, single output and single input.

$$x_1 = [A^-], x_2 = [B^+], x_3 = [X^-], y = pH, u = q_B \tag{20}$$

where$[A^-], [B^+], [X^-]$ are acid, base and buffer concentrations, respectively. The term q_B represents the flow rate of the base. The scheme of the CSTR is depicted in Fig. 2. The process dynamics is given by the following set of differential equations:

$$\dot{x}_1 = \frac{q_A}{V}\left(x_{1,i} - x_1\right) - \frac{q_B}{V} x_1 \tag{21}$$

$$\dot{x}_2 = -\frac{q_A}{V} x_2 + \frac{q_B}{V}\left(x_{2,i} - x_2\right) \tag{22}$$

$$\dot{x}_3 = -\frac{q_A}{V} x_3 + \frac{q_B}{V}\left(x_{3,i} - x_3\right) \tag{23}$$

The pH value can be determined using the implicit equation:

$$[H^+] + x_2 + x_3 - x_1 - \frac{K_W}{[H^+]} - \frac{x_3}{1 + \frac{(K_X)[H^+]}{K_W}} = 0 \tag{24}$$

where $pH = \log_{10}\left[H^+\right]$ and K_w, K_x are the dissociation constants of water and buffer, respectively.

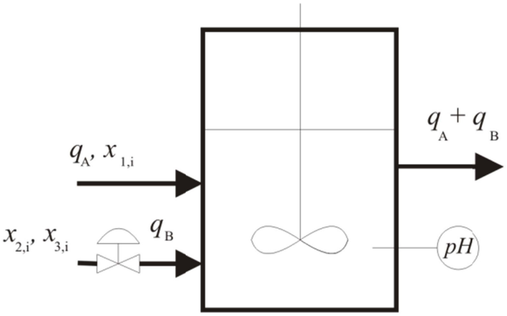


Fig. 2 pH neutralization process

Table 1 Model Parameters

symbol	parameter	value
$x_{1,i}$	acid inlet concentration	1.2×10^{-3} mol/L
$x_{2,i}$	base inlet concentration	2.0×10^{-3} mol/L
$x_{3,i}$	buffer inlet concentration	2.5×10^{-3} mol/L
K_x	buffer dissociation const.	10^{-7}mol/L
K_w	water dissociation const.	10^{-14}mol^2/L^2
V	reactor volume	2.5L

The system parameters used in this work were taken from [9] and are summarized in Table 1. For the state estimation a function $y = g(x)$ must be differentiable with respect to x.

The Equation (24) can be rewritten as a third order polynomial:

$$[H^+]^3 + \left[\frac{K_w}{K_x} + x_3 + x_2 - x_1\right][H^+]^2 + (x_2 - x_1 - K_x)\frac{K_w}{K_x}[H^+] - \frac{K_w^2}{K_x} = 0 \quad (25)$$

This polynomial has only one root with physical meaning. By introducing auxiliary variables and formula for root of the third order polynomial the relation $y = g(x)$ can be obtain:

$$a = 1, b = \frac{K_w}{K_x} + x_3 + x_2 - x_1, c = (x_2 - x_1 - K_x)\frac{K_w}{K_x}, d = -\frac{K_w^2}{K_x}$$
$$p = -\frac{b}{3a}, \; q = p^3 + \frac{bc - 3ad}{6a^2}, r = \frac{c}{3a} \quad (26)$$

The pH value is then given as:

$$pH = -log_{10}\left(p + \left(q + \sqrt{q^2 + (r - p^2)^3}\right)^{1/3} + \left(q - \sqrt{q^2 + (r - p^2)^3}\right)^{1/3}\right) \quad (27)$$

The output equation is clearly strongly nonlinear. The titration curve and the gain variation that illustrate the nonlinearity of the pH neutralization process are depicted in Fig. 3.

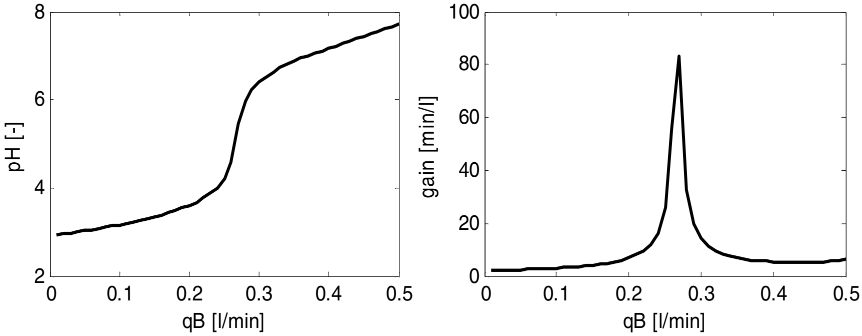


Fig. 3 Titration curve and gain variation of pH neutralization process

The sampling of the estimation and control schemes was set to 0.5 min due to the dynamics of the process and constraints of the process input are assumed to be $0 \leq u(k) \leq 0.5$. The concentrations $x_1(k), x_2(k), x_3(k)$ cannot be measured online so they must be estimated with KF which uses input signal (flow-rate) and output signal (pH value) for their computation. Six fuzzy sets with triangular membership functions were used for approximation of the nonlinear process as shown in Fig. 4. The location of the models was obtained using C-means clustering. At these operation modes the nonlinear process was linearized to obtain parameters of local models $\boldsymbol{A}_i, \boldsymbol{B}_i, \boldsymbol{C}_i, \boldsymbol{D}_i, \boldsymbol{E}_i$ and also the steady-state values of states and outputs $x_{is,y}, y_{is}$. The fuzzy model is a good approximation of the process as presented in Fig. 4 which shows both steady-state characteristic of process and fuzzy model. The fuzzy KF bank was constructed using these local models and augmented with the disturbance model with $B_{id} = B_i$.

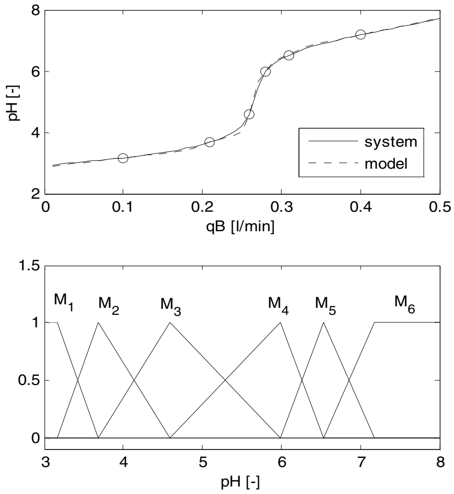


Fig. 4 Steady-state characteristic and distribution of membership functions

Each Kalman filter produces estimate of the states $x_{i1}(k), x_{i2}(k), x_{i3}(k)$ and disturbance $d_i(k)$. The initial state estimates of each KF were set to linearization steady-state values $\hat{x}_i(0) = x_{is}$ and estimate of disturbance to zero $\hat{d}_i(0) = 0$. The process and measurement noise covariances were set to:

$$Q_{k-1} = 0.1I, R_k = 1 \tag{28}$$

The prediction horizon was set to 10 samples as a result of using different values and comparing control performances. At each sampling instant linear model of the process is obtained and used for entire prediction horizon

$$\begin{aligned} &\boldsymbol{A} = \sum_{i=1}^{M} p_i(z(k))\boldsymbol{A}_i, \quad \boldsymbol{B} = \sum_{i=1}^{M} p_i(z(k))\boldsymbol{B}_i, \quad \boldsymbol{E} = \sum_{i=1}^{M} p_i(z(k))\boldsymbol{E}_i, \\ &\boldsymbol{C} = \sum_{i=1}^{M} p_i(z(k))\boldsymbol{C}_i, \boldsymbol{D} = \sum_{i=1}^{M} p_i(z(k))\boldsymbol{D}_i, \quad \boldsymbol{\xi} = \sum_{i=1}^{M} p_i(z(k))\boldsymbol{\xi}_i, \end{aligned} \tag{29}$$

To account for high variation of gain of the process the weighting factor λ used in the predictive control cost function is also weighted using membership functions:

$$\lambda(k) = \sum_{i=1}^{M} p_i\,(z(k))0.3gain_i^2 \tag{30}$$

where the gain of the local model M_i is computed:

$$gain_i = C_i(I - A_i)^{-1}B_i \tag{31}$$

All algorithms were implemented in MATLAB/Simulink and the control algorithm based on the linear model uses the quadratic programming procedure. Measurement noise is added to the output of pH model to simulate the imperfections of

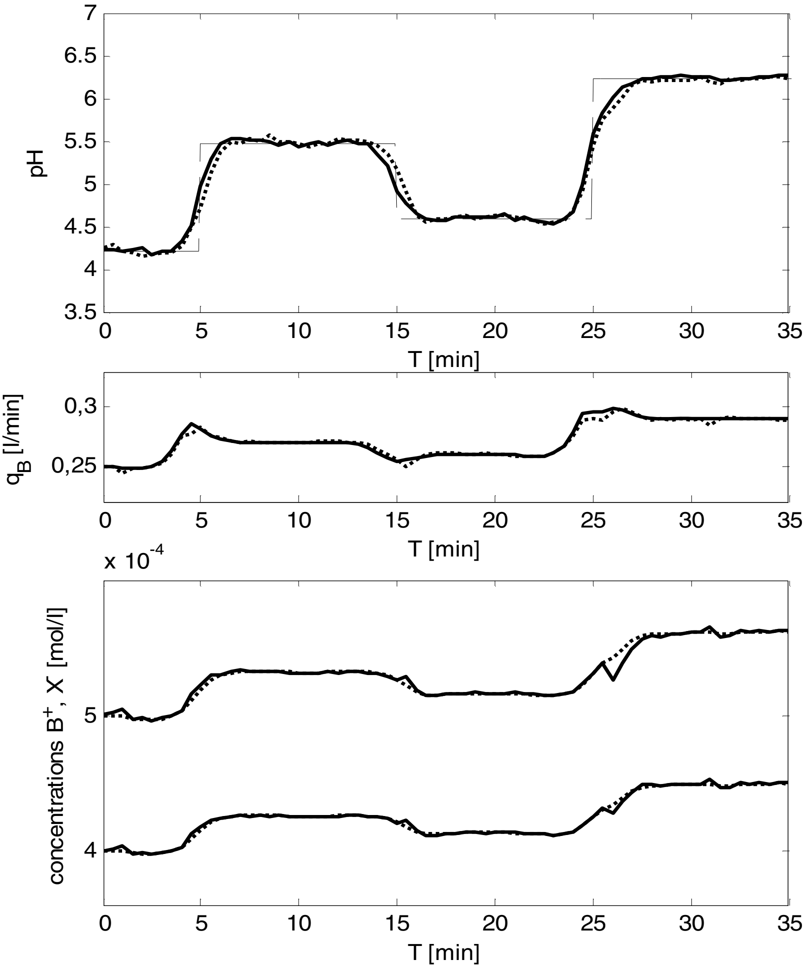


Fig. 5 Performance comparison (dotted – fuzzy MPC, solid – Nonlinear MPC, dashed - setpoint) and states estimation courses of (dotted – true states, solid – their estimates)

the measurement device. The performance of the proposed fuzzy KF and MPC controller is compared with the performance of the nonlinear MPC controller which uses the same weighting factor (30) in the reference tracking test. The states for the nonlinear controller are assumed to be measured and thus represent "optimal performance" for given weighting factor. The comparison is shown in Fig. 5. The third graph of the figure shows the courses of states and their estimates. Comparable performance can be obtained with multiple-model MPC but without the computationally demanding nonlinear optimization.

5 Conclusion

In this paper, MPC based on fuzzy estimation of the states and unmeasured disturbances was applied to simulated pH neutralization process. Using fuzzy Kalman filters, the linearization of the nonlinear process is computed off-line and thus numerical derivation at each sampling instant for computation of Jacobians of the Extended Kalman Filter is not necessary. The process model of each Kalman Filter is augmented with the disturbance model in order to remove steady-state error in case of model-plant mismatch or unmeasured disturbance. Using multiple local models, the on-line nonlinear optimization can be avoided and simple quadratic programming problem is solved at each sampling interval. It is demonstrated that control performance of the proposed fuzzy estimation scheme with MPC based on multiple models is comparable to the performance obtained when a computationally demanding nonlinear optimization procedure is used online at each sampling instant within a nonlinear MPC controller.

Acknowledgments. This work was supported by European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089.

References

1. Henson, M.A., Seborg, D.E.: Adaptive Nonlinear Control of a pH Neutralization Process. IEEE Transactions on Control Systems Technology 2, 169–182 (1994)
2. Akesson, B.M., Toivonen, H.T., Waller, J.B., Nystrom, R.H.: Neural network approximation of a nonlinear model predictive controller applied to a pH neutralization process. Computers & Chemical Engineering 29, 323–335 (2005)
3. Gomez, J.C.: Wiener model identification and predictive control of a pH neutralization process. IEE Proceedings – Control Theory and Application 151, 329–338 (2004)
4. Oblak, S., Skrjanc, I.: Nonlinear Model-Predictive Control of Wiener-type Systems in Continuous-time Domain Using a Fuzzy–system Function Approximation. In: The IEEE International Conference on Fuzzy Systems, pp. 2203–2208. IEEE Press, New York (2006)
5. Mahmoodi, S., Poshtan, J., Jahed-Motlagh, M.R., Montazeri, A.: Nonlinear model predictive control of a pH neutralization process based on Wiener-Laquerre model. Chemical Engineering Journal 146, 328–337 (2009)

6. Bagheri, P., Mardalou, V., Fatehi, A.: Multiple Model Predictive Control of Multivariable pH Process using Adaptive Weighting Matrices. In: The 18th IFAC World Congress, pp. 12366–12371. Curran Associates, New York (2011)
7. Kuure-Kinsey, M., Bequette, B.W.: Multiple Model Predictive Control of Nonlinear Systems. In: Magni, L., Raimondo, D.M., Allgöwer, F. (eds.) Nonlinear Model Predictive Control. LNCIS, vol. 384, pp. 153–165. Springer, Heidelberg (2009)
8. Camacho, E.F., Bordons, C.: Model Predictive Control. Springer, Heidelberg (2004)
9. Biagiola, S.I., Figueroa, J.L.: State Estimation in Nonlinear Processes- Application to pH Process Control. Industrial & Engineering Chemical Research 41, 4777–4785 (2003)

State Observers for Model Predictive Control

Petr Chalupa, Peter Januška, and Jakub Novák

Abstract. This paper deals with state observers with respect to their usage in the model predictive control (MPC) based on state space model of the controlled system. In case of immeasurable states a state observer (filter) is used to calculate current states in each control step. The paper is especially focused to finite impulse filters (FIR) as these filters do not require knowledge of initial state - contrary to infinite impulse response (IIR) filters. Several linear filters are tested and compared with proposed filters based on quadratic and linear programming. Different filter lengths (horizons) were tested to investigate filters' performance. Filters were tested in very noisy conditions to evaluate filter robustness and therefore its usability in real-time deployment. The simulations were carried out using data from a real-time laboratory (Amira DR300 Servo system). All the measurements and simulations were carried out in MATLAB/Simulink environment.

Keywords: State observer, FIR filter, IIR filter, Model predictive control.

1 Introduction

Model predictive control (MPC) is very popular and successful technique for control of technological processes. The control algorithm is based on model of the controlled system [1]. The model is used to predict future output courses of the controlled system on the basis of current state of the system and future course of the system inputs. There are many types of models which are implemented in MPC. The model can be divided to categories according to various criteria: linear vs. nonlinear; time invariant vs. time varying; state space vs. input-output, etc.

This paper is focused on linear time invariant (LTI) state space models of the controlled plant. This model category can be used even for nonlinear or time-varying systems when linearizing the system in some working point. Then the LTI model provides good representation of the controlled system in a neighborhood of

Petr Chalupa · Peter Januška · Jakub Novák
Faculty of Applied Informatics, Tomas Bata University in Zlin,
nám. T. G. Masaryka 5555, 760 01 Zlín, Czech Republic
e-mail: {chalupa,jnovak}@fai.utb.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 295 – 303.
DOI: 10.1007/978-3-319-00542-3_30

the working point. State space models can also represent biased control systems by adding a uncontrollable state to the model. This state represents a bias which is added to the output of the original system. Time-varying processes can be model by linear systems with on-line identification [2].

Usage of state space representation of the model is very useful and popular. One of its main advantages is general approach to computation of all signals in the control circuits regardless of the type of the model – the same approach is valid for multi-input multi-output systems, single-input single-output systems, systems with measurable disturbance, systems without measurable disturbances, etc.

In case of state space MPC is used, computation of future courses of control signals is based on knowledge of current state of the controlled plant. The current states are used as parameters of a criterion which is to be minimized by future course of the control signals. Thus, knowledge of the current states is crucial for computation of the control signal. In general, the states are not measurable and must be computed using input-output data of the controlled system. A control circuit block, which is used to compute current state, is referred to as state observer or state filter. The term "observer" is more common in the control systems area while the term "filter" is more used in the signal processing. The aim of the state observer is to reconstruct current state on the basis of previous inputs and outputs. The observers can be divided into two classes: IIR (Infinite Impulse Response) and FIR (Finite Impulse Response). The IIR observers require knowledge of the system states at the beginning of the observer horizon while FIR observers don't require initial state. Well-known IIR state observers are Kalman filter for stochastic systems [3] and Luenberger observer for deterministic systems [4]. The IIR filters in their recursive form are popular in the area of control systems but their convergence is often not guaranteed by design. In this case the convergence has to be verified for each application.

Non-recursive FIR filters are popular in signal processing. They are characterized by guarantee of stability, robustness to temporary changes of system parameters, etc. The FIR filters are using, as well as predictive control itself, the receding horizon principle. The current state estimates are computed from previous inputs and outputs of the system on a finite horizon. This horizon is moved forward each sample step. Moreover, linearity, filter error minimization and independence of the system state at the beginning of the horizon are incorporated into filter by design [5].

This paper is mainly focused on FIR observers. Several observers with a good performance [6] are compared with proposed observers based on H_1, H_2 and H_∞ norms. These observers are solved by linear programming and quadratic programming. In addition, a very simple state observer is derived. All observers are tested on a model derived from a nonlinear first order system Amira DR300 [7], [8] with a high level of noise.

The paper is organized as follows. Section 2 presents the observers which were derived and compared. Comparison of the observers is presented in Section 3 and summarized in Conclusion.

2 Observers

The observers studied in this paper are designed for discrete LTI state space system without direct feed-through. The system is described by the following equation:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + Gw_k \\ y_k &= Cx_k + v_k \end{aligned} \tag{1}$$

where x_k is the state vector, u_k is the vector of inputs, y_k is the vector of outputs of the system. Symbols w_k and v_k represents disturbances which are assumed to be a white noise. Finally, the symbol k represents current time sample.

Most of FIR observers, which will be described in following sections, use matrix equation expressing relation of recent inputs, recent outputs and current states. If the system matrix A is nonsingular, it can be derived from (1):

$$\begin{aligned} x_{k-1} &= A^{-1}x_k - A^{-1}Bu_{k-1} - A^{-1}Gw_{k-1} \\ y_{k-1} &= Cx_{k-1} + v_{k-1} = CA^{-1}x_k - CA^{-1}Bu_{k-1} - CA^{-1}Gw_{k-1} + v_{k-1} \end{aligned} \tag{2}$$

Recursive application of equation (2) over horizon of N previous samples leads to the following equation:

$$Y_{k-1} = \bar{C}_N x_k + \bar{B}_N U_{k-1} + \bar{G}_N W_{k-1} + V_{k-1} \tag{3}$$

where

$$\begin{aligned} Y_{k-1} &= \begin{bmatrix} y_{k-N}^T & y_{k-N+1}^T & \cdots & y_{k-1}^T \end{bmatrix}^T & U_{k-1} &= \begin{bmatrix} u_{k-N}^T & u_{k-N+1}^T & \cdots & u_{k-1}^T \end{bmatrix}^T \\ W_{k-1} &= \begin{bmatrix} w_{k-N}^T & w_{k-N+1}^T & \cdots & w_{k-1}^T \end{bmatrix}^T & V_{k-1} &= \begin{bmatrix} v_{k-N}^T & v_{k-N+1}^T & \cdots & v_{k-1}^T \end{bmatrix}^T \end{aligned} \tag{4}$$

and matrices $\bar{C}_N$, $\bar{B}_N$ and $\bar{G}_N$ are defined as:

$$\bar{C}_N = \begin{bmatrix} CA^{-N} \\ CA^{-N+1} \\ CA^{-N+2} \\ \vdots \\ CA^{-1} \end{bmatrix} \quad \bar{B}_N = -\begin{bmatrix} CA^{-1}B & CA^{-2}B & \cdots & CA^{-N}B \\ 0 & CA^{-1}B & \cdots & CA^{-N+1}B \\ 0 & 0 & \cdots & CA^{-N+2}B \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & CA^{-1}B \end{bmatrix} \quad \bar{G}_N = -\begin{bmatrix} CA^{-1}G & CA^{-2}G & \cdots & CA^{-N}G \\ 0 & CA^{-1}G & \cdots & CA^{-N+1}G \\ 0 & 0 & \cdots & CA^{-N+2}G \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & CA^{-1}G \end{bmatrix} \tag{5}$$

The aim of the observer is to compute current state x_k. This is usually done by minimizing a criterion. If some of the states are measurable and some have to determined by observer, this notation can be easily used. State (and subsequently matrices A, B, and C) can be reordered to the following form:

$$x_k = \begin{bmatrix} x_{k,imm} \\ x_{k,m} \end{bmatrix} \tag{6}$$

where $x_{k,imm}$ are immeasurable states and $x_{k,m}$ are measurable states. The matrix $\bar{C}_N$ is divided into two parts corresponding to the two parts of x_k. The term $\bar{C}_N$ in the equation (3) is superseded by:

$$\bar{C}_N x_k = \begin{bmatrix} \bar{C}_{N,imm} & \bar{C}_{N,m} \end{bmatrix} \begin{bmatrix} x_{k,imm} \\ x_{k,m} \end{bmatrix} = \bar{C}_{N,imm} x_{k,imm} + \bar{C}_{N,m} x_{k,m} \tag{7}$$

The second term $\bar{C}_{N,m} x_{k,m}$ is known and thus only $x_{k,imm}$ states are to be determined.

2.1 Simple Observer

The state of the first order system can be determined directly from its current output.

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k \end{aligned} \qquad \hat{x}_k = C^{-1} y_k \tag{8}$$

where $\hat{x}_k$ is an estimation of the current state.

As this state observer is using just one value to reconstruct the state, there is no robustness with respect to noise. This observer is included just as a reference for comparison and is not intended for real world application.

2.2 Dual IIR Observer

A standard Kalman filter [6] of system (1) can be written as

$$\hat{x}_{k+1} = A\hat{x}_k + AP_k C^T \left(CP_k C^T + R_v \right)^{-1} \left(y_k - C\hat{x}_k \right) \tag{9}$$

with Riccati equation

$$P_{k+1} = A\left(I + P_k C^T R_v^{-1} C \right)^{-1} P_k A^T + GQ_w G^T \tag{10}$$

where I is identity matrix and Q_w is a diagonal covariance matrix of noise signal w_k in equation (1) and (2). The structure of the observer is similar to computation of control signal in LQ control design. Due to this duality, the observer is referred to as “dual IIR observer”.

2.3 MVFIR Observer

The MVFIR observer [6] was designed to minimize the estimation error variance. It is based on equations (3) – (5). Term $\bar{G}_N W_{k-1} + V_{k-1}$ in (3) is represented by white noise with *N* dimensional covariance matrix Ξ_N . Noises w_k and v_k are characterized by covariance matrices Q_w and R_v. The matrix Ξ_N is defined as:

$$\Xi_N = \bar{G}_N Q_N \bar{G}_N^T + R_N \tag{11}$$

where Q_N and R_N are *N*-dimensional diagonal matrices with Q_w a R_v on the main diagonal.

General FIR observer defined on horizon [*k-N*, *k*] is described by the following equation:

$$\hat{x}_k = HY_{k-1} + LU_{k-1} \tag{12}$$

where *H* and *L* are observer gain matrices. Then the state estimation can be expressed as:

$$\hat{x}_k = H\left(\bar{C}_N x_k + \bar{B}_N U_{k-1} + \bar{G}_N W_{k-1} + V_{k-1}\right) + LU_{k-1} \tag{13}$$

From the unbiased condition $E\left[\hat{x}_k\right] = E\left[x_k\right]$ follows:

$$E\left[\hat{x}_k\right] = H\bar{C}_N E\left[x_k\right] + \left(H\bar{B}_N + L\right)U_{k-1} \tag{14}$$

and hence

$$H\bar{C}_N = I \qquad H\bar{B}_N = -L \tag{15}$$

Substitution of (15) into (13) leads to estimation error equation.

$$e_k = \hat{x}_k - x_k = H\bar{G}_N W_{k-1} + HV_{k-1} \tag{16}$$

From a set of all possible matrices *H*, the H_B is selected. The H_B minimizes the minimum variance criterion and can be computed as

$$H_B = \left(\bar{C}_N^T \Xi_N^{-1} \bar{C}_N\right)^{-1} \bar{C}_N^T \Xi_N^{-1} \tag{17}$$

and the MVFIR state observer is defined as

$$\hat{x}_k = H_B\left(Y_{k-1} - \bar{B}_N U_{k-1}\right) \tag{18}$$

2.4 L₂-E FIR Observer

The L_2-E FIR observer derivation is based on computation of gain matrices *H* and *L* as defined by (12) with respect to (15). The objective is to minimize the worst-case gain between the estimation error and the disturbance [6]:

$$\min_{H,L} \max_{w_k} \frac{\left[x_k - \hat{x}_k\right]^T \left[x_k - \hat{x}_k\right]}{\sum_{i=1}^{N} w_{k-i}^T w_{k-i}} \tag{19}$$

Then the estimation of the current states can be calculated as

$$\hat{x}_k = \left(\bar{C}_N^T \Xi_N^{-1} \bar{C}_N\right)^{-1} \bar{C}_N^T \Xi_N^{-1} \left(Y_{k-1} - \bar{B}_N U_{k-1}\right) \qquad \Xi_i = \bar{G}_i \bar{G}_i^T \tag{20}$$

Further details can be found in [6].

2.5 FIR H₂ Observer

The FIR H$_2$ observer is proposed for systems whose noise input matrix G is unknown or zero. Then the model of the system has the form

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + v_k \end{aligned} \tag{21}$$

Recursive application of equation (2) over horizon of N previous samples leads to the following equation:

$$Y_{k-1} = \bar{C}_N x_k + \bar{B}_N U_{k-1} + V_{k-1} \tag{22}$$

where vectors and matrices are defined by (4) and (5).

The objective is to minimize sum of squares of differences between measured output and estimated output:

$$\min_{\hat{x}_k} J_k \qquad J_k = \left(Y_{k-1} \text{-} \hat{Y}_{k-1}\right)^{\mathrm{T}} \left(Y_{k-1} \text{-} \hat{Y}_{k-1}\right) \qquad \hat{Y}_{k-1} = \bar{C}_N \hat{x}_k + \bar{B}_N U_{k-1} \tag{23}$$

This optimization problem can be solved directly by matrix inversion where gain matrix is obtained and solution has the form similar to (18) or (20). It is also possible to consider (24) as a quadratic programming problem. This approach allows application of known restrictions of the states directly to their computation (e.g. the state known to be non-negative) [9], [10]. Both versions were tested. The observer version using quadratic programming is referred to as “FIR H$_2$ QP”.

2.6 FIR H₁ Observer

The FIR H$_1$ observer is based on similar approach as the FIR H$_2$ as defined by (22). The difference between the observers consists in usage of L_1 norm instead of the L_2 norm. The criterion to be minimized sums absolute values of differences between measured and estimated output:

$$\min_{\hat{x}_k} J_k \qquad J_k = \underbrace{[1 \quad 1 \quad \ldots \quad 1]}_{N \text{ elements}} \left|Y_{k-1} \text{-} \hat{Y}_{k-1}\right| \qquad \hat{Y}_{k-1} = \bar{C}_N \hat{x}_k + \bar{B}_N U_{k-1} \tag{24}$$

The optimization problem can be considered a linear programming task. Then known restriction of the states can be used as conditions for linear programming algorithm.

2.7 FIR H_∞ Observer

The FIR H_∞ observer is based on similar approach as the FIR H_2 as defined by (22). But the criterion used is based on minmax criterion – L_∞ norm. The maximal absolute value of difference between measured and estimated output is to be minimized:

$$\min_{\hat{x}_k} J_k \qquad J_k = \max\left|Y_{k-1}\text{-}\hat{Y}_{k-1}\right| \qquad \hat{Y}_{k-1} = \bar{C}_N \hat{x}_k + \bar{B}_N U_{k-1} \tag{25}$$

Likewise FIR H_1, the optimization problem can be considered a linear programming task and allows incorporating of restrictions directly into the optimization algorithm.

3 Comparison of Observer Performance

The observers presented in previous sections were compared using a simple first order system whose states were observed. The system was derived from measurements of Amira DR300 Servo system [7]:

$$\begin{aligned} x_{k+1} &= 0.7372 x_k + \begin{bmatrix} 6.279 & -3.31 & -0.7746 \end{bmatrix} \begin{bmatrix} u_k \\ l_k \\ d_k \end{bmatrix} \\ y_k &= 0.6098 x_k + v_k \end{aligned} \tag{26}$$

where u_k is a control input, l_k is an external load, which was set to zero, and d_k is direction of rotations which were positive during the experiment (d=1).

Many experiments were carried out with different settings of the observers as well as the noise level. Representative results are presented for a very noisy environment. The system input and output courses are presented in Fig. 1. The output without noise is presented just as a reference, only output with noise was used in computations.

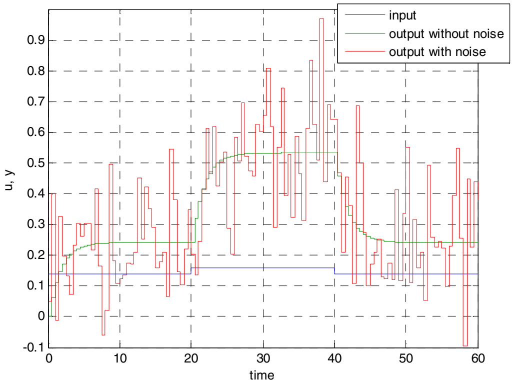


Fig. 1. Course of input and output signal

Crucial parameter of the observers was length of the horizon. For FIR observers this was the only parameter. The courses of reconstructed states for *N*=2 and *N*=10 are presented in Fig. 2 and Fig. 3 respectively.

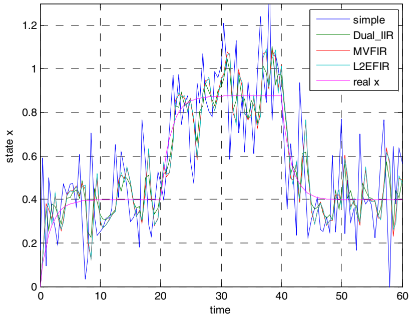


Fig. 2 Observers performance for small horizon of *N*=2 (*N*=1 in case of simple observer)

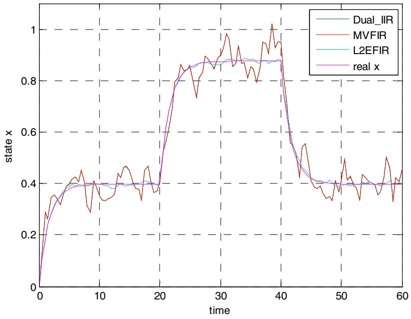


Fig. 3 Performance of the state observers using longer horizon of *N*=10

The *N*=2 case represents extremely small horizon while *N*=10 is a long horizon for this type of system. The horizon was applied even to Dual IIR observer to obtain results corresponding to the similar memory usage for all observers. The results of simple observer are shown only in Fig. 2 as it uses horizon length *N*=1.

Table 1 Performance of the state observers

observer	S_x•10^3 (N=2)	S_x•10^3 (N=10)	time (100 000 runs)	time (100 runs)
simple	34.3318	NA	1.0873	
Dual IIR	6.4263	2.5729	5.5864	
MVFIR	11.9860	2.5705	5.8106	
L2EFIR	11.7121	0.0481	6.5050	
FIR_H2	11.7113	0.0484	7.7714	
FIR_H2_qp	11.7113	0.0484		2.4821
FIR H_1	17.9226	0.0838		3.9258
FIR $H_.$	12.0058	0.2341		4.1346

The courses of Dual IIR and MVFIR observers are almost identical for horizon length of N=10. The performance of individual observers is presented in Table 1.

The accuracy of individual observers was compared using quadratic criterion. The criterion calculated as a sum of squares of differences between observed state $\hat{x}_k$ and actual state x_k and this sum is divided by number of samples.

Table 1 also summarizes time demands of individual observers. More important than absolute values are relations between them.

4 Conclusion

The paper presented several state observers which can be used in model predictive control based on state space linear time invariant model of the system. Special attention was paid to FIR observers. The FIR observers are nowadays less common in control design area comparing to IIR observers, but their properties are promising for future deployment. Observers based on Minimum variance, L_2-E norm were compared with proposed H_1, H_2, and H_∞ approaches. The proposed observers demonstrated good performance, i.e. accuracy of the estimated state. On the other hand, if they are implemented using quadratic or linear programming, the time (i.e. processor power) demands are several hundred times greater comparing to direct matrix multiplication. This property should be taken into account when designing a real time MPC controller.

Acknowledgments. This work was supported by the European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089.

References

1. Camacho, E., Bordons, C.: Model predictive control, 2nd edn. Springer, London (2007)
2. Bobál, V., Böhm, J., Fessl, J., Macháček, J.: Digital Self-tuning Controllers: Algorithms, Implementation and Applications. Springer, London (2005)
3. Kalman, R.E., Bucy, R.S.: A new approach to linear filtering and prediction problems. Transactions of the ASME–Journal of Basic Engineering 82, 35–45 (1960)
4. Luenberger, D.G.: Introduction to Dynamic Systems, Theory, Models, and Applications. John Wiley & Sons, New York (1979)
5. Song, I.Y., Kim, D.Y., Shin, V., Jeon, M.: Receding horizon filtering for discrete-time linear systems with state and observation delays. IET Radar, Sonar 6(4), 263–271 (2012)
6. Kwon, W., Han, S.: Receding horizon control: model predictive control for state models. Springer, London (2005)
7. DR300: Laboratory Setup Speed Control with Variable Load, Amira, Duisburg (2000)
8. Bobál, V., Chalupa, P., Kubalčík, M., Dostál, P.: Self-Tuning Predictive Control of Nonlinear Servo-Motor. Journal of Electrical Engineering (2010)
9. Maciejowski, J.M.: Predictive control: with constraints, 1st edn. Prentice Hall, Harlow (2002)
10. Goodwin, G.C., de Dona, J., Seron, M.: Constrained control and estimation: an optimisation approach. Springer, London (2005)

Characteristics of the Chen Attractor

Petra Augustová and Zdeněk Beran

Abstract. Within the paper a mathematical representation of the so-called Chen model is described as a particular parametric three-dimensional chaotic dynamical system, i.e. a system of three nonlinear differential equations evolving in time. The main aim of this paper is to find for the Chen system the properties that are known for the Lorenz system and its famous Lorenz attractor. First, the integrals of motion are derived for some parameters of the Chen system. The integrals of motions play an important role in physics, e.g. for conservation laws. Next, the shape of the global attractor of this system is approximated by volumes that contain the attractor. The shape predicts the future behavior of the system. To obtain these results, the already proved fact that the Chen system is a continued transition of the Lorenz system is used. According to our knowledge, the same approach of shifting the known facts about the Lorenz system to a new dynamical system, the Chen system in this context, has not been presented yet.

1 Introduction

In 1963, Edward Lorenz described a simple mathematical model for atmospheric convection [11]. Nowadays, it is the most common example of chaotic attractor that appears for certain values of parameters and initial values. The Lorenz equations also appear in models for lasers, dynamos, thermosyphons, electric circuits, chemical reactions and others. Of course, the Lorenz system serves as subject of study in the mathematical theory of dynamical systems and chaos. The references are so numerous in different areas of research and applications, that we omit them and refer the reader to its own search according to its own interests.

Petra Augustová · Zdeněk Beran
Institute of Information Theory and Automation of the Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 8, CZ-182 08 Prague 8
e-mail: {augustova,beran}@utia.cas.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 305–312.
DOI: 10.1007/978-3-319-00542-3_31

In the paper [5] and also in [6] the following general Lorenz system is studied:

$$\begin{aligned}\dot{x} &= \sigma(y-x)\\ \dot{y} &= rx-y-xz\\ \dot{z} &= xy-bz\end{aligned} \tag{1}$$

where $x(t), y(t)$ and $z(t)$ are variables and σ, b and r are positive parameters. It is well known that for $\sigma = 10, r = 28, b = 8/3$, the Lorenz system has chaotic solutions and the set of chaotic solutions make up the famous Lorenz attractor. When plotted, it resemble a butterfly or figure eight, see fig. 1 (a).

In 1999, Chen found a chaotic attractor that is similar but nonequivalent to the Lorenz attractor, see [4] and [14]. A parametric family of three-dimensional chaotic dynamical systems that contains the chaotic Lorenz and Chen system for boundary values of the parameter is studied in [12] and further developed in [7]. The continued transition (or we can also say homotopy) from Lorenz to Chen presented there is maybe surprisingly simple and confirms our idea of similar characteristics of both systems in general form, although the Chen and Lorenz attractors have different structure and properties (see [14]).

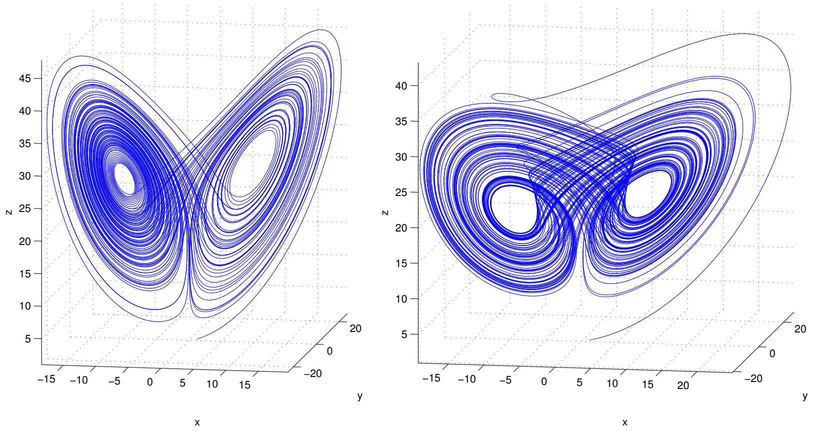
The general Chen system is the following:

$$\begin{aligned}\dot{x} &= \sigma(y-x)\\ \dot{y} &= rx+sy-xz\\ \dot{z} &= xy-bz\end{aligned} \tag{2}$$

where $x(t), y(t)$ and $z(t)$ are variables and σ, b, r and s are real parameters. For parameters values $\sigma = 36, r = 0, s = 20$ and $b = 3$ we obtain fig. 1 (b). It is clear form the figures that there is a certain similarity of both systems.

As already stated, the Lorenz attractor and the Chen attractor are topologically different as it has been proved also by different methods, see [16]. On the other side, the homotopy equivalence of both systems as showed in the already cited literature, is a powerful tool to exploit the homotopy in order to derive some unknown hardly obtainable results of one system on the base of the known results of the homotopically equivalent system.

The integral invariants play a very important role in physics, since they represent conservation laws. The hunt for integral invariants is centuries long story. It was namely the domain of physicists. Nowadays, the integral invariants enjoy the theory of Lie (semi)groups, see, e.g. [10], where the polynomial integral invariants are derived for the case of Chen systems and Lü systems. The other direction to obtain the polynomial integral invariants is through Darboux polynomials. That approach is connected with algebraic integrability and is classical and difficult. It can be traced to Darboux or Poincaré and the theory of the algebraic surfaces are the main theoretical source here, see, e.q., ([15], [3], [8], [9]). The main problem of that approach are tedious computations that can be very often finalized only with help of symbolic software.



(a) (b)

Fig. 1 Attractors plotted in Matlab [13]: (a) Lorenz attractor (b) Chen attractor

The other area of our interest is the geometrical shape of the global attractor. There are considerably many results concerning the topological shape of the global attractor that are based on the conception of so-called templates, for the case of Lorenz-type systems, namely the Lorenz attractor and the Chen attractor, see, e.g. [1]. The analysis of the case of the geometrical shape of the attractors is, based on literature search, very rare. Except for the already cited sources, there is an additional serious source [2]. The basic result concerning finite-dimensional dissipative systems is that the trajectories of these systems are confined inside some sphere with finite diameter when the time of the evolution is long enough. The knowledge of a more precise geometrical shape of the attractor can be useful when one needs to know, where precisely in the phase space the parameterized dynamical system just occurs. That knowledge together with the knowledge of the actual dynamics of the dynamical system allows us to predict the next position in the trajectories and, thus the behavior of the system in the future.

The two common approaches how to get the geometrical shape of the global attractor are based on utilization of the invariant integrals or on an analysis of that system through constrained maximization using Lagrangian multipliers with inequalities. Both approaches are encumbered with heavy computations. Our approach offers an effective use of the already done results. We take advantage of the knowledge of the properties of some dynamical systems already known to transfer them to the homotopically equivalent system with properties that are yet unknown.

To our best knowledge our approach has not been described till now. The main results are subject of the Theorem 1 and of the Theorem 2. So, the rest of the paper is divided into the part concerning the integral invariants, into the part concerning the shapes and finally in the Conclusions where future possible continuation of the research is mentioned.

2 Integrals of Motion for the Chen System

In this section we give time-dependent integrals of motion. Using the results by [12], we are able to obtain similar formulas to the known integrals of motion for the Lorenz system, see e.g. [6] and the references therein.

Theorem 1. *The following are some integrals of motion for the Chen-like system with $dI/dt = 0$.*

1. $I(x,y,z,t) = (x^2 - 2\sigma z)e^{2\sigma t}$ *with* $b = 2\sigma$ *and* σ, r, s *arbitrary.*
2. $I(x,y,z,t) = (y^2 + z^2)e^{2bt}$ *with* $s = -b, r = 0$ *and* σ, b *arbitrary.*
3. $I(x,y,z,t) = (-rx^2 + y^2 + z^2)e^{2t}$ *with* $\sigma = 1, s = -1, b = 1$ *and r arbitrary.*
4. $I(x,y,z,t) = (4\sigma x^2 + \sigma y^2 + 2\sigma xy - \frac{1}{4\sigma}x^4 + x^2 z)e^{4\sigma t}$ *with* $b = 0, r = -4\sigma, s = -3\sigma$ *and* σ *arbitrary.*
5. $I(x,y,z,t) = (-rx^2 - y^2 + 2xy + \frac{1}{4}x^4 - x^2 z + 4(r-1)z)e^{4t}$ *with* $\sigma = 1, b = 4, s = -1$ *and r arbitrary.*

Proof. Rougly speaking, first we realize that there is a continuous transition between the Lorenz and Chen systems, see [12]. This fact brings the idea that their integrals of motion might also have similar form.

We give here a sketch of the technique for finding integral 1. We guess its form according to the known integral of motion for the Lorenz system $I(x,y,z,t) = (x^2 - 2\sigma z)e^{2\sigma t}$ with $b = 2\sigma$ and σ, r arbitrary. Therefore, we can presume that the coresponding integral of motion for the Chen system has the form $I(x,y,z,t) = (x^2 - Az)e^{Bt}$ with A, B dependent on σ, b, r and s. Hence, $\dot{I} = dI/dt = (2x\dot{x} - A\dot{z} + Bx^2 - ABz)e^{Bt} = (2x\sigma(y-x) - A(xy - bz) + Bx^2 - ABz)e^{Bt} = (2xy\sigma - 2x^2\sigma - Axy + Abz + Bx^2 - ABz)e^{Bt} = 0$. This is satisfied if $A = 2\sigma$ and $B = 2\sigma$ and at the same time $AB = Ab$. Hence, also $b = 2\sigma$. This gives the integral 1. with $b = 2\sigma$ and σ, r, s arbitrary.

The search for other integrals of motion is more technical, with possibly more freedom of picking the values of the parameters, but the idea remains the same. We omit the calculations and leave it to the readers.

It is also clear from the process that the proposed integrals of motion satisfy $\dot{I} = dI/dt = 0$. □

3 Shape of the Chen System

The Lorenz system (1) is dissipative as its divergence is $-\sigma - 1 - b < 0$, because the numbers σ, b are non-negative real numbers. Different situation occurs while analyzing the Chen system (2). The direct computation of its divergence gives $-\sigma + s - b$. In order to get this divergence negative, the following inequality $s < \sigma + b$ has to be satisfied. So, the dissipativity of the Chen system is *conditional* in that sense.

To analyze the shape of the Chen system, we always suppose that the following condition is satisfied:

Condition. Let σ, b, s are real numbers. We suppose that $s < \sigma + b$.

Let

$$\mathscr{M} = \{(x(t),y(t),z(t)) |\ solves$$
$$the\ Chen\ system\ with\ some\ initial\ condition\ (x(0),y(0),z(0))\}.$$

Under the **Condition** the trajectories of the Chen system are confined inside some sphere when the time $t \to \infty$. More preciously, there is some real number $R > 0$ and some real number $T > 0$ such that

$$\mathscr{M} \subseteq \mathscr{S} = \{(x(t),y(t),z(t)) |\ \| x(t)^2 + y(t)^2 + z(t)^2 \| \leq R\}\ \ \forall t \geq\ T.$$

That result is valid for every finite-dimensional dissipative system. Our task is to describe the attracting set of the Chen system in more details. The goal is to show that the global attractor of the Chen system is contained in a volume bounded by a sphere, a cylinder, the volume between two parabolic sheets and a cone. The task was inspired by [5] and substantially utilizes the results herein. That approach is possible since a linear homotopy between Lorenz and Chen systems was established in [12].

For the case of the sphere, the result follows from the dissipativity of the Chen system under the **Condition**. Moreover, we do not pretend on exact estimates of the geometrical sizes of the shapes but only a rough form of the shapes is sufficient for our purpose. Now, we are ready to formulate the result.

Theorem 2. *We suppose that the **Condition** is granted. Then the global attractor of the Chen system is confined inside the following sets.*

1. *An absorbing cylinder*
$$\mathscr{C} = y^2 + (r-z)^2\ \leq r^2$$
when $b \leq s$, and
$$\mathscr{C} = y^2 + (r-z)^2\ \leq \frac{b^2r^2}{4(b+s)}$$
when $b > s$.
2. *An absorbing volume between two parabolic sheets*
$$\frac{x^2}{2\sigma} + \left(1 - \frac{b}{2\sigma}\right) Z_{min} \leq z \leq \frac{x^2}{2\sigma} + \left(1 - \frac{b}{2\sigma}\right) Z_{max}$$
where Z_{min}, Z_{max} are some bounds on z those follows from that fact that the global attractor is bounded.
3. *A repelling cone. The inside of the pair of cones*
$$\sigma(y^2 + z^2) = rx^2$$
is repeling when $\sigma \geq b \geq -s$.

Remark 1. As stated above, we will substantially utilize the results of [5] done for the case of Lorenz system. One can see that the structure of the Lorenz system is very similar to the one of the Chen system. It is clear from the general formulas that the only difference between both systems shows in the second equation of the

Lorenz system (1) and in the second equation of the Chen system (2), where the additional parameter s occurs. Due to results in [12], a linear homotopy between the two systems exists. Therefore, the hypothesis is that the results obtained for the case of the Lorenz system will be helpful when we analyze the Chen system.

Now we prove the theorem.

Proof. The case of an absorbing cylinder. Accordingly to [5], we define a set

$$\mathscr{C} = y^2 + (r-z)^2$$

representing a cylinder in the plane (y,z) with the axis along the $x-$axis. When we differentiate that cylinder along the Chen system, we get

$$\frac{1}{2}\dot{\mathscr{C}} = sy^2 - b(r-z)^2 + br(r-z).$$

The largest value of the $\mathscr{C}$ when $t \to \infty$ is no larger that its largest value inside of the cylinder given by the condition that the right-hand side is positive. That leads to the constrained maximization problem

$$\max\{\mathscr{C}|sy^2 - b(r-z)^2 + br(r-z) \geq 0\}.$$

Using method of the Lagrange multipliers with the Lagrangian

$$L(y,z,\lambda) = y^2 + (r-z)^2 + \lambda(-sy^2 + bz^2 - brz),$$

one gets the set of equations

$$2y - 2\lambda sy = 0$$
$$-2(r-z) + 2\lambda bz - \lambda br = 0$$
$$sy^2 - bz^2 + brz = 0.$$

The case of $\lambda = 0$ immediately gives $y = 0,\ z = r$, and afterward one gets $\mathscr{C} = 0$, which is evidently the minimum. So, starting with

$$sy^2 - bz^2 + brz = 0,$$

and after some arithmetics, one gets the result

$$\mathscr{C} \leq r^2$$

when $b \leq s$ and

$$\mathscr{C} \leq \frac{b^2r^2}{4(b+s)}$$

when $b > s$, which had to be proved.

The case of parabolic sheets. It was stated that, because of dissipativity due to **Condition**, for large times, z is confined within a cylinder with bounds

$$Z_{min} \leq z \leq Z_{max}.$$

Accordingly to [5], we choose

$$\mathscr{P} = x^2 - 2\sigma z.$$

As the conic is fully independent on the variable y, we can enjoy the result of [5] and refer the reader to that source.

The case of repelling cones. Again, accordingly to [5], we choose

$$\mathscr{R} = \sigma(y^2 + z^2) - rx^2.$$

On $\mathscr{R} = 0$, there is a double cone whose axis of symmetry is the $x-$axis which tends to zero at the origin. Similarly as in the first case, we evaluate the derivative along the Chen system in order to get

$$\frac{1}{2}\dot{\mathscr{R}} = r\sigma x^2 + s\sigma y^2 - b\sigma z^2.$$

Now, we eliminate the variable z^2 using the definition of the $\mathscr{R}$. We obtain

$$\frac{1}{2}\dot{\mathscr{R}} = -b\mathscr{R} + r(\sigma - b)x^2 + \sigma(s+b)y^2.$$

If $\sigma \geq b \geq -s$, one gets the following inequality

$$\frac{1}{2}\dot{\mathscr{R}} \geq -b\mathscr{R}.$$

If we start from inside the cone where $\mathscr{R}(0) < 0$, then $\dot{\mathscr{R}} > 0$ and $\mathscr{R}$ increases. As a result, the trajectories of the Chen system run away from the interior of the cone $\mathscr{R} = 0$ to the outside where $\mathscr{R} > 0$. As a result, one gets $\mathscr{R} = \sigma(y^2 + z^2) - rx^2 \geq 0$, that means $\sigma(y^2 + z^2) \geq rx^2$. The theorem is proved. □

Remark 2. In the case of the Chen chaotic attractor, one has $\sigma = 35.0,\ r = -7,\ s = 28,\ b = 3$. The **Condition** is evidently fulfilled, so the system is dissipative. Because of $b < s$ the radius of the absorbing cylinder is $r^2 = 49$ and the repelling cone is described by $35(y^2 + z^2) + 7x^2 = 0$. The parabolic sheets are described by $\frac{x^2}{70} + \frac{67}{70}Z_{min} \leq z \leq \frac{x^2}{70} + \frac{67}{70}Z_{max}$.

4 Conclusions

In this paper we have studied the Chen system for a general set of parameters. The main goal of the paper is to show how the homotopy between two dynamical systems can help when some otherwise hardly obtainable properties of a dynamical system (the Chen system in our case) can be derived from an other system for which the properties are known (the Lorenz system in our case). Especially, the integral invariants play a significant role in physics, in particular for the search of conservation laws since the derivation of them belongs very often to trial-and-error approach. The same situation comes up if we want to analyze the phase space of an attractor by asking what a geometrical shape the attractor is. The knowledge of the geometrical shape of the attractor in the phase space as a function of the system parameters can predict how the system can behave in the future. This paper has shown much more simple method of deriving some properties of the unknown system using homotopy with a well-known system. It would be a very useful task to study much more intensively the homotopies between dynamical systems, i.e. between already known systems and still unknown systems to get the additional properties of practically usable systems, in the future.

Acknowledgements. The paper is supported by the Czech Science Foundation through the research grant no. 13-20433S.

References

1. Anastassiou, S., Bountis, T., Petalas, Y.G.: On the topology of the Lü attractor and related systems. J. Phys. A: Math. Theor. 41, 1–13 (2008)
2. Barboza, R., Chen, G.: On the Global Boundedness of the Chen System. Int. J. Bifurcation and Chaos 21(11), 3373–3385 (2011)
3. Cao, J., Chen, C., Zhang, X.: The Chen system having an invariant algebraic surface. Int. J. Bifurcation and Chaos 18(12), 3753–3758 (2008)
4. Chen, G., Ueta, T.: Yet another chaotic attractor. Int. J. Bifurcation and Chaos 9, 1465–1466 (1999)
5. Doering, C.R., Gibbon, J.D.: On the shape and dimension of the Lorenz attractor. Dynamics and Stability of Systems 10(3), 255–268 (1995)
6. Giacomini, H., Neukirch, S.: Integrals of motion and the shape of the attractor for the Lorenz model. Physics Letters A 227, 309–318 (1997)
7. Li, D., Cui, L., Yin, Z.: New Results on Connecting the Lorenz and Chen Systems. In: Proceedings of the 7th World Congress on Intelligent Control and Automation, Chongqing, China, June 25 - 27, pp. 840–844 (2008)
8. Liu, Y., Yang, Q.: Dynamics of the Lü system on the invariant algebraic surface and at infinity. Int. J. Bifurcation and Chaos 21(9), 2559–2582 (2011)
9. Llibre, J., Messias, M., Da Silva, P.R.: Global dynamics in the Poicaré ball of the Chen system having invariant algebraic surfaces, Int. J. Bifurcation Chaos 22(6),1250154-1–17 (2012)
10. Llibre, J., Valls, C.: Polynomial first integrals for the Chen and Lü systems. Int. J. Bifurcation and Chaos 22(11), 1250262-1–7 (2012)
11. Lorenz, E.N.: Deterministic nonperiodic flow. J. Atmosph. Sci. 20, 130–141 (1963)
12. Lü, J., Chen, G., Cheng, D., Čelikovský, S.: Bridge the gap between the Lorenz system and the Chen system. Int. J. Bifurcation and Chaos 12(12), 2917–2926 (2002)
13. MATLAB version 8.1.0.604. Natick, Massachusetts: The MathWorks Inc. (2013)
14. Ueta, T., Chen, G.: Bifurcation analysis of Chen's attractor. Int. J. Bifurcation and Chaos 10, 1917–1931 (2000)
15. Wu, K., Zhang, X.: Global dynamics of the generalized Lorenz systems having invariant algebraic surfaces. Physica D: Nonlinear Phenomena 244(1), 25–35 (2013)
16. Yu, P., Yao, W., Chen, G.: Analysis on topological properties of the Lorenz and the Chen attractors using GCM. Int. J. Bifurcation Chaos 17(8), 2791–2797 (2007)

Message Embedded Synchronization for the Generalized Lorenz System and Its Use for Chaotic Masking

Sergej Čelikovský and Volodymyr Lynnyk

Abstract. This paper implements and analyzes the well-known message embedded synchronization scheme for the case of the generalized Lorenz system. Such a synchronization may be used for chaotic masking scheme using a single channel only. This method was already discussed in the earlier literature for the particular classes of systems. In this paper, a more general class where message embedded synchronization is possible is described. Then, it is shown that the generalized Lorenz system falls within that class. Furthermore, using the resulting synchronization, the novel secure encryption scheme is proposed. It requires very reasonable amount of data to encrypt and time to decrypt one bit. Basically, to encrypt one bit, only one iteration (i.e. only one real number of 6 valid digits) is needed. At the same time, 100 percent of the carrying chaotic signal can be used. The method is also demonstrated by numerical simulations of a digital data encryption and decryption.

1 Introduction

A large number of communication schemes that are based on chaos synchronization have been proposed during the last two decades [3, 11, 14, 18, 20, 21, 22]. As matter of fact, the well-known features of the chaotic systems like strong dependence on the initial conditions, topological transitivity, wide spread spectrum of its signal, etc., directly suggest the idea to use suitable chaos generators to build a new generation of secure encryption methods. Nevertheless, methods using continuous time chaos usually consider analogue communication while discrete time chaotic systems are used for digital data encryption. Unfortunately, the use of the continuous time chaotic systems for the encryption of the digital data and both its practical aspects

Sergej Čelikovský · Volodymyr Lynnyk
Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Prague, Czech Republic
e-mail: {celikovs,voldemar}@utia.cas.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 313–322.
DOI: 10.1007/978-3-319-00542-3_32

and security analysis have been studied much less [1, 12, 13] due to the prevalently used analogue chaotic masking [2, 16]. Important role for using the continuous time chaotic systems for the secure encryption is played by the synchronization [15, 19]. The purpose of this paper is to provide further detailed analysis of the chaotic masking scheme based on the generalized Lorenz system. Moreover, it will be shown that based on the so-called message embedded synchronization one can even use the chaotic masking in the continuous time systems for the encryption of the digital information.

Digital information secure encryption using continuous time chaotic systems was already studied in [8, 10, 17], where the so-called desynchronization chaos shift keying (DECSK) method was introduced and studied. This method uses the transmitter which consists of two different chaotic systems and the receiver which consists of two perfectly synchronized chaotic systems being their copies. If even very short signal is being sent, one of those two copies becomes quickly **desynchronized**, which enables to detect the transmitted bit. To make that method realistic it was achieved using the second derivative error detection that even 2 iterations are sufficient to detect desynchronization. All these ideas were implemented in detail for the so-called generalized Lorenz system (GLS), [5, 6, 7].

In this paper, the very same GLS will be used to show its message embedded synchronization ability. Then we will show that it is possible to use this property to implemented chaotic masking for a digital signal modulation (binary step-like function). This binary step-like function can have very small amplitude, up to 10^{-5}, which makes the signal undetectable, unless one has at his/her disposal the correct carrying chaotic signal to subtract it from the encrypted message.

This paper presents both the general result for some class of systems that can be message embedded synchronized, as well as secure encryption implementation using the fact that GLS belong to that general class of systems. Numerical implementation will be demonstrated as well.

The rest of the paper is organized as follows. In Section 2, we briefly repeat some known facts about the generalized Lorenz system. Section 3 introduced the message embedded synchronization scheme for the generalized Lorenz system which is illustrated by numerical experiments described in Section 4. Final section gives some conclusion and outlooks for future research.

2 The Generalized Lorenz System and Its Synchronization

First, let us recall some previously published results on the generalized Lorenz system classification and synchronization. Further details may be found in [5, 6, 7].

Definition 2.1. *The following general nonlinear system of ordinary differential equations in $\mathbb{R}^3$ is called a generalized Lorenz system (GLS):*

$$\dot{x} = \begin{bmatrix} A & 0 \\ 0 & \lambda_3 \end{bmatrix} x + \begin{bmatrix} 0 \\ -x_1x_3 \\ x_1x_2 \end{bmatrix}, \; A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \tag{1}$$

where $x = [x_1 \ \ x_2 \ \ x_3]^\top$, $\lambda_3 \in \mathbb{R}$, *and* A *has eigenvalues* $\lambda_1, \lambda_2 \in \mathbb{R}$, *such that*

$$-\lambda_2 > \lambda_1 > -\lambda_3 > 0. \tag{2}$$

The inequality (2) goes back to the well-known Shilnikov's chaos analysis near the homoclinicity and can be viewed as the necessary condition for the chaos existence, see more detailed discussion in [5]. GLS is said to be *nontrivial* if it has at least one solution that goes neither to zero nor to infinity nor to a limit cycle. The following result, enabling the efficient synthesis of a rich variety of chaotic behaviors for GLS, has been obtained in [5]:

Theorem 2.2. *For the nontrivial generalized Lorenz system* (1) – (2)*, there exists a nonsingular linear change of coordinates,* $z = Tx$*, which takes* (1) *into the following generalized Lorenz canonical form:*

$$\dot{z} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} z + cz \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & -1 \\ 1 & \tau & 0 \end{bmatrix} z, \tag{3}$$

where $z = [z_1, z_2, z_3]^\top$*,* $c = [1, -1, 0]$ *and parameter* $\tau \in (-1, \infty)$*.*

Actually, the parameter τ plays important role of single scalar bifurcation parameter, while remaining parameters has only qualitative influence being eigenvalues of the approximate linearization of GLS at the origin. These qualitative parameters are just required to satisfy robust condition (2), so that fine tuning may be done using the single scalar parameter τ only. In [4] GLS is further extended to the so-called hyperbolic-type generalized Lorenz systems (HGLS) which has the same canonical form as (4) but with $\tau \in (\infty, -1)$. In such a way, the parameter range to be used in the encryption later on is further extended. In [6] complete and nice classification of all related systems is given showing that many recently introduced in the literature classes are actually particular cases of the GLS or the HGLS.

Synchronization of GLS is based on yet another canonical form, the so-called **observer canonical form of GLS** provided by the following

Theorem 2.3. *Both nontrivial GLS (1) and its canonical form (3) are state equivalent to the following form:*

$$\frac{\mathrm{d}\eta}{\mathrm{d}t} = \begin{bmatrix} (\lambda_1 + \lambda_2)\eta_1 + \eta_2 \\ -\eta_1[\lambda_1\lambda_2 + (\lambda_1 - \lambda_2)\eta_3 + \frac{(\tau+1)\eta_1^2}{2}] \\ \lambda_3\eta_3 + K_1(\tau)\eta_1^2 \end{bmatrix} \tag{4}$$

$$K_1(\tau) = \frac{\lambda_3(\tau+1) - 2\tau\lambda_1 - 2\lambda_2}{2(\lambda_1 - \lambda_2)}, \tag{5}$$

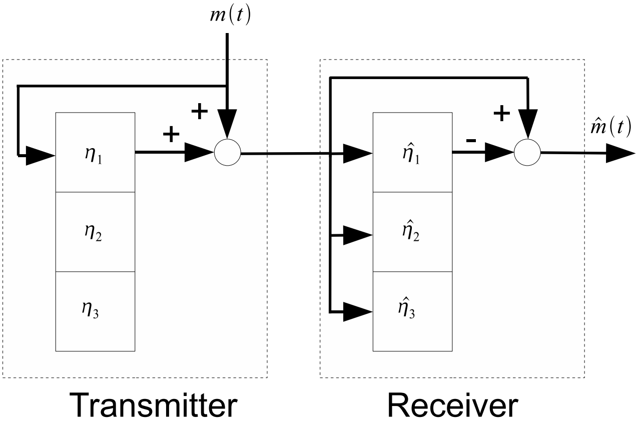


Fig. 1 Message embedded synchronization secure encryption using GLS

where $\eta = [\eta_1, \eta_2, \eta_3]^\top$*, which is referred to as the* observer canonical form. *The corresponding smooth coordinate change and its inverse are*

$$\eta = \left[z_1 - z_2,\ \ \lambda_1 z_2 - \lambda_2 z_1,\ \ z_3 - \frac{(\tau+1)(z_1-z_2)^2}{2(\lambda_1-\lambda_2)} \right]^\top \tag{6}$$

$$z = \left[\frac{\lambda_1\eta_1+\eta_2}{\lambda_1-\lambda_2},\ \ \frac{\lambda_2\eta_1+\eta_2}{\lambda_1-\lambda_2},\ \ \eta_3 + \frac{(\tau+1)\eta_1^2}{2(\lambda_1-\lambda_2)} \right]^\top. \tag{7}$$

Indeed, the above observer canonical form, when viewing $\eta_1 = x_1 = z_1 - z_2$ as the output, is almost in the form linearizable by output injection. This leads to the following observer-based synchronization of two copies of GLS.

Theorem 2.4. *Consider system (4-5) with the output* η_1 *and its uniformly bounded trajectory* $\eta(t),\ t \geq t_0$*. Further, consider the following system having input* η_1^m *and state* $\hat\eta = (\hat\eta_1, \hat\eta_2, \hat\eta_3)^\top$*:*

$$\begin{aligned}\frac{\mathrm{d}\hat\eta}{\mathrm{d}t} = & \begin{bmatrix} l_1 & 1 & 0 \\ l_2 & 0 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \hat\eta + \begin{bmatrix} \lambda_1+\lambda_2-l_1 \\ -\lambda_1\lambda_2-l_2 \\ 0 \end{bmatrix} \eta_1^m + \\ & + \begin{bmatrix} 0 \\ -(\lambda_1-\lambda_2)\eta_1^m\hat\eta_3 - (1/2)(\tau+1)(\eta_1^m)^3 \\ K_1(\tau)(\eta_1^m)^2 \end{bmatrix},\end{aligned} \tag{8}$$

where $l_{1,2} < 0$*. For all* $\varepsilon \geq 0$*, assume* $|\eta_1(t) - \eta_1^m(t)| \leq \varepsilon$*. Then, it holds exponentially in time that*

$$\overline{\lim}_{t\to\infty}\|\eta(t) - \hat\eta(t)\| \leq C\varepsilon,$$

for a constant $C > 0$. In particular, for $\eta_1^m \equiv \eta_1$, system (8) *is a global exponential observer for system* (4)-(5).

Proofs of the Theorems 2.3 -2.4 may be found in [7]. In the sequel, the system (4)-(5) will be often called as the master while (8) as the slave.

Theorem 2.4 can be used for the simple chaotic masking in the sense of [11]. Here, message to be masked is added to the synchronizing signal which corrupts the synchronization, as claimed by Theorem 2.4, where synchronization error does not go to zero completely. Namely, $\eta_1^m = \eta_1 + m(t)$, where $m(t)$ is message to be masked, i.e.

$$\overline{\lim}_{t\to\infty}\|\eta(t) - \hat{\eta}(t)\| \leq C\|m(t)\|.$$

This drawback will be removed in the next section using the so-called message embedded synchronization.

3 Message Embedded Synchronization for the Generalized Lorenz System and Its Use for Chaotic Masking

In this section we propose the so-called message embedded synchronization scheme. Such a synchronization may be used for chaotic masking scheme using single channel only. This method was discussed by Lian K.-Y. et. al. in [16] for a particular class of systems. Let us characterize more general class where message embedded synchronization is possible.

Consider a nonlinear system of the form

$$\begin{bmatrix} \dot{x}^1 \\ \dot{x}^2 \end{bmatrix} = \begin{bmatrix} F_1 & 0 \\ 0 & F_2 \end{bmatrix} \begin{bmatrix} x^1 \\ x^2 \end{bmatrix} + \begin{bmatrix} \varphi^1(Hx^1, x^2) \\ \varphi^2(Hx^1) \end{bmatrix}, \tag{9}$$

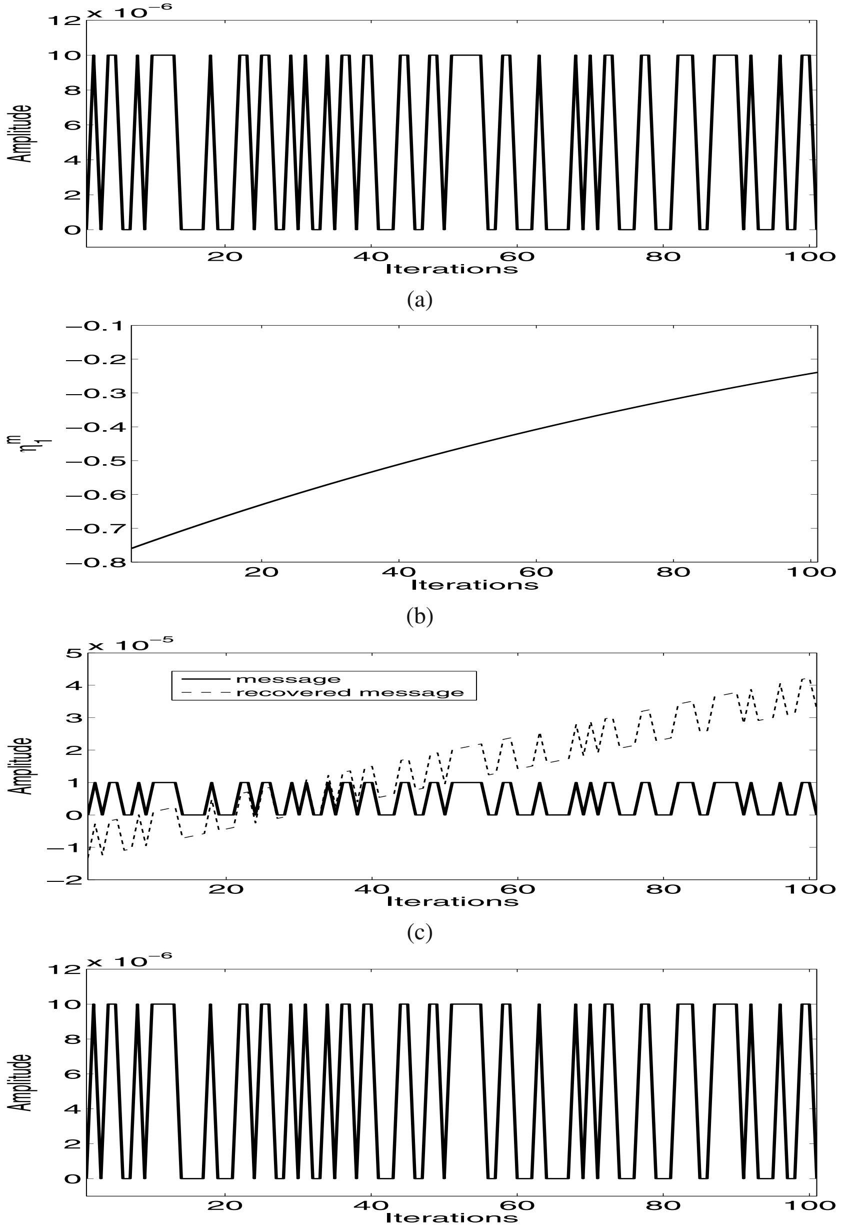
where $\begin{bmatrix} x^1 \\ x^2 \end{bmatrix} = x \in R^n$, $x^1 \in R^{n_1}$, $x^2 \in R^{n_2}$, $n_1 + n_2 = n$, F is $(n \times n)$ matrix, H is $(n_1 \times 1)$ matrix, F_1 is $(n_1 \times n_1)$ matrix, F_2 is $(n_2 \times n_2)$ matrix. Suppose (F_1, H) is detectable pair and F_2 is Hurwitz. Further, let nonlinear functions φ^1, φ^2 be such that

$$\varphi^1 : R^{n_2+1} \to R^{n_1}, \varphi^2 : R \to R^{n_2}.$$

Then, the synchronized copy of (9) can be obtained using the scalar synchronizing signal $Hx(t)$ as follows

$$\begin{bmatrix} \dot{y}^1 \\ \dot{y}^2 \end{bmatrix} = \begin{bmatrix} F_1 & 0 \\ 0 & F_2 \end{bmatrix} \begin{bmatrix} y^1 \\ y^2 \end{bmatrix} + \\ + \begin{bmatrix} \varphi^1(Hx^1, y^2) \\ \varphi^2(Hx^1) \end{bmatrix} + \begin{bmatrix} L_1 H(y^1 - x^1) \\ 0 \end{bmatrix}. \tag{10}$$

Here L_1 is $(1 \times n_1)$ matrix such that $F_1 + L_1 H$ is Hurwitz. Namely, define $e = (e^1, e^2)^\top = (y^1 - x^1, y^2 - x^2)$. Then, subtracting (9) from (10) gives



(a)

(b)

(c)

(d)

Fig. 2 Time histories related with the encryption and decryption of the plaintext “0101100101111100000100011011001010010110110001100110111110011000100000101011 0001100011100111110100010011” using the chaotic masking. Namely, (a) plaintext time signal $m(t)$; (b) ciphertext $\eta_1(t)$; (c) reconstructed plaintext $\hat{m}(t)$ with error; (d) reconstructed plaintext without error. “Intruder” can see **only** the second graph from the top which does not give any clue for the possible decryption.

$$\begin{bmatrix} \dot{e}^1 \\ \dot{e}^2 \end{bmatrix} = \begin{bmatrix} F_1 + L_1 H & 0 \\ 0 & F_2 \end{bmatrix} \begin{bmatrix} e^1 \\ e^2 \end{bmatrix} + \\ + \begin{bmatrix} \varphi^1(Hx^1, y^2) - \varphi^1(Hx^1, x^2) \\ 0 \end{bmatrix}. \tag{11}$$

Notice, that $e_2 \to 0$ exponentially since F_2 is Hurwitz. Assuming that the synchronization signal $Hx(t)$ of (9) is bounded guarantees that

$$\varphi^1(H(x(t)), y^2(t)) - \varphi^1(H(x(t)), x^2(t)) \to 0$$

exponentially as $t \to \infty$ as well. Therefore, $e_1 \to 0$ exponentially as $t \to \infty$, since $F_1 + L_1 H$ is Hurwitz. That is $e \to 0$ exponentially as $t \to \infty$ and therefore (9) and (10) are synchronized.

Chaotic Masking via Precise Message Embedded Synchronization. Consider system

$$\begin{bmatrix} \dot{x}^1 \\ \dot{x}^2 \end{bmatrix} = \begin{bmatrix} F_1 & 0 \\ 0 & F_2 \end{bmatrix} \begin{bmatrix} x^1 \\ x^2 \end{bmatrix} + \\ + \begin{bmatrix} \varphi^1(Hx^1 + \tilde{m}(t), x^2) \\ \varphi^2(Hx^1 + \tilde{m}(t)) \end{bmatrix} + \begin{bmatrix} L_1 \tilde{m}(t) \\ 0 \end{bmatrix} \tag{12}$$

and its copy to be synchronized

$$\begin{bmatrix} \dot{y}^1 \\ \dot{y}^2 \end{bmatrix} = \begin{bmatrix} F_1 & 0 \\ 0 & F_2 \end{bmatrix} \begin{bmatrix} y^1 \\ y^2 \end{bmatrix} + \begin{bmatrix} \varphi^1(Hx^1 + \tilde{m}(t), x^2) \\ \varphi^2(Hx^1 + \tilde{m}(t)) \end{bmatrix} + \\ + \begin{bmatrix} L_1 H \\ 0 \end{bmatrix} y_1 - \begin{bmatrix} L_1(Hx^1 + \tilde{m}(t)) \\ 0 \end{bmatrix}. \tag{13}$$

Then $|y - x| \to 0$ as $t \to \infty$ exponentially. Namely, define $e = (e^1, e^2)^\top = (y^1 - x^1, y^2 - x^2)$. Then subtracting (12) from (13) gives

$$\begin{bmatrix} \dot{e}^1 \\ \dot{e}^2 \end{bmatrix} = \begin{bmatrix} F_1 + L_1 H & 0 \\ 0 & F_2 \end{bmatrix} \begin{bmatrix} e^1 \\ e^2 \end{bmatrix} + \\ + \begin{bmatrix} \varphi^1(Hx^1 + \tilde{m}(t), y^2) - \varphi^2(Hx^1 + \tilde{m}(t), x^2) \\ 0 \end{bmatrix}. \tag{14}$$

Now, assuming synchronization signal $Hx + \tilde{m}(t)$ is bounded, one has again that $e \to 0$ exponentially as $t \to \infty$. The message embedded scheme with precise synchronization can be implemented as follows: Let $m(t)$ be the message to be sent. Let $\tilde{m}(t) = m(t) + \mathscr{M}(x(t))$ be the embedded message. Here, $\mathscr{M}(x(t))$ is arbitrary bounded function of the state $x(t)$, which should be independent of scalar synchronizing signal Hx^1 as much as possible. Then using (12) one generates transmitted signal as

$$s(t) = \tilde{m}(t) + Hx^1(t) = m(t) + Hx^1 + \mathscr{M}(x(t)).$$

Recovered message $\hat{m}(t)$ would be $\hat{m}(t) = s(t) - Hy^1(t) - \mathscr{M}(y(t))$. Therefore $\hat{m}(t) - m(t) = H(x^1(t) - y^1(t)) + \mathscr{M}(x(t)) - \mathscr{M}(y(t))$, i.e. $\hat{m}(t) - m(t) \to 0$ as $t \to \infty$ exponentially.

This explains the term “precise” chaotic masking scheme synchronization: in contrast to synchronization and chaotic masking described in [11], where message corrupts synchronization due to Theorem 2.4, the method just presented completely filters out the influence of the modulated message on the synchronization.

Remark 3.1. *Notice that, observer canonical form of GLS* (4) *is the system exactly in the form* (9)*, where* $F_1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $F_2 = \lambda_3$, $H = [1,0]$, $x^1 = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix}$, $x^2 = [\eta_3]$, $\varphi^1 = \begin{bmatrix} (\lambda_1 + \lambda_2)\eta_1 \\ -\lambda_1\lambda_2\eta_1 - (\lambda_1 - \lambda_2)\eta_1\eta_3 - \frac{(\tau+1)\eta_1^3}{2} \end{bmatrix}$, $\varphi^2 = K(\tau)\eta_1^2$. *Therefore, GLS in its canonical form can be used for chaotic masking using precise message embedded synchronization.*

4 Numerical Experiments

The chaotic masking scheme diagram is presented in the Fig. 1. Here $m(t)$ is an original sent message and $\hat{m}(t)$ is a recovered message. Transmitter generates an output signal which depends on the input message.

Fig. 2 illustrates an application of chaotic masking scheme based on the generalized Lorenz system for encryption and decryption of information. Fig. 2 (a) shows the original message which influences the dynamics of the transmitter and is added to the transmitted signal to be sent via the communication channel. Since the amplitude of the original message is much smaller than the signal generated by the transmitter, the intruder cannot notice a change in the transmitted signal. Fig. 2 (b) illustrates the signal which is transmitted via the communication channel. Next, Fig. 2 (c) shows the encrypted message versus the original message decrypted in Fig. 2 (d). The signal generated by the transmitter exceeds the amplitude of the original message 100000 times thus allowing the use of the message embedded synchronization for the secure encryption. The above described decryption scheme of this method requires initial synchronization of chaotic system on the transmitter side and chaotic system on the receiver side up to the best available numerical precision[1], called in the sequel as the “numerical zero”. Therefore, the initial condition is the immediate candidate for the secret key. As our “numerical zero” is 10^{-4}, this key space is naturally discretized in the sense that two initial conditions that are closer to each other than the numerical zero should be represented by the same key. In papers [8, 17], published by the authors earlier, the secure encryption system based on the GLS was described the so-called Desynchronization Chaos Shift Keying (DECSK). It used a similar principle for the formation of the secure key. At the same time,

[1] MATLAB-SIMULINK ode4 Runge-Kutta procedure with the fixed step size equal to 0.001 is being used throughout the paper.

DECSK secure encryption scheme is not allowed to use 100 percent of the carrier signal and for the carrier signal at values close to zero the use of more than one iteration per bit is required. The chaotic masking communication scheme does not have the above mentioned problems and can be used for encryption of digital data using 100 percent of the carrier signal.

5 Conclusions and Outlooks

The generalized Lorenz system family has been analyzed and used for the chaotic masking of the digital information. More precisely, the message embedded synchronization has been used to avoid corrupting the synchronization by the transferred message. This enabled to use a digital modulation with very small amplitude, thereby enhancing the security of the masking. In other words, it was shown that the proposed digital communication method has a potential of introducing a high degree of security at a low receiver complexity. At the same time, it requires reasonable amount of data to encrypt a single bit, thereby making of practical use of continuous time chaotic system for digital data encryption. Further research will be devoted to making the message expansion even smaller and secure of the communication method will be analyzed in detail. Another future goal is to apply those methods to encrypt information transmitted between nodes in complex networks [9].

Acknowledgements. This work is supported by the Czech Science Foundation through the research grant No. 13-20433S.

References

1. Alvarez, G., Li, S.: Some basic cryptographic requirements for chaos-based cryptosystems. International Journal of Bifurcation and Chaos 16, 2129–2151 (2006)
2. Alvarez-Ramirez, J., Puebla, H., Cervantes, I.: Stability of observer-based chaotic communications for a class of Lur'e systems. International Journal of Bifurcation and Chaos 12(7), 1605–1618 (2002)
3. Carroll, T.L., Pecora, L.M.: Using multiple attractor chaotic systems for communication. Chaos 9(2), 445–451 (1999)
4. Čelikovský, S.: Observer Form of the Hyperbolic-Type Generalized Lorenz System and its Use for Chaos Synchronization. Kybernetika 40(6), 649–664 (2004)
5. Čelikovský, S., Chen, G.: On a generalized Lorenz canonical form of chaotic systems. International Journal of Bifurcation and Chaos 12, 1789–1812 (2002)
6. Čelikovský, S., Chen, G.: On the generalized Lorenz canonical form. Chaos Solitons and Fractals 26(5), 1271–1276 (2005)
7. Čelikovský, S., Chen, G.: Secure synchronization of a class of chaotic systems from a nonlinear observer approach. IEEE Transactions on Automatic Control 50(1), 76–82 (2005)
8. Čelikovský, S., Lynnyk, V.: Desynchronization chaos shift keying method based on the error second derivative and its security analysis. International Journal of Bifurcation and Chaos 22(9), 1250231-1–1250231-11 (2012)

9. Čelikovský, S., Lynnyk, V., Chen, G.: Robust synchronization of a class of chaotic networks. Journal of The Franklin Institute (accepted, 2013)
10. Čelikovský, S., Lynnyk, V., Šebek, M.: Observer-based chaos sychronization in the generalized chaotic Lorenz systems and its application to secure encryption. In: Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, USA, pp. 3783–3788 (2006)
11. Cuomo, K.M., Oppenheim, A.V.: Circuit Implementation of Synchronized Chaos with Application to Communications. Physical Review Letters 71(1), 65–68 (1993)
12. Dachselt, F., Schwarz, W.: Chaos and cryptography. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 48(12), 1498–1509 (2001)
13. Kocarev, L.: Chaos-based Cryptography: A Brief Overview. IEEE Circuits and Systems Magazine 1(3), 6–21 (2001)
14. Kocarev, L., Parlitz, U.: General approach for chaotic synchronization with applications to communication. Physical Review Letters 74, 5028–5031 (1995)
15. Kolumbán, G., Kennedy, M.P., Chua, L.O.: The Role of Synchronization in Digital Communications Using Chaos-part II: Chaotic Modulation and Chaotic Synchronization. IEEE Transactions on Circuits and Systems-I:Fundamental Theory and Applications 45(11), 1129–1140 (1998)
16. Lian, K.-Y., Liu, P.: Synchronization with message embedded for generalized Lorenz chaotic circuits and its error analysis. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 47(9), 1418–1424 (2000)
17. Lynnyk, V., Čelikovský, S.: On the anti-synchronization detection for the generalized Lorenz system and its application to secure encryption. Kybernetika 46(1), 1–18 (2010)
18. Parlitz, U., Kocarev, L., Stojanovski, T., Preckel, H.: Encoding messages using chaotic synchronization. Physical Review E 53, 4351–4361 (1996)
19. Pecora, L.M., Carroll, T.L.: Synchronization in chaotic systems. Physical Review Letters 64(8), 821–824 (1990)
20. Volkovskii, A.R., Rulkov, N.F.: Synchronouns chaotic response of a nonlinear ocsillating system as a principle for the detection of the information component of chaos. Technical Physics Letters 19, 97–99 (1993)
21. Yang, J., Zhu, F.: Synchronization for chaotic systems and chaos-based secure communications via both reduced-order and step-by-step sliding mode observers. Communications in Nonlinear Science and Numerical Simulation 18(4), 926–937 (2013)
22. Zhang, G., He, L.-F., Zhang, T.-Q.: A secure communication system based on DCSK. In: Zhao, M., Sha, J. (eds.) ICCIP 2012, Part I. CCIS, vol. 288, pp. 135–143. Springer, Heidelberg (2012)

Using Complex Network Topologies and Self-Organizing Maps for Time Series Prediction

Juan C. Burguillo and Bernabé Dorronsoro

Abstract. A Self-organizing Map (SOM) is a competitive learning neural network architecture that make available a certain amount of classificatory neurons, which self-organize spatially based on input patterns. In this paper we explore the use of complex network topologies, like small-world, scale-free or random networks; for connecting the neurons within a SOM, and apply them for Time Series Prediction (TSP). We follow the classical VQTAM model for function prediction, and consider several benchmarks to evaluate the quality of the predictions. The results presented in this work suggest that the most regular the network topology is, the better results it provides in prediction. Besides, we have found that not updating all the cells at the same time provides much better results.

1 Introduction

Time Series Prediction (TSP) is a function approximation to estimate (predict) future values after a time sequence of ordered observations. The aim of TSP methods is to find the model that best fits with the empirical observations, and once a particular model has been selected for TSP, the next step concerns with estimating its parameters from the available data. Finally, a model evaluation is done concerning its predictive ability usually over a new set of testing data.

In the last two decades, a growing interest in TSP methods has been observed, particularly within the field of neural networks [13, 11] with the application of some well-known supervised neural architectures, such as the Multilayer Perceptron (MLP), the Radial Basis Functions (RBF) networks and, more recently, the

Juan C. Burguillo
EE Telecomunicación, University of Vigo, 36310-Vigo, Spain
e-mail: J.C.Burguillo@uvigo.es

Bernabé Dorronsoro
LIFL, University of Lille, France
e-mail: bernabe.dorronsoro_diaz@inria.fr

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 323–332.
DOI: 10.1007/978-3-319-00542-3_33

Self-organizing Maps (SOM) [5]. The application of these neural networks architectures in TSP problems can be explained because of prediction can be considered as a supervised learning problem.

Among those possible architectures, the Self-organizing Map (SOM) [9] is a well-known competitive learning neural network. SOM learns from examples a mapping (projection) from a high-dimensional continuous input space onto a low-dimensional discrete space (lattice). After training, the neuron weights in the map can provide a model of the training patterns mainly through: vector quantization, regression and clustering. Using these techniques, self-organizing maps have been applied in the last decades to multiple applications [7] in areas like automatic speech recognition, monitoring of plants and processes, cloud classification, micro-array data analysis, document organization, image retrieval, etc. In the last decade, there has been also an increasing interest in using SOM for TSP problems [5].

In this paper we consider the use of complex network topologies [2] like spatial, small-world, scale-free or random networks for connecting the neurons within a SOM and explore its performance for time series prediction using the VQTAM model [4]. Yang et al. [14] implement a SOM with small-world topology, in which neighborhood sizes are progressively reduced during the learning process. The resulting SOM seems to have a faster convergence and have a more reasonable weight distribution. Jiang et al. [8] use SOM with dynamic complex networks topologies, obtained by means of evolutionary optimization algorithms, and apply them for classification purposes. To our knowledge, ours is the first paper exploring the use of SOM together with complex networks for prediction purposes.

The main contribution of the paper is twofold: first to explore the application of this self-organizing neural network with different complex topologies for time series prediction, and second to evaluate the topologies and the conditions that provide a better quality in prediction.

The rest of the paper is organized as follows. Sect. 2 introduces Self-organizing Maps together with the VQTAM model that we use for time series prediction. Sect. 3 describes the different types of complex networks that we use in the paper. Sect. 4 describes the results obtaining after using the network topologies to connect the neurons in the SOM, and apply them to different TSP problems. Finally, Sect. 5 presents our conclusions and describe some future work.

2 Self-Organizing Maps (SOM)

Self-organizing Maps (SOM) [9], also denoted as Self-organizing Feature Maps (SOFM) or Kohonen Neural Networks, make available a certain amount of classificatory resources, usually denoted as neurons, cells or units, which self-organize based on the input patterns. From a topological point of view, a SOM is a single layer neural network, where the neurons are set along a d-dimensional grid. In most applications this grid is 2-dimensional and rectangular, but hexagonal grids or other dimensional spaces have been also used in the literature. Associated with each neuron is a weight vector of the same dimension as the input vectors (patterns) and a position in the map space (see Fig.1). The self-organizing map describes a mapping from a higher dimensional input space to a lower dimensional map space. The procedure for placing a vector from the input space onto the map is to find the neuron with the closest (smallest distance metric) weight vector. After several iterations, the result is an ordered network in which nearby neurons will share certain similarities. Therefore similar input patterns activate similar areas in the SOM producing a local specialization in the global self-organized network.

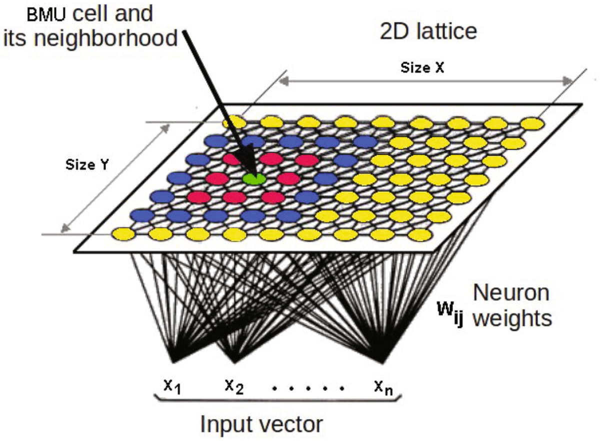


Fig. 1 A SOM map with an input vector, a winner cell and its neighborhood

Before training, the neurons may be initialized randomly, and like most artificial neural networks, SOMs operate in two modes: training and mapping. *Training* builds the map using input examples (it is a competitive process, also called vector quantization), while *Mapping* automatically classifies a new input vector.

Fig. 2 presents a 2-dimensional to 1-dimensional mapping to visualize the training process [9], and represents the evolution of the neurons in the input space. The 2-dimensional input data points are uniformly distributed in a triangle, and a 1-dimensional SOM is trained with these patterns. As training proceeds, the line first unfolds (steps 1 to 100), and then fine-tunes itself to cover the input space.

2.1 Formal Definition

The SOM learns from the input patterns a mapping from a high-dimensional continuous input space X onto a low-dimensional discrete space *L* (the lattice) of *m* neurons which are arranged in fixed topological forms, e.g., as a rectangular 2-dimensional array. The number of neurons *m* in the map is usually defined after experimentation with the dataset.

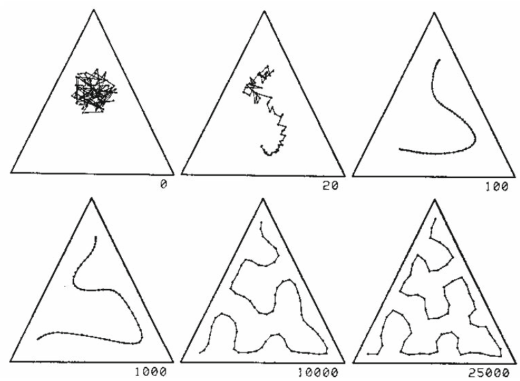


Fig. 2 The picture presents a 2D to 1D mapping by a SOM [9]

The function map $b(x) : X \rightarrow L$ is defined by assigning to the current input vector $x(t) \in X \subset \mathbb{R}^n$ a neuron index b in the map obtained by:

$$b(x(t)) = argmin||x(t) - w_i(t)||, \quad \forall i \in \{1,\ldots,m\} \ , \tag{1}$$

where $||\cdot||$ denotes the Euclidean distance, n is the input vector dimension and t is the discrete time step associated with the iterations of the algorithm. The weight vectors $w_i(t)$ in the map are trained according to a competitive-cooperative learning rule, where the weight vector of a winning neuron with index b (usually denoted as Best Matching Unit, BMU) and its neighbors in the output map array are updated with this formula:

$$w_i(t+1) = w_i(t) + \alpha(t)h(b,i,t)[x(t) - w_i(t)], \quad i = 1,\ldots,m \ , \tag{2}$$

where $0 < \alpha(t) < 1$ is the learning rate and $h(b,i,t)$ is a weighting function which limits the neighborhood of the BMU. This neighborhood function assumes values in $[0,1]$ and is high for neurons that are close to the BMU, and small (or 0) for neurons far away. The neighborhood radio and $\alpha(t)$ should both decay with t to guarantee convergence of the weight vectors in the map to stable steady states.

2.2 *The VQTAM Model*

In order to do TSP with SOM, we use the Vector-Quantized Temporal Associative Memory (VQTAM) model [4]. In a general formulation, the VQTAM models the input vector at time step t, $x(t)$ by two parts: the first part, denoted $x^{in}(t) \in \mathbb{R}^p$, carries data about the input of the dynamic mapping to be learned. The second part, denoted $x^{out}(t) \in \mathbb{R}^q$, contains data concerning the desired output of this mapping. To do such estimation, the weight vector of neuron i, $w_i(t)$, has its dimension adapted as follows:

$$x(t) = \begin{pmatrix} x^{in}(t) \\ x^{out}(t) \end{pmatrix} \quad and \quad w_i(t) = \begin{pmatrix} w_i^{in}(t) \\ w_i^{out}(t) \end{pmatrix} \ , \tag{3}$$

where $w_i^{in}(t) \in \mathbb{R}^p$ and $w_i^{out}(t) \in \mathbb{R}^q$ are, respectively, the portions of the weight (prototype) vector which store information about the inputs and the outputs of the desired mapping. In general we have that $p > q$, and particularly for the univariate TSP that we consider in this paper, we have that $p > 1$ and $q = 1$, so that the following definitions apply:

$$x^{in}(t) = [y(t) \quad y(t-1) \quad \ldots \quad y(t-p+1)]^T \tag{4}$$

$$x^{out}(t) = s(t) \ , \tag{5}$$

where $s(t)$ is the value generated by the process at the time step t, $y(t)$ is a vector containing the p past samples, and the T denotes the transpose vector. During learning, the winning neuron at time step t is determined based only on $x^{in}(t)$:

$$b(x^{in}(t)) = argmin||x^{in}(t) - w_i^{in}(t)||, \quad \forall i \in \{1,\ldots,m\} \ . \tag{6}$$

For updating the weights, both $x^{in}(t)$ and $x^{out}(t)$ are used:

$$w_i^{in}(t+1) = w_i^{in}(t) + \alpha(t)h(b,i,t)[x^{in}(t) - w_i^{in}(t)], \quad i = 1,\ldots,m \tag{7}$$

$$w_i^{out}(t+1) = w_i^{out}(t) + \alpha(t)h(b,i,t)[x^{out}(t) - w_i^{out}(t)], \quad i = 1,\ldots,m \ , \tag{8}$$

where these two learning rules performs topology-preserving vector quantization on the input and output spaces of the mapping being learned. As training proceeds, the SOM learns to associate the input prototype vectors w_i^{in} with the corresponding output prototype vectors w_i^{out}. Once the SOM has been trained, its output $z(t)$ for a new input vector is estimated from the learned codebook vectors as follows:

$$z(t) = w_b^{out}(t) \ , \tag{9}$$

where $w_b^{out}(t)$ is the weight output vector of the corresponding winning neuron. From it we define the error function used in this paper as:

$$e(t) = s(t) - z(t) = s(t) - w_b^{out}(t) \ . \tag{10}$$

3 Complex Networks for SOM

In this work we consider, besides spatial SOM topologies, other types of complex network topologies in order to explore their performance in time series prediction scenarios. In the last years, complex network topologies, e.g. small-world or scale-free ones, have been applying in multiple research works [2]. According to the famous algorithm of Watts and Strogatz [12], small-world networks are intermediate topologies between regular and random ones. Their properties are most often quantified by two key parameters: the clustering coefficient and the mean-shortest path.

On the one hand, the clustering coefficient quantifies how the neighbors of a given network node (i.e., the nodes to which it is connected) are on average interconnected. It reflects the network capacities of local information transmission. The graph distance between two nodes of the network is the smallest number of links one has to travel to go from one node to the other. On the other hand, the mean-shortest path is the average graph distance of the network, and indicates the capacities of long distance information transmission.

Random Networks (RN) are randomly generated by placing a fixed number of edges between vertices at random with uniform probability [6]. Random networks are characterized by their short characteristic path length, meaning that the maximum distance between any to nodes in the graph is short.

Small-world (SW) networks [1] share properties of both random nets and regular lattices, because they have a short characteristic path length and a high clustering coefficient, i.e., there is a high connectivity degree among the vertices in the graph. This SW topology can be generated by the Watts and Strogatz algorithm [12]. In our case we change the links from a node to its neighbors with probability $\beta = 0.1$

Finally, Scale-free (SF) networks follow a power law concerning degree distribution, at least asymptotically, i.e, the fraction $P(k)$ of nodes in the network having k connections to other nodes goes for large values of k as $P(k) \sim k^{-\lambda}$ usually having $2 < \lambda < 3$. Scale-free networks show characteristics present in many real world networks like the presence of hubs connecting almost disconnected sub-networks. The preferential attachment method [3] can be used to build such topologies, reflecting also the dynamical aspect of those networks.

4 Experimental Results

In this section we present the results obtained after applying the VQTAM model in TSP scenarios considering several complex network topologies. All the simulations have been repeated 30 times and, except it is explicitly said, the results correspond to the average value obtained all along the 30 runs. The simulations have been performed in a Java simulator, and every execution took less than a minute in a Pentium dual-core 3GHz with 3 GBytes of RAM. The particular parameters used in all the next simulations are: p=2 (input values), q=1 (output value), initial $\alpha = 0.5$ that decreases linearly reaching zero at the end of the training phase. Every training set is repeated 20 times (epochs) before proceeding to the testing phase. The values selected for these parameters have been applied after an exhaustive testing in multiple experiments, providing us the best results. In order to simplify the analysis as much as possible, and considering the particularities introduced by the complex topologies, our neighborhood function $h(b,i,t)$ considers only the closer cell's neighborhood, which means the four cells conforming the Von Neumman neighborhood (North, East, South, West) in the spatial case, and the direct neighbors in the case of the complex topologies used here (SW, SF and RN). The value defined for the BMU is $h(b,i,t) = 1$ and for its neighbor cells is $h(b,i,t) = 0.5$.

We have tested several functions and benchmarks for TSP, but for a lack of space, in this section we only present three representative cases that we consider interesting. We have selected a trigonometric function (SCSTS) obtained by $f(x) = 2sin(x) - cos(3x) + sin(5 * x)$ because it is continuous and its smooth variation induced higher error rates than classical discrete TSP benchmarks. We also have consider the benchmark (BRTS) presented in [5] and the Mackey-Glass series (MGTS), which are based on the Mackey-Glass differential equation [10]. This is a benchmark widely regarded for comparing the generalization ability of different methods, and corresponds to a chaotic time series generated from a time-delay ordinary differential equation.

4.1 Number of Neurons in the Lattice

We first start considering how many cells there must be in the lattice for a particular problem, depending on the topology. There is a heuristic value, used in the SOM literature and in the SOM toolbox released by [9], which is: $m = K_u.\sqrt{N_{In}}$ where usually $K_u = 5$ and N_{In} is the number of input samples available.

Fig. 3 describes the influence of the K_u parameter over the error rate depending on the type of topology and using the function SCSTS. We can see in the figure how the best results are obtained using a spatial network, while the worst ones are provided by the scale-free net. Besides, using values of K_u above 5 does not improve the error rate significantly. The results are similar, with minor variations for the other benchmarks. The variance in the error value obtained along the different executions has the order $(1e-4)$, so in general there is no much difference along the multiple executions. Therefore, for the next experiments presented here, we have selected $m = 5.\sqrt{N_{In}}$ neurons, as recommended by the literature, for all the topologies.

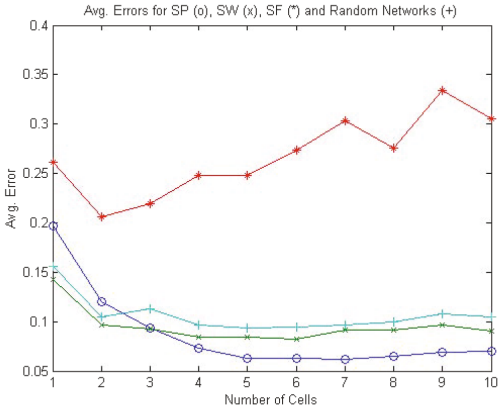


Fig. 3 The figure shows the influence of the parameter K_u in the error rate, depending on the type of topology: spatial (SP), small-world(SW), scale-free(SF) and random net (RN)

4.2 Probability of Neighbor Updating

Now we consider what happens if we do not select all the cells in the BMU's neighborhood always for update, as it is done in the literature. We define P_u as the probability of updating a particular neighbor cell. We will consider here the MGTS benchmark introduced before. Fig. 4 presents a snapshot of this benchmark after $10,000$ iterations over the input samples. Fig. 5 describes the normalized results obtained after applying our prediction model over the benchmark. In the horizontal axis we represent the probability of updating the weights of a BMU's neighbor cell, according to the VQTAM model introduced before. We only have considered four probabilities $[0.25, 0.5, 0.75, 1]$ corresponding respectively to select in average one neighbor, two, three or the four cells in a Von Neumann neighborhood. We did the same in the other complex network topologies, where a cell may have an arbitrary number of neighbors. As we can see, the best results correspond again to the spatial topology, but in all topologies it seems that we get better results if we do not update all the cells in the neighborhood at the same time.

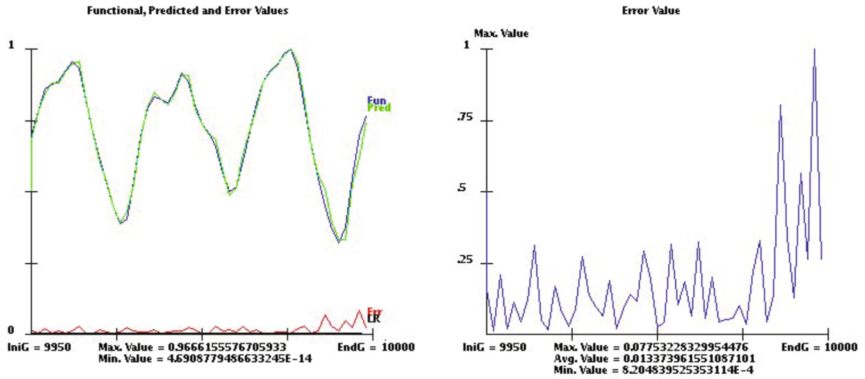


Fig. 4 Snapshot of the Java simulator showing in the left picture the MGTS values (blue), the prediction (green) and the error (red) at the bottom. The right picture shows the error values magnified.

Finally, Table 1 presents the results for the three benchmarks considered here, but only considering the spatial topology that provides the best results in our experiments. The table mainly describes the influence of P_u in the average error obtained in every case along the 30 executions. As we can see, P_u values between $[0.25, 0.5]$ seem to be the best option for cell updating and the variance obtained is low in all the runs. The average accuracy improvement of the best case (in bold font in the table) and the standard SOM (i.e., when all neighbors are updated) is 12.56%, 4.13%, and 10.54% for SCSTS, MGTS, and BRTS problems, respectively.

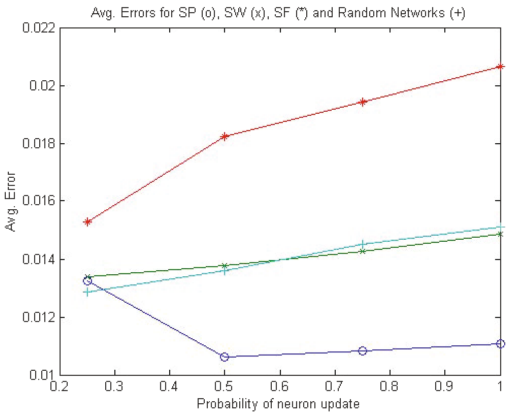


Fig. 5 Average error depending on the probability of updating a neighbor cell (Mackey-Glass)

Table 1 Results obtained for several problems using the spatial topology, and selecting different probabilities for P_u. The best result for every problem appears in bold font.

Problem	P_u	Avg. Error	Var. Error
SCSTS	0.25	0.061360065698206914	3.981810690609938E-7
SCSTS	0.5	**0.056077601916607836**	6.857511003258309E-5
SCSTS	0.75	0.05943884979541651	5.95047587581242E-5
SCSTS	1.0	0.06413329835252152	4.3691880189338955E-5
MGTS	0.25	0.013234076724205592	1.6721832073302142E-4
MGTS	0.5	**0.010617894573664272**	6.68503022490138E-8
MGTS	0.75	0.010828351521748793	1.544678187249031E-7
MGTS	1.0	0.011074839936379505	4.981605739558393E-7
BRTS	0.25	**0.03468846296514643**	1.3275132605892995E-6
BRTS	0.5	0.03665173603623375	4.6085559730992295E-9
BRTS	0.75	0.0381303608909958	9.800985921138E-7
BRTS	1.0	0.03877596086299864	1.5471953178892961E-6

5 Conclusions

In this paper we consider the Time Series Prediction problem (TSP), and the use of Self-organizing Maps (SOM) neural networks for modeling it. We have selected the VQTAM model from the recent literature as a particular case of SOM applications in several TSP scenarios. We have considered the use of complex network topologies like small-world, scale-free and random networks for connecting the neurons within a SOM, and explore its performance considering several benchmarks for TSP. To our knowledge, this is the first paper studying prediction problems with different types of complex network topologies for interconnecting the neurons at a SOM.

The results presented in this work suggest that the classical spatial SOM network topology provides better results than the other types studied. Among them, small-world and random networks provide similar results considering that the resulting nets have a relatively regular structure and low clustering coefficient. We have found specially inadequate the use of scale-free topologies for TSP. Nevertheless, the most interesting contribution presented in this paper is that it seems better to do not update all the cells at the same time in the BMU's neighborhood. In this sense, we got up to 12.56% improvement in the average accuracy of the SOM. This means that even with the case of a minimal neighborhood used in a classical SOM (i.e., the Von Neumann neighborhood), it is much better to do not update the four cells always, but only subsets of them.

Future work will consider to evaluate a higher spectra network topologies and prediction benchmarks, trying to infer the conditions to select the right cells that must be updated in the BMU neighborhood.

Acknowledgements. B. Dorronsoro acknowledges the support offered by the National Research Fund, Luxembourg, AFR contract no 4017742.

References

1. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. Reviews of Modern Physics 74, 47–97 (2002)
2. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.U.: Complex networks: Structure and dynamics. Physics Reports 424, 175–308 (2006)
3. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science 286(5439), 509–512 (1999)
4. Barreto, G.A., Araújo, A.F.R.: Identification and control of dynamical systems using the self-organizing map. IEEE Transactions on Neural Networks 15(5), 1244–1259 (2004)
5. Barreto, G.A.: Time Series Prediction with the Self-Organizing Map: A Review. Perspectives of Neural-Symbolic Integration, Studies in Computational Intelligence 77, 135–158 (2007)
6. Erdos, P., Renyi, A.: On random graphs. Publicationes Mathematicae 6, 290–297 (1959)
7. Van Hulle, M.M.: Self-organizing maps: Theory, design, and application, Tokyo (2001)
8. Jiang, F., Berry, H., Schoenauer, M.: The Impact of Network Topology on Self-Organizing Maps. In: GEC 2009, Shanghai, China (2009)
9. Kohonen, T.: Self-Organizing Maps, 3rd edn. Springer (2001)
10. Mackey, M.C., Glass, J.: Oscillation and chaos in physiological control systems. Science 197, 287 (1977)
11. Palit, A.K., Popovic, D.: Computational Intelligence in Time Series Forecasting: Theory and Engineering Applications, 1st edn. Springer (2005)
12. Watts, D.J., Strogatz, S.H.: Collective dynamics of small-world networks. Nature 393, 440–442 (1998)
13. Weigend, A., Gershefeld, N.: Time Series Prediction: Forecasting the Future and Understanding the Past. Addison-Wesley (1993)
14. Yang, S., Luo, S.-W., Li, J.: An extended model on self-organizing map. In: King, I., Wang, J., Chan, L.-W., Wang, D. (eds.) ICONIP 2006. LNCS, vol. 4232, pp. 987–994. Springer, Heidelberg (2006)

Initial Errors Growth in Chaotic Low-Dimensional Weather Prediction Model

Hynek Bednar, Ales Raidl, and Jiri Miksovsky

Abstract. The growth of small errors in weather prediction is exponential. As an error becomes larger, the growth rate decreases and then stops with the magnitude of the error about at a value equal to the size of the average distance between two states chosen randomly.

This paper studies an error growth in a low-dimensional atmospheric model after the initial exponential divergence died away. We test cubic, quartic and logarithmic hypotheses by ensemble prediction method. Furthermore quadratic hypothesis that was suggested by Lorenz in 1969 is compared with the ensemble prediction method. The study shows that a small error growth is best modeled by the quadratic hypothesis. After the initial error exceeds about a half of the error saturation value, logarithmic approximation becomes superior.

Keywords: Chaos, Atmosphere, Prediction, Error growth.

1 Introduction

Due to the fact that the atmosphere is chaotic dynamical system, the growth of small errors in weather prediction is exponential. In the case of sufficiently small initial error, the governing equations can be linearized, which leads to exponential growth of the error. Generally, whether an error is small enough to guarantee the exponential growth depends on specific meteorological conditions and/or model under study. This issue, in context of the model studied in this paper, was addressed in [1-2]. For a more comprehensive introduction to the problem of weather predictability we refer reader to the book of Palmer and Hagedorn [3].

Hynek Bednar · Ales Raidl · Jiri Miksovsky
Department of Meteorology and Environment Protection, Faculty of Mathematics and Physics,
Charles University in Prague
e-mail: hynek.bednar@seznam.cz
{Ales.Raidl,Jiri.Miksovsky}@mff.cuni.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 333 – 342.
DOI: 10.1007/978-3-319-00542-3_34

If the system which governs the change of the error is linear, then the exponential growth will continue unabated. The Earth's atmosphere is a non-linear system and as an error becomes larger, the growth rate decreases. Eventually, all systematic growth should stop and the size of the error should be equal to the average size of the distance of two randomly chosen states. In 1969 Lorenz [4] introduced a quadratic hypothesis which 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 will also be quadratic. In the present study we examine this hypothesis by low-dimensional atmospheric model introduced by Lorenz in 1996 [5].

2 Model

Lorenz [5] introduced a model of nonlinear behavior, with N variables $X_1,\ldots, X_N$ connected by governing equations

$$dX_n/dt = -X_{n-2}X_{n-1} + X_{n+1}X_{n-1} - X_n + F. \tag{1}$$

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 equation (1) simulate advection. Linear terms represent mechanical and thermal dissipation. The model quantitatively, to a certain extent, describes weather systems, but equations (1) cannot be derived from any atmospheric dynamic equations. The goal was to formulate the simplest possible set of dissipative chaotically behaving differential equations that share some properties with the "real" atmosphere. In [1-2] the reasoning for usability of such model is discussed in more detail.

For our computation we choose $N = 36$, so each sector covers 10 degrees of longitude. Parameters F were selected equal to *8*, *9* and *10* successively. We first choose arbitrary values of the variables, and, using a fourth order Runge-Kutta method with a time step Δt = *0.05* or *6 hours*, we integrate forward for *14400 steps*, or *10 years*. We then use final values, which should be more or less free of transient effect. For individual parameters F, we estimated the global largest Lyapunov exponents, by the method of numerical calculation presented in [6].

$$F = 8 \rightarrow \lambda_{\max,8} = 0.33, \quad F = 9 \rightarrow \lambda_{\max,9} = 0.39, \quad F = 10 \rightarrow \lambda_{\max,10} = 0.46.$$

By the definition [6]: „A bounded dynamical system with a positive Lyapunov exponent is chaotic". Because all values of the largest Lyapunov exponents of the model are positive, the system is bounded and therefore is chaotic for all three chosen values of F. The choice of parameters F and *time unit = 5 days* is made to obtain the same values of the largest Lyapunov exponents as state of the art models of complete global circulation.

3 Ensemble Prediction Method

The ensemble prediction method is similar to [5] 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 50 days (*K=200 steps*), obtaining *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.* To get more representative values, we make a total of *M* = *250* runs in the same manner, in each run letting new values of X_{n0} be the old values of X_{nK}. Finally, we let $e^2(\tau) = 1/N\left(e^2_{1k} + \ldots + e^2_{Nk}\right)$ be the average of the *N* values, where $\tau = k\Delta t$ is the predictable range and $\log E^2(\tau) = 1/M\left(\log e^2(\tau)_1 + \ldots + \log e^2(\tau)_M\right)$ is the average of M values. Logarithmic average of *M* values is chosen, because it is more suitable for comparison with growth governed by the largest Lyapunov exponent. For further information see [7-8].

4 Quadratic Hypothesis

According to Lorenz [5] there is the eventual cessation of exponential growth due to processes represented by nonlinear terms in 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 will also be quadratic. To describe this, Lorenz [5] defined equation

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

where $E(t)$ is a distance at time *t* between two originally nearby trajectories and *a*, *b* are constants. The quadratic hypothesis is also used to describe behavior of initial error growth for example in [9-10].

4.1 Method

Because we want to study behavior of the equation (2), we make differences $y_k = \left(E(\tau+\Delta t) - E(\tau)\right)/\Delta t$ at points $x_k = \left(E(\tau) + E(\tau+\Delta t)\right)/2$, where E is average initial error growth calculated from the ensemble prediction method (section 3).

Next we interpolate the data (x_k, y_k). Interpolation equations are:

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

$$y(t) = dE(t)/dt = aE(t) - bE(t)^3, \tag{3}$$

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

$$y(t) = dE(t)/dt = -aE(t)\ln\big(bE(t)\big). \tag{5}$$

Eq. (2) is the examined quadratic hypothesis. Equations (3) and (4) are added, because Lorenz [4] noticed that the cubic equation would also fit his data; (4) represents Quartic hypothesis. Equation (5) is chosen because, if we let $Q(t) = \ln\left(\overline{E(t)}\right)$, $\bar{E}$ is normalized *E,* and then $dQ(t)/dt = a\left(1 - e^{Q(t)}\right)$ is the quadratic hypothesis. In [11] it is assumed, that linear fit $dQ(t)/dt = -aQ(t)$ is better than quadratic hypothesis. In terms of *E* it is the logarithmic law.

Parameters *a* and *b* in equations (2-5) are examined and discussed in the next chapter.

4.2 Results

Different initial errors e_0 exhibit different behavior of an error growth. To study that, we selected for each *F* six magnitudes of $\| e_0 \|$: $\| e_{0,1} \|$ = 0.0001, $\| e_{0,2} \|$ = 0.001, $\| e_{0,3} \|$ = 0.01, $\| e_{0,4} \|$ = 0.1, $\| e_{0,5} \|$ = 0.6, $\| e_{0,6} \|$ = 1, where $\| \ \|$ marks the Euclidean norm. The interpolation equations are tested for all three values of parameter *F* and initial error e_0 . Table 1 shows the rms error between values obtained from the ensemble prediction and from the interpolation equations. The error is divided by the average value of $\left(E(\tau + \Delta t) - E(\tau)\right)/\Delta t$. Fig. 1 displays the error growth rate dE/dt versus *E* for all parameters *F* and for $e_{0,2}$ and $e_{0,5}$. Each interpolation equation gives specific values of *a* and *b* for particular *F* and e_0. Our aim is to find a general description of *a* and *b* by well-known parameters of the system. The early growth rate should be close to $dE/dt = \lambda_{\max} E$. That means $a = \lambda_{\max}$ for all interpolation equations. Results for equations (2-4) are following. The constant *a* measures the growth rate of small errors, the quadratic (cubic, quartic) term has to be negative if *a* is positive, since it is the only factor that can stop the growth. If *E* is normalized, so that the value which it approaches as *t*→∞ is unity, *b* = *a*. For not normalized *E,* $b = \lambda_{\max}/E^*$ for the equation (2), $b = \lambda_{\max}/E^{*2}$ for the equation (3) and $b = \lambda_{\max}/E^{*3}$ for the equation (4), where E^* denotes the saturation value for *E*. For the equation (5) then $b = 1/E^*$.

From Table 1 and Fig. 1, it is obvious, that the most precise and therefore usable hypotheses are the quadratic and logarithmic (equations (2) and (5)) ones. Theoretical values of parameters *a* and *b* would make the impreciseness of cubic and quartic hypotheses even greater. Therefore we will from now work only with quadratic and logarithmic hypotheses. Table 2 shows the rms error between values obtained from the ensemble prediction and from equations (2) and (5), where parameters *a* and *b* are the expected theoretical values.

From Table 2 we can see higher increase of the percent error for the Eq. (5), than for the Eq. (2). The difference between Table 2 and Table 1 is displayed in Table 3.

Table 1 Rms error between values obtained from the ensemble prediction and from interpolation equations. The error is divided by the average value of $\left(E(\tau+\Delta t)-E(\tau)\right)/\Delta t$ and displayed in percent. (Gray cells mark values with the best results)

inter-	initial	percent error																	
polat.	error	0.0001			0.001			0.01			0.1			0.6			1		
equat.	F	8	9	10	8	9	10	8	9	10	8	9	10	8	9	10	8	9	10
ax-bx^2		16	19	12	13	11	13	16	12	14	16	15	14	34	27	29	47	41	36
ax-bx^3		23	27	22	23	22	20	21	22	25	29	25	27	40	38	38	53	50	51
ax-bx^4		30	29	32	30	29	28	28	33	32	37	32	35	48	44	43	60	52	55
axln(bx)		20	21	19	22	23	21	22	22	22	22	25	21	26	21	23	39	31	34

Table 2 The rms error between values obtained from the ensemble prediction and from equations (2) and (5), where parameters a and b are the expected theoretical values. The error is divided by the average value of $\left(E(\tau+\Delta t)-E(\tau)\right)/\Delta t$ and displayed in percent.

inter-	initial	percent error																	
polat.	error	0.0001			0.001			0.01			0.1			0.6			1		
equat.	F	8	9	10	8	9	10	8	9	10	8	9	10	8	9	10	8	9	10
ax-bx^2		32	25	29	30	20	19	22	20	16	18	13	15	34	29	28	48	45	40
axln(bx)		97	89	94	83	85	82	73	77	84	73	63	72	42	41	46	39	36	34

Table 3 The absolute value of the difference between Table 2 and Table 1

inter-	initial	percent error																	
polati.	error	0.0001			0.001			0.01			0.1			0.6			1		
equat.	F	8	9	10	8	9	10	8	9	10	8	9	10	8	9	10	8	9	10
ax-bx^2		16	6	17	17	9	6	6	8	2	2	2	1	0	2	1	1	4	4
axln(bx)		67	68	75	61	62	61	51	55	62	51	38	51	16	20	23	0	5	0

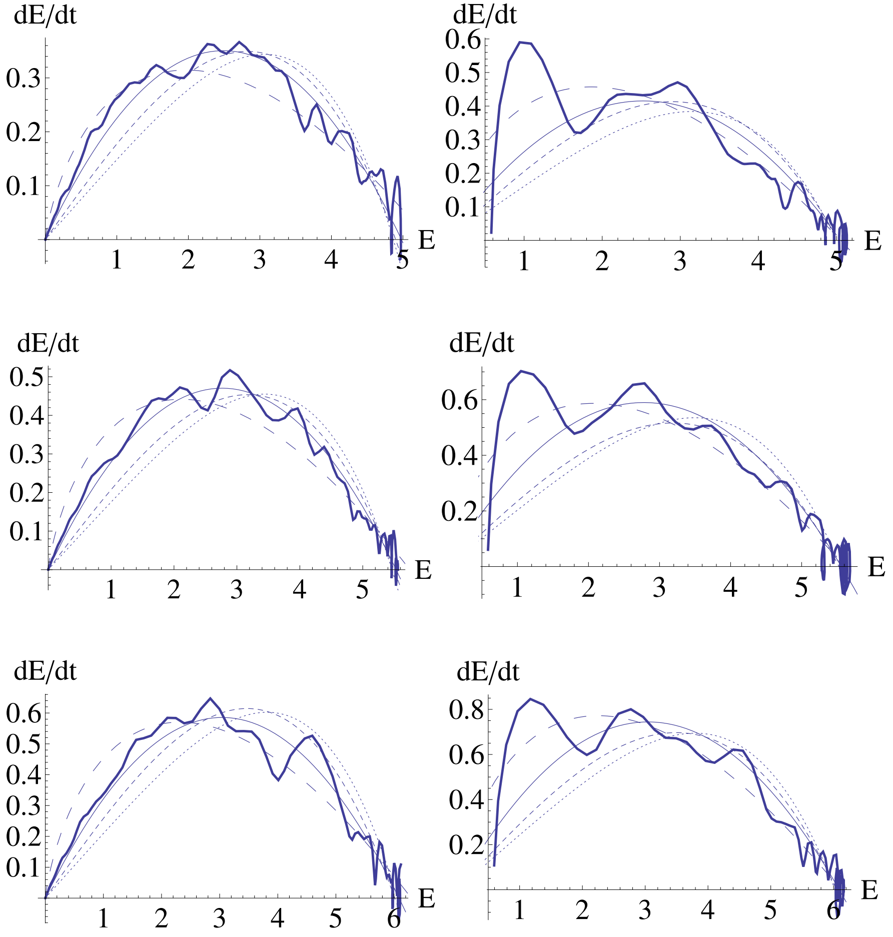


Fig. 1 The error growth rate dE/dt versus E for all parameters F and for $e_{0,2}$ and $e_{0,5}$. The first row is for $F = 8$, the second for $F = 9$ and the third for $F = 10$. The first column is for $e_{0,2}$, and the second for $e_{0,5}$. The thick line represents ensemble prediction, the thin line is equation (2), the dashed line is the equation (3), the thinly dashed line is the equation (4) and the largely dashed line is the equation (5).

5 Exponential Growth

Usability of exponential model of initial error growth is depended on the size of initial error as well as the model parameters F. In this section, we compare ensemble prediction method (section 3) with exponential approximations. Through interpolation we try to find a general rules that can lead that.

5.1 Results

In Table 4, time length t_e ,during which results from ensemble prediction method are close to theoretical exponential growth, is displayed for each initial error e_0 and each parameter *F*. In the same table, predictability t_p (time intervals, where *E* is growing) is also displayed. Fig. 2 then shows time length t_e versus natural logarithm of initial errors from $e_{0,1}$ to $e_{0,5}$ for all parameters *F*. Fig. 2 also displays predictability t_p versus time length t_e. We omit results from $e_{0,6}$, because exponential growth is limited to small initial errors.

Table 4 Time length t_e, where results from ensemble prediction method are close to theoretical exponential growth and predictability t_p (time intervals, where *E* is growing) for each initial error e_0 and each parameter *F*.

error	0.0001			0.001			0.01			0.1			0.6			1		
F	8	9	10	8	9	10	8	9	10	8	9	10	8	9	10	8	9	10
t_e [day]	22	16	16	15	12	12	9	8	7	4	4	3	0	0	0	0	0	0
t_p [day]	50	45	39	42	39	33	36	32	27	30	26	22	25	22	19	17	15	13

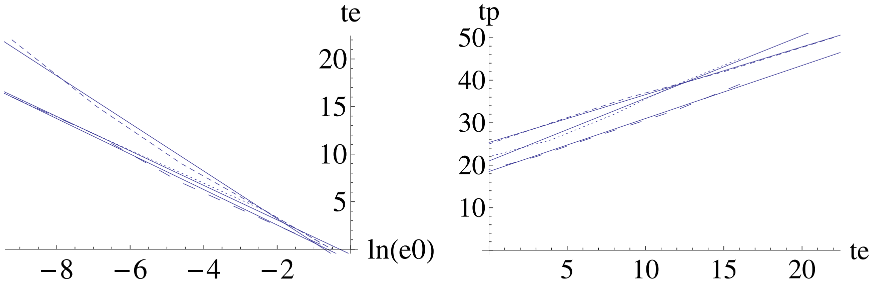


Fig. 2 Time length t_e versus natural logarithm of initial errors from $e_{0,1}$ to $e_{0,5}$ for all parameters *F* (the left side) and predictability t_p versus time length t_e (the right side). The dashed line is for *F = 8*, the thinly dashed line is for *F = 9*, the largely dashed line is for *F = 10* and solid lines are linear interpolations.

From left side of Fig. 2 and Table 4 we can see linear dependence between t_e and $\ln(e_0)$, suggesting that as we choose bigger initial error, the window of exponential growth decreases with natural logarithm of the error. For each parameter *F*, there are different coefficients *a* and *b* of linear interpolation on the logarithmic scale:

$$t_e = a + b\ln(e_0) . \tag{6}$$

Coefficient *a* lies between $(-0.5;-1.7)$ and *b* between $(-1.8;-2.5)$. For each parameter *F*, the logarithmic decrease is unique and the authors didn't discover any theoretical explanation of the value of this coefficients. Right side of Fig. 2 illustrates the linear dependence between t_p and t_e, suggesting that, as predictability t_p grows, the period of usability of theoretical exponential growth t_e grows proportionally. For each parameter *F*, there are different coefficients *c* and *d* of linear interpolation:

$$t_p = c + dt_e . \tag{7}$$

Coefficient *c* lies between $(18;25)$ and *d* between $(1.1;1.5)$, for each parameter *F* this linear increase is unique. Coefficient *c* is close to predictability t_e for $e_{0,5}$ and coefficient *d* is close to *1*.

6 Conclusion

Lorenz's results [4] pretty well fulfilled cubic relation (3), though he had only used limited number of available data for his study. We showed that neither the equation (3) nor the equation (4) fits our data properly, compared to other alternatives. Two usable hypotheses for error growth rate are:

$$dE(t)/dt = \lambda_{\max} \cdot E(t) - \left(\lambda_{\max}/E^*\right) \cdot E(t)^2 , \tag{8}$$

$$dE(t)/dt = -\lambda_{\max} \cdot E(t) \cdot \ln\left(E(t)/E^*\right), \tag{9}$$

where $\lambda_{\max}$ is the largest Lyapunov exponent and E^* is the saturated value for *E*.

If we are looking for the best approximation of model values of the error growth rate $\dot{E}$, quadratic law (2) fits the best for e_0 up to about *0.5*. For higher values it is better to use logarithmic law (5). On other hand, if we want to estimate parameters of the model or use equations (8) and (9) directly, it is, according to Fig. 2, Table 2 and Table 3, better to use quadratic (Eq. 8) for $e_0 = \langle 0;1 \rangle$ and logarithmic law (Eq. 9) for $e_0 = \langle 1;2 \rangle$. Here, we also need to mention that variables $X_1, \ldots, X_N$ lie between *-5* and *+10*. The average value $\overline{X}$ of X_n is *2*, this means that for $e_0 \le \overline{X}/2$ it is better use the (Eq. 8) and for $e_0 \ge \overline{X}/2$ it is better to use logarithmic law (Eq. 9) . This is in good agreement with [11], where they suggested the same.

Solutions of Eq. (8) and (9) are:

$$E(t) = \frac{E^*}{\left(\left(E^*/e_0\right) - 1\right)\exp\left(-\lambda_{\max} t\right) + 1}, \tag{10}$$

$$E(t) = E^* \left(e_0 / E^*\right)^{\exp(-\lambda_{\max} t)}. \tag{11}$$

Fig. 3 shows dependence of predictability t_p (time intervals, where E is growing) on natural logarithm of initial errors from $e_{0,1}$ to $e_{0,5}$ for all parameters F. Linear dependence between t_e and $\ln(e_0)$ was identified: As we choose greater initial error, the predictability t_p decrease with natural logarithm of its magnitude. For each parameter F, there is different coefficient e and similar coefficient f of interpolation:

$$t_e = e + f \ln(e_0). \tag{12}$$

Coefficient e lies between $(17;23)$ and f between $(-2.3;-2.8)$ and for each parameter F, the evolutions are almost parallel. Coefficient e is close to predictability t_e for $e_{0,5}$ and the authors didn't discover any theoretical reason for the value of the coefficient f. Predictability is depended on the size of the initial error and model parameter F and/or Lyapunov exponent. The maximum predictability governed by equation (10) is *50* days for F = *8* and $e_{0,1}$. The lower predictability, governed by equation (11), is for F = *10* and $e_{0,6}$ (Table 4). For contemporary forecast models, the limit of predictability is getting close to two weeks. Fig. 4 displays, time variations of the average prediction error E obtained from Lorenz's model, from equations (10) and (11) and from the exponential growth governed by the largest Lyapunov exponent.

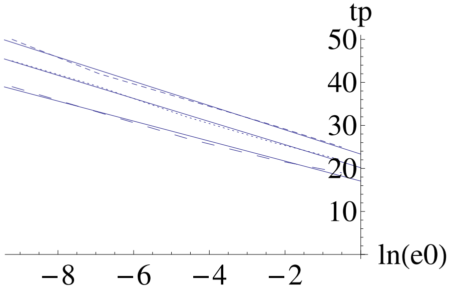


Fig. 3 Predictability t_p versus natural logarithm of initial errors from $e_{0,1}$ to $e_{0,5}$ for all parameters F .The dashed line is for F = *8*, the thinly dashed line is for F = *9*, the largely dashed line is for F = *10* and solid lines are linear interpolations.

To find out more details about equations (6, 7, 12) we would have to use more data and different models. Comparison with the behaviour of the real meteorological forecast models would also be highly desirable, such task, however, would be quite challenging, due to the complexity of such simulations as well as other factor.

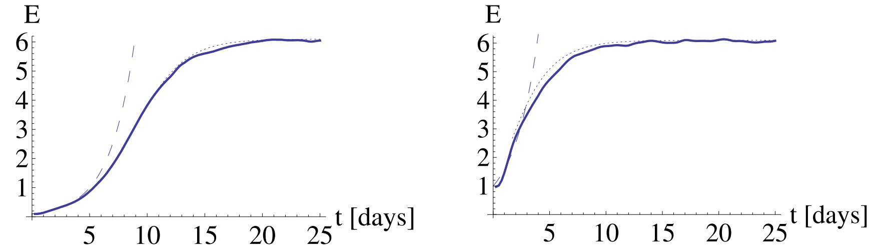


Fig. 4 Time variations of the average prediction error E obtained from the Lorenz's model (the thick line), from equations (8) and (9) (the thinly dashed line) and exponential growth governed by the largest Lyapunov exponent (the largely dashed line). The left side is for the equation (8) , $e_{0,4}$ and $F = 9$. The right side is for the equation (9), $e_{0,6}$ and $F = 9$.

Acknowledgements. The authors were supported by Research Plan no. MSM0021620860 and by project no. SVV-2013-267308.

References

1. Lorenz, E.N., Emanuel, K.A.: Optimal Sites for Supplementary Weather Observations: Simulation with a Small Model. J. Atmos. Sci. 55, 399–414 (1998)
2. Lorenz, E.N.: Designing Chaotic Models. J. Atmos. Sci. 62, 1574–1587 (2005)
3. Palmer, T., Hagegorn, R.: Predictability of Weather and Climate, 718 p. Cambridge University Press (2006)
4. Lorenz, E.N.: Atmospheric predictability as revealed by naturally occurring analogs. J. Atmos. Sci. 26, 636–646 (1969)
5. Lorenz, E.N.: Predictability: A problem partly solved. In: Proc. Seminar on predictability, vol. 1, pp. 1–18. CMWF, Reading (1996); reprinted Palmer T., Hagegorn R.: Predictability of Weather and Climate, 718 p. Cambridge University Press (2006)
6. Sprott, J.C.: Chaos and Time-Series Analysis, 507p. Oxford University Press, New York (2003)
7. Trevisan, A.: Impact of transient error growth on global average predictability measures. J. Atmos. Sci. 50, 1016–1028 (1993)
8. Benzi, R., Carnevale, F.C.: A possible measure of local predictability. J. Atmos. Sci. 46, 3595–3598 (1989)
9. Lorenz, E.N.: Atmospheric predictability experiments with a large numerical model. Tellus 34, 505–513 (1982)
10. Bengtsson, L., Hodges, K.I.: A note on atmospheric predictability. Tellus 58A, 154–157 (2006)
11. Trevisan, A., Malguzzi, P., Fantini, M.: A note on Lorenz's law for the growth of large and small errors in the atmosphere. J. Atmos. Sci. 49, 713–719 (1992)

EEE Method: Improved Approach of Compass Dimension Calculation

Vlastimil Hotař

Abstract. In our research has improved an approach of compass dimension calculating for topological one-dimensional objects (especially signals, time series or dividing lines). The method was named **E**valuation of length changes with **E**limination of insignificant **E**xtremes (EEE). The method stems from an estimation of the fractal dimension, so it measures changes of lengths in sequential steps. The EEE method does not use a fixed "ruler" for measurement in every step, but the line is defined by local extremes (maxima and minima). Mathematically generated functions (e.g. based on the Hurst coefficient), time series from real production processes and dividing lines (surface profiles and surface roughness) were used for experiments. The results show good potential for applications in off-line evaluations of data sets and on-line monitoring and control.

1 Introduction

Evaluation of signals (time series) from experiments, monitoring or production control are standard part of analyses. The choice of the analyses used for the monitoring should correspond with the character of the data obtained. The fractal dimension [1, 2] and a combination of statistical tools were used experimentally and are interesting and powerful tools for complex data quantification, for poor quality troubleshooting, production optimization and non-stability of systems troubleshooting in laboratories and in industrial applications [3].

The fractal dimension is closely connected to fractals that were defined by Benoit Mandelbrot [1], though scientists found some geometric problems with specific objects (e.g. the measurement of coast lines using different length of ruler by Richardson). A potentially powerful property of the FD is describing complexity by using a single number that defines and quantifies structures. The number is mostly a non-integer value and the FD is higher than the topological dimension. The compass dimension is a common and useful tool for an estimating the FD.

Vlastimil Hotař
Technical University of Liberec, Department of Glass Producing Machines and Robotics,
Studentska 2, 460 17 Liberec 6

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 343–351.
DOI: 10.1007/978-3-319-00542-3_35

The method is based on measuring the dividing line (roughness profile) using different sizes of rulers (fig. 1 A) according to the equation:

$$L_i(r_i) = N_i(r_i)\cdot r_i \tag{1}$$

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 that is given by the power law:

$$N(r_i) = const\cdot r_i^{-D_R}\ . \tag{2}$$

If the line is fractal and hence the estimated fractal dimension is larger than the topological dimension, the length measured increases as the ruler size is reduced. Using equations (1) and (2)

$$L_i(r_i) = N_i(r_i)\cdot r_i = const\cdot r_i^{-D_R}\cdot r_i = const\cdot r_i^{1-D_R} \tag{3}$$

D_R is the estimated dimension: the compass dimension.

The logarithmic dependence between $\log_2 L(r)$ and $\log_2 r$ is called the Richardson-Mandelbrot plot (Fig. 1 B). The compass dimension is then determined from the slope s of the regression line:

$$D_R = 1 - s = 1 - \frac{\Delta\log_2 L(r)}{\Delta\log_2 r} \tag{4}$$

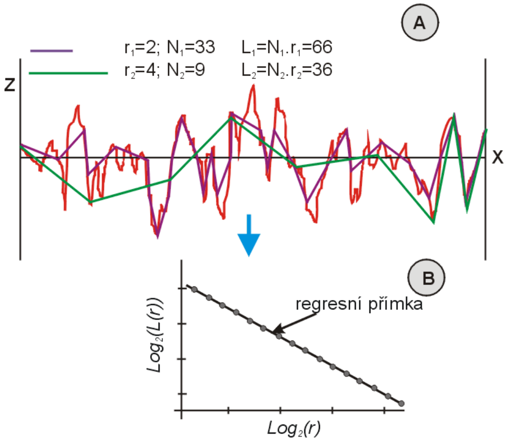


Fig. 1 Estimation of fractal dimension using Compass method

The developed method is an approach of compass dimension calculating: **E**valuation of length changes with **E**limination of insignificant **E**xtremes (EEE).

2 Methodology of EEE

The method is based on length evaluation of a curve (signal). The curve is defined by measured values and they are isolated points $x_1, x_2, \ldots, x_n$ in the range $y(x_1), y(x_2), \ldots, y(x_n)$. The points represent local extremes (maxima and minima). On the curve, unnecessary extremes are classified with a defined rule and a new simplified function is defined by the remaining points. The new function is used for the next classification.

An example of a function defined by points and connected into the linear by parts function *f* is in Fig. 2. In the first step the function *f* is purged of points which are not local extremes using the rule:

First, the difference proportion of the dependent variables *y* to the independent variables *x* between neighbouring points x_i, x_{i+1} is determined from function *f*:

$$\Delta f\left(x_i, x_{i+1}\right) = \frac{\Delta y(x_i)}{\Delta x_i} = \frac{y(x_{i+1}) - y(x_i)}{x_{i+1} - x_i} \tag{5}$$

Second, points where the difference Δf does not change sign are eliminated, because these points can be regarded as irrelevant. Remaining points are considered as the local extremes (black points in the Fig. 2). First and last points are added to the extremes. The simplified function *g* is generated from such received points, Fig 3. A relative length of the function *g*, is computed and the result is saved. The relative length $L_{rel\ 1}$ is evaluated from the absolute length of the function from point to point and divided by the length of its projection onto the x axis.

The procedure for the elimination of insignificant extremes is applied to the simplified function *g*. The procedure uses functions formed from maxima and minima of the function *g*, Fig. 4, the functions g_{max} and g_{min} extended with the first and last points of the function *g*. The function g_{max} is generated from the maxima

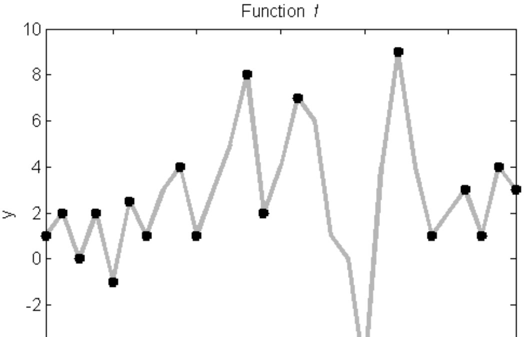


Fig. 2 Function f and its local extremes

of the function g (black line in Fig. 4). Local maxima are found in this function (black dots in Fig. 4). The definition of local maxima using above-mentioned rule, but only maxima are used. The function g_{min} is generated from minima and the local minima are found, using above-mentioned rule, but only minima are used.

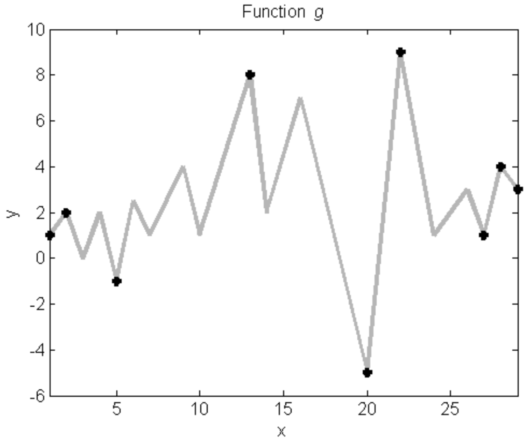


Fig. 3 Simplified function g and its local extremes

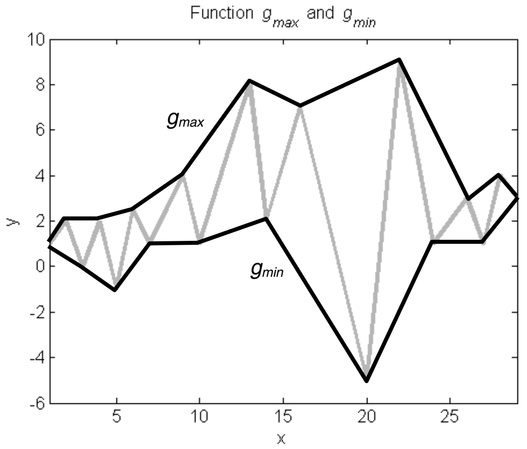


Fig. 4 Functions g_{max} and g_{min} generated from local extremes of function g

The local maxima and minima of the functions g_{max} and g_{min} are used for the generation of the function g_{red} , Fig. 5. In this function again local maxima and minima are defined using the rule (Fig. 5, black dots). These final local extremes of the function *g* (Fig. 3, black dots) and the first and last points from the function *g* define the function *h*, Fig. 6. The relative length $L_{rel\ 2}$ of function *h* is computed and the result is saved.

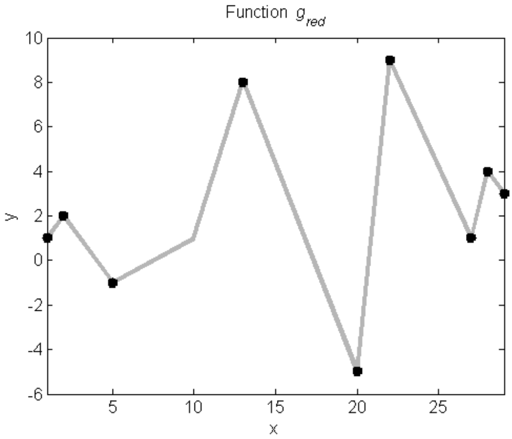


Fig. 5 Function g_{red} generated from maxima of function g_{max} and from minima of function g_{min}

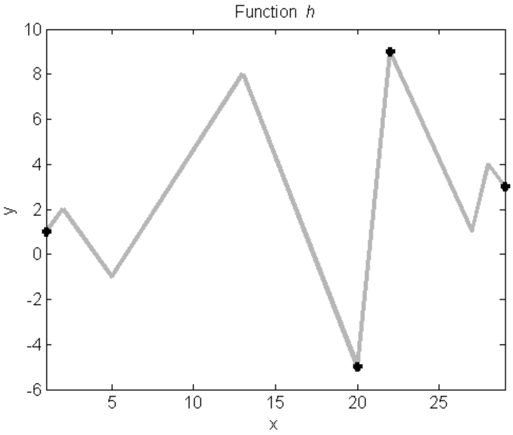


Fig. 6 Simplified function *h* and its local extremes

The same procedure is used for the new simplified function *h* and its global extremes and the first and last point (Fig. 6, black dots) define the function *k*. The function is formed from the global maximum and minimum of all functions (*f, g, h*), therefore the analysis is stopped. All functions are depicted in Fig. 7.

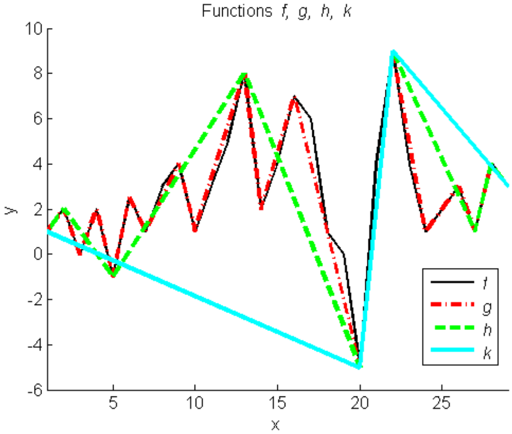


Fig. 7 Function *f* and simplified functions *g, h, k*

The steps *j* of the analysis are plotted versus the computed relative lengths $L_{rel\ j}$ of functions *g, h, k*, Fig.8. The relation between the relative lengths $L_{rel\ j}$ and the steps of elimination *j* are evaluated by a sufficient regression function that can be: a regression line (Fig. 8), a quadratic function or a hyperbolical function. The parameters of the regression functions are used for the evaluation of the function *f*. (Using logarithmic axis is not useful.)

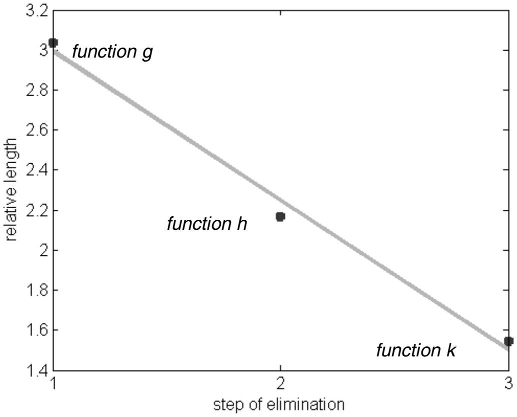


Fig. 8 Plot of the relation between steps of analysis and relative length, regression line

3 Experimentation with Randomly Generated Functions

Time series obtained from simulation of fractional Brownian motion using Cholesky-Factorization of the related covariance matrix (FBM) were used to test of the developed method. An example of testing time series is in Fig. 9 and is generated using the input Hurst coefficient H=0.4. The coefficient represents the character of time series and can be between value 0 and 1 (higher coefficients generate smoother functions, more information can be found in [2, 4]). The dependence between the relative lengths and the steps of elimination is in Fig. 10.

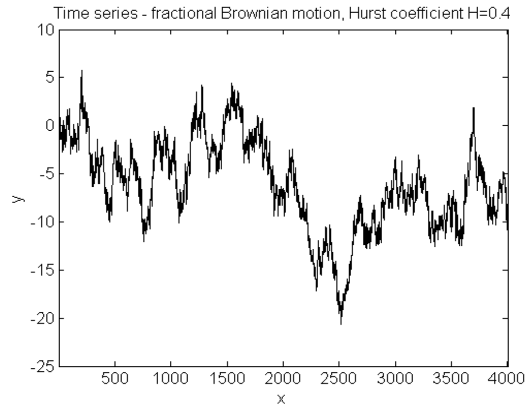


Fig. 9 Simulation of the time series using fractional Brownian motion

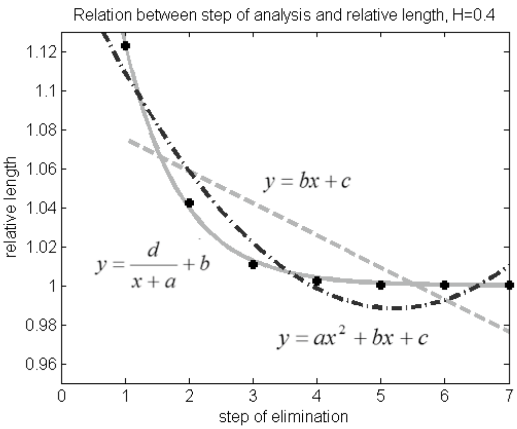


Fig. 10 Relation between number of steps of analysis and relative length for the time series FBM, H=0.4

A standard process to estimating of fractal dimension using dependence between $\log_2 L_{rel\ j}$ and $\log_2 j$ was tested, but this representation was not usable. The method does not use rulers with different lengths, and therefore the information cannot be used in the plot.

The best sufficient regression function between the relative lengths $L_{rel\ j}$ and the numbers of steps j is estimated by the hyperbolic regression model:

$$y = \frac{d}{x+a} + b \ . \tag{6}$$

It can be evaluated by parameters *d* and *a*. The parameter *b* is always *b*=1. The parameter *a* has to be computed numerically using an error function.

900 simulated time series from FBM were used for an evaluation of the method EEE with a Hurst coefficient between 0.1 and 0.95. Fig. 11 shows the dependence between Hurst coefficients and the average value of parameters *a* and *d*.

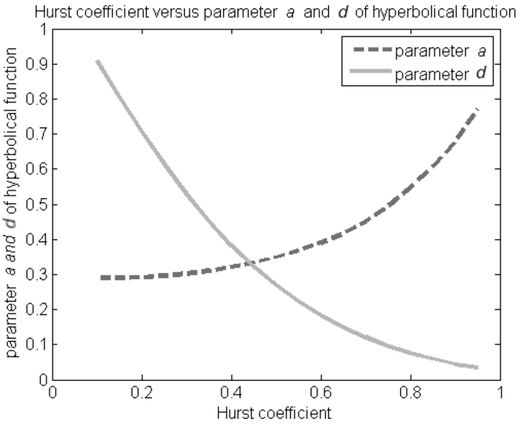


Fig. 11 Hurst coefficient versus average value of parameters *a* and *d* of hyperbolic function

4 Conclusion

The method EEE is an alternative to the compass dimension and describes character of curves and can be potentially used to classify their complexity. Currently, tests with real topological one-dimensional curves are being conducted. Time series from measurement, real production processes and the dividing line between two materials are being used. The results of the tests will be compared with parameters of statistical tools, fractal dimensions and spectral analysis. The correlation between estimating fractal dimension with standard methods and the EEE method have to be studied, and will be published. Important is also compared sensitivity. The first results have been promising.

Acknowledgments. This work was supported by the OP Research and Development for Innovation: Centre for Nanomaterials, Advanced Technologies and Innovation, CZ.1.05/2.1.00/01.0005 and by the grant of Students grant contest of the Technical University of Liberec, number SGS 2827/115, which use special-purpose support for the university research and is financed by the Ministry of Education, Czech Republic.

References

[1] Mandelbrot, B.B.: The fractal geometry of nature. W. H. Freeman and Co., New York (1982)

[2] Peitgen, H.O., Juergens, H., Saupe, D.: Chaos and Fractals: New Frontiers of Science. Springer, Heidelberg (1992)

[3] Hotař, V.: Fractal geometry for industrial data evaluation. Computers & Mathematics with Applications (2013)

[4] Evertsz, C.J.G., Peitgen, H.O., Voss, R.F.: Fractal Geometry and Analysis. World Scientific Publishing Co.Pte. Ltd., Singapore (1996)

Chaotic Analysis of the GDP Time Series

Radko Kříž

Abstract. The goal of this paper is to analyze the Czech Gross domestic product (GDP) and to find chaos in the Czech GDP. At first we will estimate the time delay and the embedding dimension, which is needed for the Lyapunov exponent estimation and for the phase space reconstruction. Subsequently we will compute the largest Lyapunov exponent, which is one of the important indicators of chaos. Then we will calculate the 0-1 test for chaos. Finally we will compute the Hurst exponent by Rescaled Range analysis and by dispersional analysis. The Hurst exponent is a numerical estimate of the predictability of a time series. In the end we will display a phase portrait of detrended GDP time series. The results indicated that chaotic behaviors obviously exist in GDP.

Keywords: Chaos theory, GDP, Time series analysis, Phase Space Reconstruction, Hurst exponent, largest Lyapunov exponent.

1 Introduction

Humanity has always been concerned with the question of whether the processes in the real world are deterministic in nature. Determinism can be understood variously. In this paper we assume a mathematical sense of determinism, which is given by equations and initial conditions. Mathematical models that are not deterministic because they involve randomness are called stochastic. Are the processes in the real world deterministic or stochastic in nature? Real processes in nature, according to the expectation of Mandelbrot [15], lie somewhere between pure deterministic process and white noise. This is why we can describe reality either by a stochastic or deterministic model. The Hurst coefficient can give us an answer to this.

An interesting case of determinism is deterministic chaos. The only purely stochastic process is a mathematical model described by mathematical statistics. The statistical model often works and is one of many possible descriptions if we do not know the system. This also applies to economic quantities, including forecasts for GDP. The basic question is therefore the existence of chaotic behavior. If the system behaves chaotically, we are forced to accept only limited predictions. In this paper we will try to show the chaotic behavior of GDP.

Radko Kříž
University of Pardubice, Pardubice, Czech Republic
e-mail: Radko.Kriz@upce.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 353–362.
DOI: 10.1007/978-3-319-00542-3_36

2 Methods of Analyzing

In short, we will describe the basic definitions and the basic methods for examining the input data.

2.1 Phase Space Reconstruction

According to Henry [9], the main goal in nonlinear time series analysis is to determine whether or not a given time series is of a deterministic nature. If it is, then further questions of interest are: What is the dimension of the phase space supporting the data set? Is the data set chaotic?

The key to answering these questions is embodied in the method of phase space reconstruction, which has been rigorously proven by the embedding theorems of Takens [19]. Takens theorem was independently suggested for example Packard [17]. Takens' theorem transforms the prediction problem from time extrapolation to phase space interpolation. Takens' embedding theorem asserts that measured time series need not be components of the attractor, but only a sufficiently smooth transformation or maps of the component or components (so called measurement function) of the dynamical system under study.

Let there be given a time series $x_1, x_2, \dots, x_N$ which is embedded into the m-dimensional phase space by the time delay vectors. A point in the phase space is given as:

$$Y_n = x_n, x_{n-\tau}, \dots, x_{n-(m-1)\tau} \quad n = 1,2,\dots,N-(m-1)\tau \tag{1}$$

where τ is the time delay and m is the embedding dimension. Different choices of τ and m yield different reconstructed trajectories. How can we determine optimal τ and m?

2.2 Optimal Time Delay

A one-to-one embedding can be obtained for any value of the time delay $\tau > 0$. However, very small time delays will result in near-linear reconstructions with high correlations between consecutive phase space points and very large delays might obscure the deterministic structure linking points along a single degree of freedom. If the time delay is commensurate with a characteristic time in the underlying dynamics, then this too may result in a distorted reconstruction.

In order to estimate τ, two criteria are important according to Kodba [12]. First, τ has to be large enough so that the information we get from measuring the value of x at time $n + \tau$ is significantly different from the information we already have by knowing the value of x at time n. Only then will it be possible to gather enough information about all other system variables that influence the value of x to reconstruct the whole attractor. Second, τ should not be larger than the typical time in which the system loses memory of its initial state. This is particularly important for chaotic systems, which are intrinsically unpredictable and hence lose memory of the initial state as time progresses.

Following this reasoning, Fraser and Swinney [3] introduced the mutual information between x_n and $x_{n+\tau}$ as a suitable quantity for determining τ. The mutual information between x_n and $x_{n+\tau}$ quantifies the amount of information we have about the state $x_{n+\tau}$ presuming we know the state x_n. Now we can define mutual information function:

$$I(\tau) = -\sum_{h=1}^{j}\sum_{k=1}^{j} P_{h,k}(\tau)\ln\frac{P_{h,k}(\tau)}{P_h P_k} \tag{2}$$

where P_h and P_k denote the probabilities that the variable assumes a value inside the h^{th} and k^{th} bins, respectively, and $P_{h,k}(\tau)$is the joint probability that x_n is in bin h and $x_{n+\tau}$is in bin k. Hence, the first minimum of $I(\tau)$ marks the optimal choice for the time delay.

2.3 Optimal Embedding Dimension

The embedding dimension m is conventionally chosen using the "false nearest neighbors'" method. This method measures the percentage of close neighboring points in a given dimension that remain so in the next highest dimension. The minimum embedding dimension capable of containing the reconstructed attractor is that for which the percentage of false nearest neighbors drops to zero for a given tolerance level ε.

In order to calculate the fraction of false nearest neighbors the following algorithm is used according to Kennel [11]. Given a point $p(i)$ in the m-dimensional embedding space, one first has to find a neighbour $p(j)$, so that

$$\|p(i) - p(j)\| \le \varepsilon \tag{3}$$

We then calculate the normalized distance R_i between the $(m + 1)th$ embedding coordinate of points $p(i)$ and $p(j)$ according to the equation:

$$R_i = \frac{|x_{i+m\tau} - x_{j+m\tau}|}{\|p(i) - p(j)\|} \tag{4}$$

If R_i is larger than a given threshold R_{tr} , then $p(i)$ is marked as having a false nearest neighbor. Equation (4) has to be applied for the whole time series and for various $m = 1, 2, \ldots$ until the fraction of points for which $R_i > R_{tr}$ is negligible [12].

2.4 The Largest Lyapunov Exponent

Lyapunov exponent λ of a dynamical system is a quantity that characterizes the rate of separation of infinitesimally close trajectories. Quantitatively, two trajectories in phase space with initial separation δZ_0 diverge.

$$\delta Z(t) \approx e^{\lambda t}|\delta Z_0| \tag{5}$$

The largest Lyapunov exponent can be defined as follows:

$$\lambda = \lim_{\substack{\delta Z_0 \to 0 \\ t \to \infty}} \frac{1}{t} \ln \frac{|\delta Z(t)|}{|\delta Z_0|} \tag{6}$$

The limit *δZ0→0* ensures the validity of the linear approximation at any time. Largest Lyapunov exponent determines a notion of predictability for a dynamical system. A positive largest Lyapunov exponent is usually taken as an indication that the system is chaotic (provided some other conditions are met, e.g., phase space compactness) [14].

We have used the Rosenstein algorithm, which counts the largest Lyapunov exponent as follows:

$$\lambda_1(i) = \frac{1}{i\Delta t} . \frac{1}{(M-i)} \sum_{j=1}^{M-i} \ln \frac{d_j(i)}{d_j(0)} \tag{7}$$

Where $d_j(i)$ is distance from the j point to its nearest neighbor after i time steps and M is the number of reconstructed points. For more information see [6, 18].

2.5 The 0-1 Test for Chaos

New test for the presence of deterministic chaos was developed by Gottwald & Melbourne [7]. Their '0 - 1 test for chaos takes as input a time series of measurements, and returns a single scalar value usually in the range 0 - 1. In contrast the 0 - 1 test does not depend on phase space reconstruction but rather works directly with the time series given. The input is the time-series data and the output is 0 or 1, depending on whether the dynamics is non-chaotic or chaotic.

Briefly, the 0-1 test takes as input a scalar time series of observations $\varphi_1, \dots,$ φ_N. We have used the algorithm according to Dawes & Freeland [2]. First, we must fix a real parameter c and construct the Fourier transformed series:

$$z_n = \sum_{j=1}^{n} \phi_j e^{ijc}, \quad n = 1, \dots, N \tag{8}$$

Then we have computed the smoothed mean square displacement:

$$M_c(n) = \frac{1}{N-p} \sum_{j=1}^{N-p} \left| z_{j+n} - z_j \right|^2 - \left(\sum_{k=1}^{N} \frac{\phi_k}{N} \right)^2 \frac{1-\cos nc}{1-\cos c} \tag{9}$$

Finally we have estimated correlation coefficient to evaluate the strength of the linear growth

$$r_c = \frac{\operatorname{cov}(n, M_c(n))}{\sqrt{\operatorname{cov}(n,n)\operatorname{cov}(M_c(n), M_c(n))}} \tag{10}$$

2.6 Long Memory in Time Series

Hurst exponent (*H)* is widely used to characterize some processes. Hurst exponent is used to evaluate the presence or absence of long-range dependence and its degree in a time-series. For more information see [8, 10]. The Hurst exponent is a measure that has been widely used to evaluate the self-similarity and correlation properties of fractional Brownian noise, the time-series produced by a fractional Gaussian process [16]. We can describe self-similarity process following equation:

$$X(at) = a^{H} X(t) \tag{11}$$

where a is a positive constant, and H is the self-similarity parameter, for $0 < H < 1$.

We have used two methods for computing long memory in GDP time series. First, we have used a methodology known as Rescaled Range analysis or R/S analysis. To calculate the Hurst exponent, one must estimate the dependence of the rescaled range on the time span n of observation. The Hurst exponent is defined in terms of the asymptotic behavior of the rescaled range as a function of the time span of a time series as follows:

$$E\left[\frac{R(n)}{S(n)}\right] = Cn^{H} \text{ as } n \rightarrow \infty \tag{12}$$

Where *[R(n)/S(n)]* is the rescaled range; *E[y]* is expected value; *n* is number of data points in a time series, *C* is a constant. For more information see [13].

Second method is dispersional analysis. This method was introduced by Bassingthwaighte [1] In the original algorithm, the *x(t)* series is divided into non-overlapping intervals of length *n*. The mean of each interval is computed, and then the standard deviation (*SD*) of these local means, for a given length *n*. These computations are repeated over all possible interval lengths. *SD* is related to *n* by a power law:

$$SD \propto n^{H-1} \tag{13}$$

3 Analysis of GDP Time Series

3.1 Input Data

Gross domestic product by type of expenditure in current prices is used in this paper. We have used data (quarterly, without seasonal adjustment) from the Czech Statistical Office. We analyze data from the Czech Republic between the years 1995 -2012 (cf. Figure 1).

The main problem in analyzing the GDP time series is the lack of data. We have only 72 values for the Czech GDP time series. The analysis of such short time series in the context of nonlinear dynamics or in the presence of chaos can be questionable. We know, according to Horák [4] or Galka [5], that for this kind of method results are provable for at least 10^3 data-points. Analysis of short time

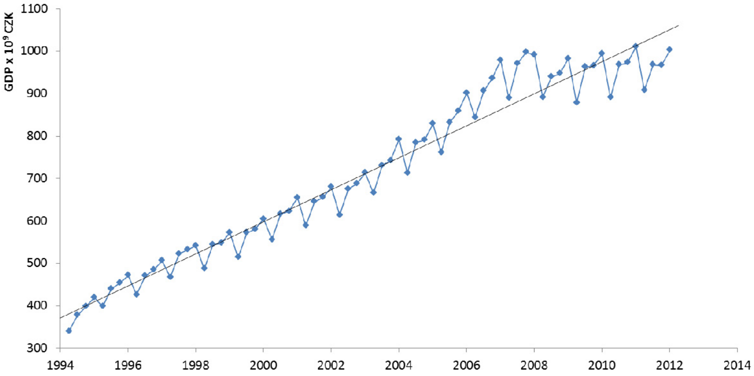


Fig. 1 GDP with linear trend of GDP

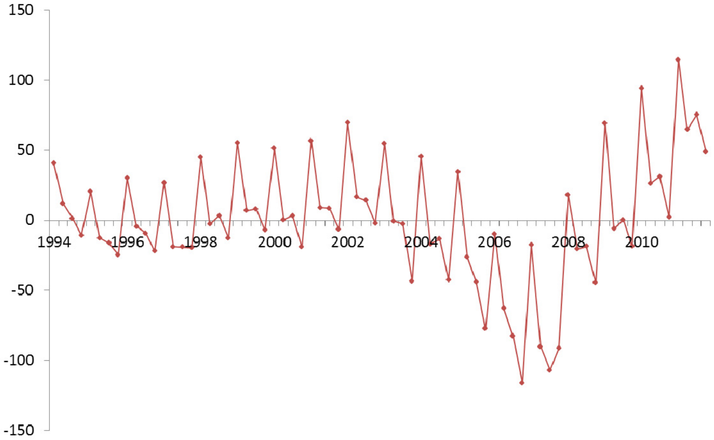


Fig. 2 GDP without trend

series (order of 10^1) may lead to a spurious estimation of the invariants e.g. Lyapunov exponents. Despite the above, we have no choice but to analyze GDP time series in the context of nonlinear dynamics and try to find chaotic behavior of GDP time series.Therefore, all results are only estimates.

Trended data are not suitable for future analysis to study chaos dynamics. There is no universal way to remove the trend from the data set. The results often depend strongly on how the data are detrended. The trend is removed by subtracting the linear interpolation. Denote GDP without trend as *Y(t)* (cf. Figure 4).

3.2 Calculation of the Time Delay

In this chapter we will use the mutual information approach to determine the time delay. This approach is described above. This variable is estimated from the graph

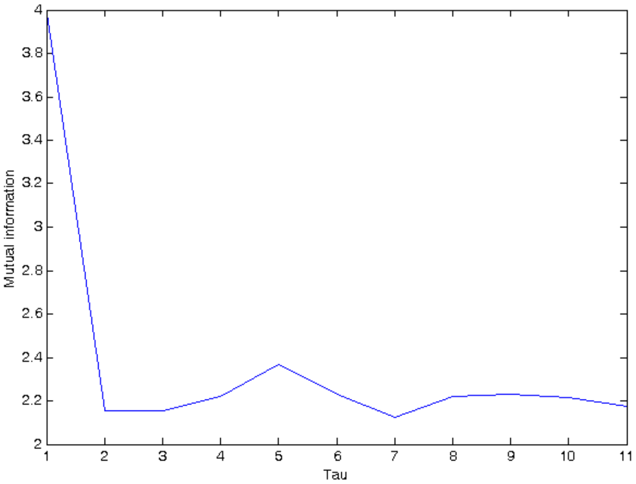


Fig. 3 Estimation of the time delay

(cf. Figure 3). The first minimum of the mutual information function *I(τ)* (2) marks the optimal choice for the time delay. Thus, the time delay *τ* is 2.

3.3 Calculation of the Embedding Dimension

In this chapter we will use the false nearest neighbor method to determine the minimal sufficient embedding dimension. The embedding dimension *m* is chosen using the “false nearest neighbors’” method. This variable is estimated from the graph (cf. Figure 4). The minimum embedding dimension capable of containing the reconstructed attractor is that for which the percentage of false nearest neighbors drops to zero for a given tolerance level ε. Thus, the embedding dimension *m* is 4, but the value 2 can be sufficient.

3.4 Calculation of the Largest Lyapunov Exponents

In this chapter we calculate the largest Lyapunov exponent as was shown above. We used the Rosenstein algorithm. The calculation of the largest Lyapunov exponent depends on the estimation of the embedding dimension. Importantly, for every relevant embedding dimension value (2 - 10) there is a positive largest Lyapunov exponent. A positive largest Lyapunov exponent is one of the necessary conditions for chaotic behavior. This shows that the GDP evolution is sensitive to the initial conditions. The value of the largest Lyapunov exponent was estimated at 0,004 for embedding dimension 4. Notice that the largest Lyapunov exponent is relatively small. Consequently, the rate of GDP evolution is rather slow, showing that it is possible to accurately make a short-term forecast.

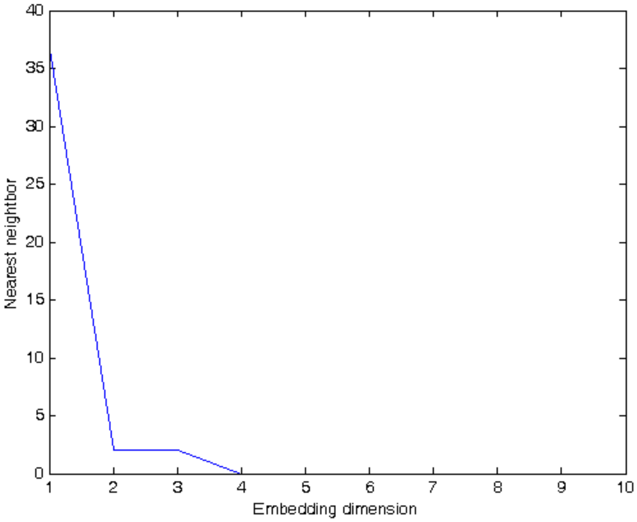


Fig. 4 Estimation of the embedding dimension

3.5 Results of the 0-1 Test for Chaos

In this chapter we calculate the correlation coefficient as was shown above. The value of the correlation coefficient was computed at 0,88. The correlation coefficient is near to 0 for non-chaotic data and near 1 for chaotic data. The value 0,88 is closer to 1. Hence we can assume to chaotic behavior in the Czech GDP time series.

3.6 Calculation of the Hurst Exponent

The Rescaled Range analysis gave us the value of the Hurst exponent 0,96 and the dispersional analysis gave us the value 0,87. Both values indicate the presents of long memory in detrended GDP time series. Those values are in accordance with our expectations. We know that the value of H is between 0 and 1, whilst real time series are usually higher than 0,5. If the exponent value is close to 0 or 1, it means that the time-series has long-range dependence. We can assume that the true value lies somewhere between those values. We think that those values are sufficient for a credible prediction. Now we also know that the fractal dimension $D_F = 2\text{-}H$. We have estimated the value of the fractal dimension between 1,04 and 1,13.

3.7 Phase Portrait of GDP Time Series

Phase portrait 2D of GDP is constructed so that each ordered pair of $\{Y_t;Y_{t-2}$, $t=3,\ldots,N\}$ is displayed in the plane where the x-axis represents the values of Y_t and y-axis value Y_{t-2} (cf. Figure 5). The individual points $\{Y_t;Y_{t-2}\}$ of phase space are connected by a smooth curve. This curve looks like a chaotic attractor.

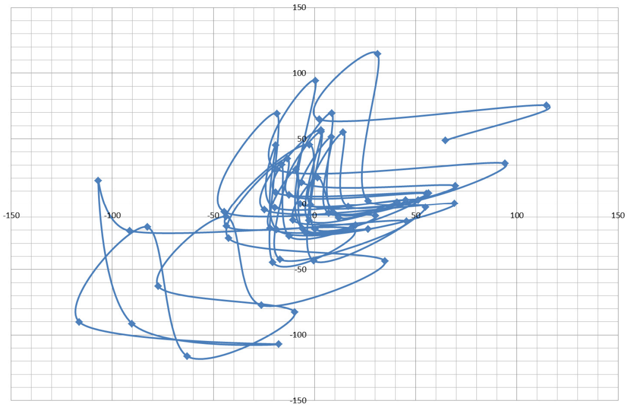


Fig. 5 GDP in 2D phase space (Y_t, Y_{t-2})

4 Conclusions

Chaos theory has changed the thinking of scientists and the methodology of science. Making a theoretical prediction and then matching it to the experiment is not possible in chaotic processes. Long term forecasts are, in principle, also impossible according to chaos theory. The main problem is in the quantity and quality of data. Some improvement of measurement cannot help us adequately, because it is a fight against power of exponential rate. Nonlinear dynamics and chaos theory have also corrected the old reductionist tendency in science. Now it is known that real processes are nonlinear and a linear view can be wrong. The basic question is therefore - the existence of chaotic behavior. If the system behaves chaotically, we are forced to accept only limited predictions. But it is much better than random processes.

We have shown in this paper that the GDP time series is chaotic and contains long memory. First, we computed the values of the time delay $\tau = 2$ and the embedding dimension $m = 4$. The estimated largest Lyapunov exponent is 0,004. If the fractal dimension is low, the largest Lyapunov exponent is positive and the Kolmogorov entropy has a finite positive value, chaos is probably present. Then, we conducted the 0-1 test for chaos and chaos is present according to this test. From these estimations it can be concluded that the GDP time series is chaotic. Long memory was deduced conclusively from the calculation of the values of the Hurst exponent.

We know that the main problem when analyzing GDP time series is the lack of data. In the future we would like to focus on the proper statistical significance for nonlinearity and on predicting the GDP. In particular, the surrogate data approach (e.g. Theiler et al. [20]) is a powerful tool for detecting actual nonlinear behavior, and distinguishing it from other phenomena.

References

[1] Bassingthwaghte, J.B.: Physiological heterogeneity: Fractals link determinism and randomness in structure and function. News in Physiological Sciences 3, 5–10 (1988)

[2] Dawes, J.H.P., Freeland, M.C.: The '0–1 test for chaos' and strange nonchaotic attractors (2008), http://people.bath.ac.uk/jhpd20/publications

[3] Fraser, A.M., Swinney, H.L.: Independent coordinates for strange attractors from mutual information. Phys. Rev. A 33, 1134–1140 (1986)

[4] HoráK, J., KrlíN, L., Raidl, A.: Deterministický chaos a jeho fyzikální aplikace, Academia, Praha, 437 (2003)

[5] Galka, A.: Topics in Nonlinear Time Series Analysis. World Scientific (2000)

[6] Gotthans, T.: Advanced algorithms for the analysis of data sequences in Matlab, Master's Thesis, University of technology Brno (2010)

[7] Gottwald, G.A., Melbourne, I.: A new test for chaos in deterministic systems. Proc. Roy. Soc. A 460, 603–611 (2004)

[8] Grassberg, P., Procaccia, I.: Characterization of strange attractors. Phys. Rev. Lett. 50, 346 (1983)

[9] Henry, B., Lovell, N., Camacho, F.: Nonlinear dynamics time series analysis. In: Akay, M. (ed.) Nonlinear Biomedical Signal Processing, pp. 1–39. Insititue of Electrical and Electronics Engineers, Inc. (2001)

[10] Hurst, H.E.: Long term storage capacity of reservoirs. Trans. Am. Soc. Eng. 116, 770–799 (1951)

[11] Kennel, M.B., Brown, R., Abarbanel, H.D.I.: Determining embedding dimension for phase space reconstruction using a geometrical construction. Phys. Rev. A 45, 3403–3411 (1992)

[12] Kodba, S., Perc, M., Marhl, M.: Detecting chaos from a time series. European Journal of Physics 26, 205–215 (2005)

[13] Kříž, R.: Chaos in GDP. Acta Polytechnica 51(5) (2011)

[14] Lorenz, H.-W.: Nonlinear Dynamical Economics and Chaotic Motion. Springer (1989)

[15] Mandelbrot, B.B.: The Fractal Geometry of Nature. W.H. Freeman and Co. (1983)

[16] Mandelbrot, B.B., Van Ness, J.W.: Fractional Brownian motions, fractional noises and applications. SIAM Rev. 10, 422 (1968)

[17] Packard, N.H., Crutchfield, J.P., Farmer, J.D., Shaw, R.S.: Geometry from a time series. Phys. Rev. Lett. 45, 712–726 (1980)

[18] Rosenstein, M.T., Collins, J.J., Luca, C.J.: A practical method for calculating largest Lyapunov exponents from small data sets. Physica D 65, 117–134 (1993)

[19] Takens, F.: Detecting Strange Attractor in Turbulence. In: Rand, D.A., Young, L.S. (eds.) Dynamical Systems and Turbulence, Warwick 1980. Lecture Notes in Mathematics, vol. 898, p. 366. Springer, Berlin (1981)

[20] Theiler, J., Eubank, J., Longtin, A., Galdrikian, B., Farmer, J.D.: Testing for nonlinearity in time series: The method of surrogate data. Physica D 58, 77 (1992)

Daily Temperature Profile Prediction for the District Heating Application

Juraj Koščák, Rudolf Jakša, Rudolf Sepeši, and Peter Sinčák

Abstract. We show an application of artificial neural networks for local weather prediction. By employment of appropriate network structure and proper selection of input/output signals, solid results can be achieved. Our system was implemented in the local district heating company, where it was used to predict daily temperature profile with period of 15 minutes. Further, weekly and yearly profiles were predicted, and also heat consumption profiles. Whole prediction system consists of several chained neural networks and data processing modules. Training data for neural networks were collected from meteorological stations around the Košice city. Additional training data were collected by web-robots from internet from several weather forecast agencies.

1 Introduction

Local weather prediction with neural networks is based on approximation of weather function by black-box model from weather data collected in particular local region. This will create the weather model for single place on the map. Weather forecast agencies produce forecasts for bigger regions or even for continents or whole globe [3]. The local prediction method is not practical for forecasts agencies. However, for local applications, where we are interested in local weather course, this approximation-based weather prediction can be useful. It is very flexible and can produce different types of results not available elsewhere. For instance it is possible to predict weather dependent technology signals, like the heat consumption [1]. Also prediction with short 15 minutes period is possible. Utilization of locally

Juraj Koščák · Rudolf Jakša · Peter Sinčák
Center for Intelligent Technologies, Technical University of Košice, Slovakia,
e-mail: jurajkoscak@gmail.com, jaksa@neuron.tuke.sk,
peter.sincak@tuke.sk

Rudolf Sepeši
Tepláreň Košice a.s., Košice, Slovakia

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 363–371.
DOI: 10.1007/978-3-319-00542-3_37

measured data is possible too. The local weather prediction can be used as an extension and refinement of generally available weather forecast.

In our case we will focus on next day prediction of air temperature with 15 minutes period. We call it Daily Temperature Profile (TPD). It is a course of temperature during a single day. Our predictor system will look into weather data from last couple of days and according to them it will produce TPD prediction for tomorrow. In addition to TPD prediction we will provide weekly and yearly prediction. And in addition to temperature we will predict the technology-related signals - the heat consumption in the city. Our data, which we will use in prediction system, are company own weather and technology data, public data about current weather condition in area, and publicly available weather forecast data for next days.

2 Neural Networks Based Weather Prediction System

Our system was implemented for local heating company Tepláreň Košice a.s. which provides heat for significant part of the 230 000 citizens of Košice city. Most of load is during the winter period, when it is used for heating of buildings. Main variable for the heat demand is thus the air temperature. If we know the future temperature, all the heat demand and technology variables, with some accuracy, might be derived from it. Using predicted temperature, the company is able to heat up whole system on time using just required amount of energy. Better accuracy of prediction brings economical and also to ecological advantages.

Our prediction system had to be implemented into already existing infrastructure of district heating company system (see Fig.1). As the data source, we used data management system D2000, where all collected data from sensors, meteorological stations and web-robots are stored. Predicted values are sent back to D2000 to be

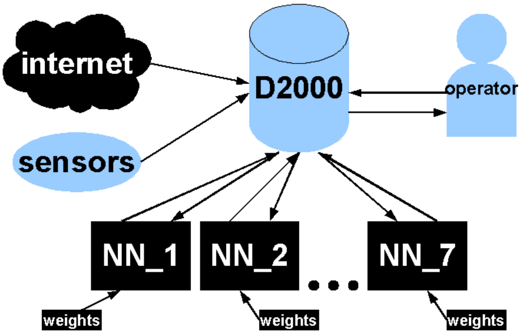


Fig. 1 Implementation of prediction system in district heating company infrastructure. The D2000 is the data management system which we had available to connect to sensors data and to operators. Prediction system consists of seven neural network prediction modules and web-robots for collecting of weather forecast data from internet.

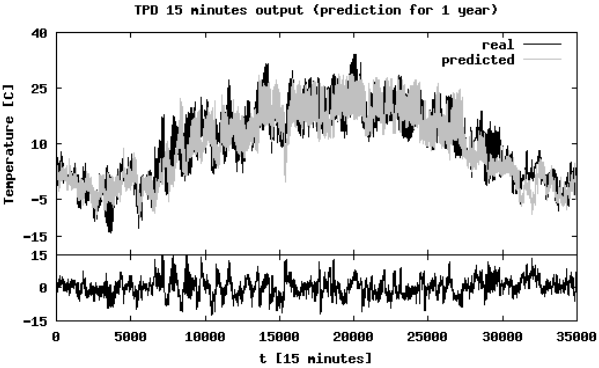


Fig. 2 Test of main TPD prediction (network *NN1*) on one year data from 1st of January to 31st of December, real versus predicted values. The bottom curve is the error, difference between predicted and real values.

available for the technology, for operators, and also, when needed, as inputs to another neural networks among our prediction modules.

The system consists of seven neural networks (see list on Tab. 2). Some of them are used in chained configuration. For instance *NN3* can be used to further process

Table 1 Topology is the number of inputs, hidden units and outputs. Training patterns: 100 patterns from all 1458 were used for training. The number of training cycles is "Iterations". The γ is learning rate of backpropagation algorithm, α is momentum parameter.

	Topology	Train. Patterns	γ	α	Iterations
NN1	1008 - 48 - 96	100 from 1458	0.1	0.1	600
NN2	5616 - 192 - 96	100 from 105	0.3	0.1	500
NN3	9 - 200 - 96	100 from 1458	0.1	0.1	400
NN4	312 - 110 - 730	3 from 10	0.1	0	300
NN5	63 - 25 - 14	4717	0.01	0.5	300
NN6	1104 - 48 - 96	100 from 1458	0.2	0.3	300
NN7	1032 - 48 - 96	100 from 1458	0.2	0.3	300

Table 2 List of neural networks in prediction system

	Predictor	Outputs	Inputs
NN1	TPD - 24hours/15min period	$24 \times 4 = 96$	company data
NN2	TPD with web-robots inputs	$24 \times 4 = 96$	comp. data + web robots
NN3	TPD using min/max data only	$24 \times 4 = 96$	min/max data
NN4	Min/Max Temp. 365 days	$365 \times 2 = 730$	past years data
NN5	Min/Max Temp. 7 days	$7 \times 2 = 14$	past days data
NN6	Steam amount	$24 \times 4 = 96$	company data and TPD
NN7	Steam power	$24 \times 4 = 96$	company data and TPD

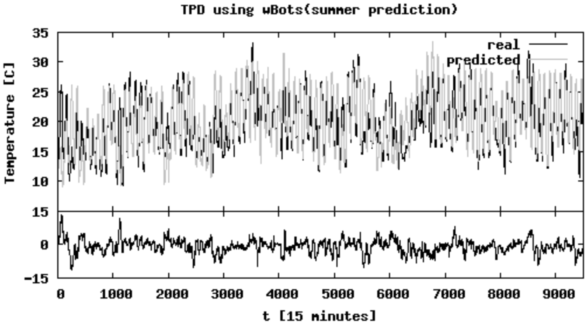


Fig. 3 Test of TPD prediction using weather forecast data from web-robots (network *NN2*) on one 2-months data. Difference between predicted and real values is the bottom curve.

outputs from *NN4*. Five networks are designed to predict temperature for different time periods with different accuracy. We have network for prediction of whole year temperature (365 days), for 7days temperature, for daily temperature profile with and without web-robots data, and for daily temperature profile from minimal and maximal temperatures in past days (Tab. 2). Two networks predict the steam amount and amount of steam power used in heating processes per day with 15 minutes period. In this application we used common backpropagation-momentum version of algorithm [7] [2] [8] [9]. Our further research with weather prediction was focused on stochastic weight update learning including recurrent neural networks [4] [5] [6].

All neural networks are standard feed-forward multilayer perceptron networks with one hidden layer. Number of input and output neurons varied according to type of prediction and required time periods (see Tab. 1). Networks inputs are described in the Tab.3. Weather variables are measured air temperatures, humidity, wind speed, solar irradiation and precipitation. Technology variables are several temperatures, amounts and powers parameters from the heat production processes. Date variables describe the date and the time of actual sample, the day is computed as the time-distance from reference day in middle of summer. Weather forecast data

Table 3 Input data description (also see Tab.1 network topologies)

	No. of inputs	Description
NN1	$36 \times 28 = 1008$	36 samples of 28 weather, technology and date variables
NN2	$144 \times 39 = 5616$	144 samples of 39 weather, technology and date variables
NN3	$3 \times 3 = 9$	min/max temperature and day-in-year ID from past 3 days
NN4	$2 \times 3 \times 52 = 312$	min/max temperatures for 52 weeks for 3 years
NN5	$3 \times 7 \times 3 = 63$	min/max temperature and day-ID for 3 weeks
NN6	$1008 + 96 = 1104$	*NN1* inputs plus next day TPD data with 15min period
NN7	$1008 + 24 = 1032$	*NN1* inputs plus TPD data with 1 hour period

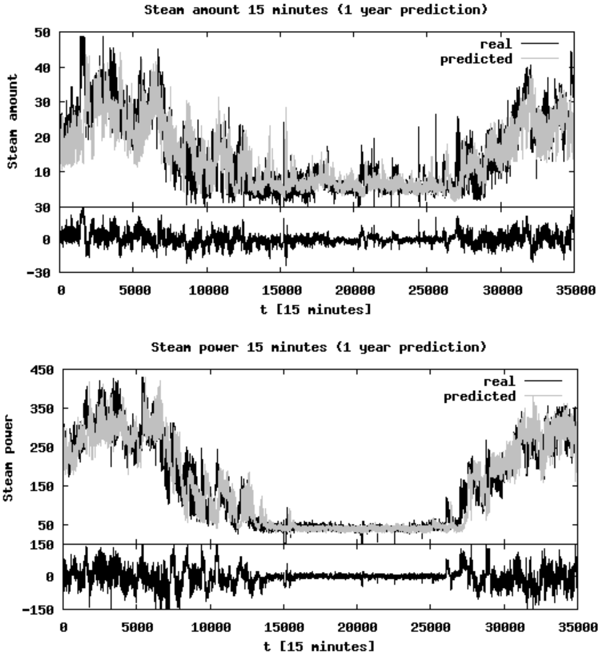


Fig. 4 Test of steam data prediction (networks *NN6* and *NN7*) on one year data from 1st of January to 31st of December, real versus predicted values. Error is displayed in bottom parts of frames.

from web-robots consist of min/max temperature forecasts from three agencies, and wind and humidity forecasts. We used linear and also nonlinear time-windows of data on input, nonlinear sampling was for instance every hour from yesterday, every second hour from day before, and 4 samples from 2days before.

System was trained using various data, which could influence temperature in the Košice region. To incorporate weather trends data into process, we used data from different meteorological stations around Košice, and in the network *NN2* also data collected from three weather forecast agencies through internet using web-robots. Web robots are implemented as system daemons, periodically (hourly) polling web-pages with forecasts. Downloaded pages are parsed by Java and Perl programs using syntactic analysis of html code. Extracted data are stored back into system database, from where they are queried by prediction modules when needed.

3 Predictors Configuration

NN1 Whole data set for *NN1* network consist of measured values for past 4 years. Training set was represented by randomly chosen 100 days. For testing all 4 years data were used. *NN1* input are several measured values of weather

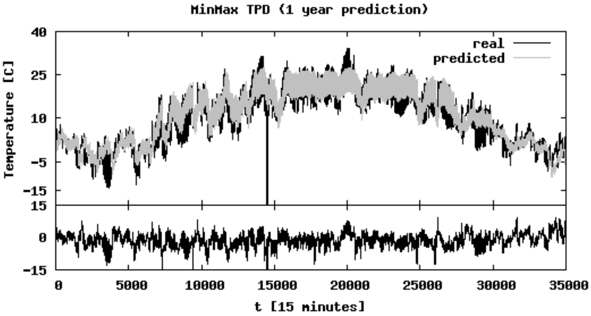


Fig. 5 Test of *NN3* for TPD prediction on one year data using min/max temperature data only. Bottom curve is the error plot.

Table 4 Errors on training and on testing data. The training errors are measured after the displayed number of training cycles. Testing errors, although more important then the training errors, are sometimes measured on bigger testing set, sometimes on smaller one, depends on how much data we had available for particular type of prediction. For *NN2* we didn't have enough data to build any meaningful testing data set.

	Training	Testing	Cycles
NN1	4.1%	11.4%	600
NN2	4.1%	-	500
NN3	7.7%	10.3%	400
NN4	10.2%	13.1%	300
NN5	12.8%	12.9%	300
NN6	17.7%	48.4%	300
NN7	10.5%	24.1%	300

variables from 3 days from the history. See tables Tab.2 and Tab.1 for details, the Fig.2 illustrates performance of *NN1* on one year test data.

NN6,7 The results from *NN1* network are part of input of *NN6* steam amount prediction network. The rest of *NN6* inputs are the same as *NN1*. The accuracy of prediction of *NN6* is actually influenced by the accuracy of the *NN1* temperature prediction. Similar is the *NN7* network for steam power prediction. The *NN7* output is the steam power. See Fig.4 for performance on one year data.

NN2 Network *NN2* is using web-robots data from weather forecasts. This should be an advantage over local-only-data network *NN1*. Unfortunately we had not enough weather-forecast data, so this network was trained only on few months, and thus it is usable only for prediction during this summer period. Its output is not as stable as of networks trained using 4-years data. See Fig.3 for results illustration and the Fig.7 for example of different behavior of *NN2* and *NN1*.

NN3 The *NN3* has the simplest topology, it's inputs are only minimum and maximum temperature of the 3 days before. Output is the same 24 × 4 as with the rest of TPD predictors. This network is used to derive daily course of

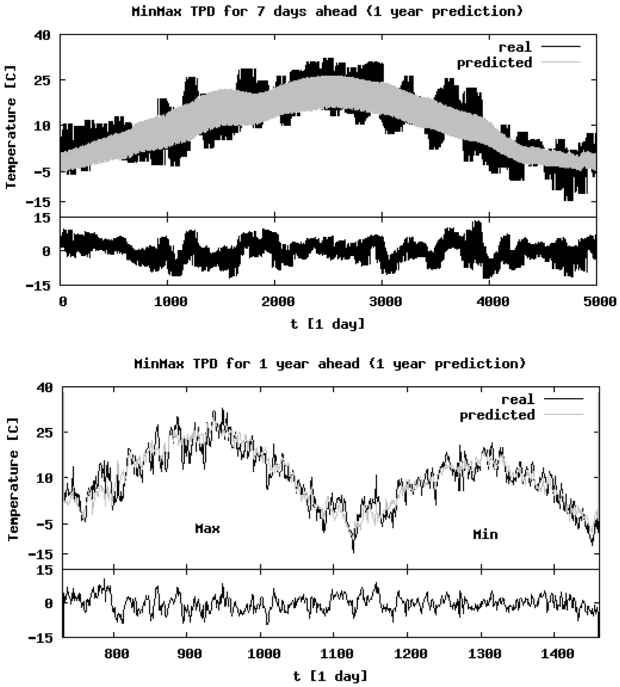


Fig. 6 Long-term prediction networks, *NN5* for 7 days TPD prediction an *NN4* for 365 days prediction. Both produce two values for every day of output – minimal and maximal temperature this day. The plot for 365 days predictor is split: left side are maximal temperatures for whole year, right halve are minimal temperatures for whole year. Error is displayed in bottom parts of frames.

temperature from the min/max temperature in the past days. This prediction is not as accurate as the other two TPD predictors, but can be used for days in future, for which we have only the forecast data available, not the measured history.

NN5 The *NN5* is used for 7-days prediction. This network can be combined with *NN3* for TPD prediction from minimum and maximum data which themselves are produced by *NN5*. Input for this network are minimum and maximum temperatures from past three weeks and the output is predicted minimum and maximum temperature for next 7 days. See Fig.6 for the performance. It was trained on 12 years data.

NN4 *NN4* is for long-term prediction. We are trying to predict weather course for the whole year. The purpose is to support operators when they make preparations of heating system for next year. This 365-days prediction can be used the same way as the 7-days prediction in combination with *NN3* network to produce TPD outputs with 15 minutes period. Training data set consist of three years with minimum and maximum temperature for every week. Current year prediction is derived from this year already recorded data, from data from past year and two

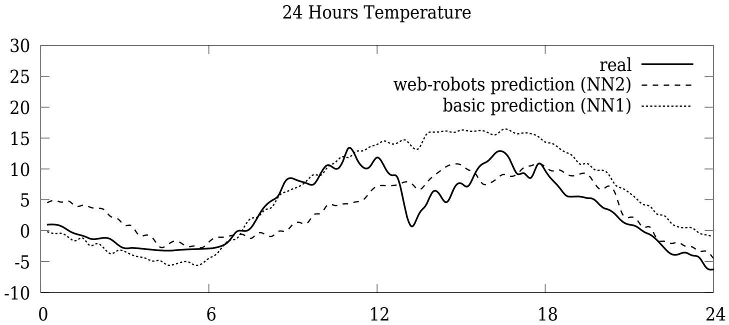


Fig. 7 Example of effect of using web-robots weather forecast data on the prediction. The dotted line is basic prediction, based on weather history for this day from past years and the weather in past three days. This curve is smooth because it models historical average for given conditions. However, it missed the sudden weather change on noon. Dashed web-robots prediction, although not as fitting in the morning, and not as smooth (probably because of lack of training data), did model the second half of day better by using the forecast information about possible weather change.

years before. Missing data for current year are substituted with average values from previous years for particular weeks. See Fig.6 for the *NN4* performance.

4 Results

Achieved prediction accuracies are shown in Table 4. Considering the type of prediction of every network, the *NN1* predictor is the most accurate one, and we used it as the primary source of prediction for heating company. Combinations of other predictors are used when the *NN1* cannot be used. In future, when we have more weather forecast data collected, we believe the *NN2* might outperform the *NN1* predictor.

During experiments we found the temperature to be most influencing factor from all the input data on the prediction accuracy. In time-windows, the most influencing part were the last three-four hours of past temperature course. Technology variables were in general less influencing compare to weather variables, however, if we skipped them the accuracy decreased a little.

5 Conclusion

Our experience with neural networks based local weather prediction for the heating company, and our experiments with the topic show promising results. Achieved accuracies are technologically useful and beneficial. We would recommend neural networks for local weather predictions for technological applications, and we want to emphasize the availability of local weather prediction as an alternative and extension of regional weather forecasts.

From the company feedback, neural networks prediction is more useful then standardized daily temperature profiles for temperature modelling. It is mostly because of better handling of shifts of air temperature behavior during the year. In company implementation, output from predictors required additional post-processing for visualization purposes. Especially, the connection, or merging, of several predictions may require special handling.

Due to lack of historical data from weather forecasts which we had, prediction without considering forecasts outperformed the one with forecasts. But, we expect this to change with more data available in future. In our experiments, more weather variables we used, better accuracy we obtained. Even if we skipped some technology signals, our prediction accuracies dropped (if only a little).

Acknowledgements. This research is supported by the "Center of Competence of knowledge technologies for product system innovation in industry and service", with ITMS project number: 26220220155 for years 20012-2015 and by the National Research and Development Project Grant 1/0667/12 "Incremental Learning Methods for Intelligent Systems" 2012-2015.

References

1. Dostál, P., Chramcov, B., Baláti, J.: Prediction of the heat supply daily diagram via artificial neural network. In: Proceedings of the 4th International Carpathian Control Conference, ICCC 2003 (2003)
2. Haykin, S.: Neural Networks (A Comprehensive Foundation). Macmillan (1994)
3. Kalnay, E.: Historical Overview of numerical weather prediction. In: Atmospheric Modeling, Data Assimilation and Predictability. Cambridge University Press (2003)
4. Koščak, J., Jakša, R., Sinčák, P.: Stochastic weight update in the backpropagation algorithm on feed-forward neural networks. In: The 2010 International Joint Conference on Neural Networks (IJCNN), pp. 3828–3831 (2010)
5. Koščak, J., Jakša, R., Sinčák, P.: Prediction of temperature daily profile by stochastic update of backpropagation through time algorithm. In: International Symposium on Forecasting, ISF 2011 (2011)
6. Koščák, J., Jakša, R., Sinčák, P.: Influence of number of neurons in time delay recurrent networks with stochastic weight update on backpropagation through time. In: Zelinka, I., Snasel, V., Rössler, O.E., Abraham, A., Corchado, E.S. (eds.) Nostradamus: Mod. Meth. of Prediction, Modeling. AISC, vol. 192, pp. 133–141. Springer, Heidelberg (2013)
7. Krose, B., van der Smagt, P.: An Introduction to Neural Networks. University of Amsterdam (1996)
8. Sinčák, P., Andrejková, G.: Neurónové siete I, II. Elfa Košice (1996)
9. Werbos, P.: The Roots of Backpropagation: From Ordered Derivation To Neural Networks and political forecasting. Wiley-Interscience, New York (1994)

Adaptive Classifier of Candlestick Formations for Prediction of Trends

Ján Vaščák, Peter Sinčák, and Karol Prešovský

Abstract. Candlestick charts have become in last decades a popular means in predicting trends on stock markets. Their properties enable in the form of the so-called formations to represent some symptoms of market changes in a user-friendly manner. Thus experienced businessmen are able using such kind of information to predict situations and in advance to correctly decide. However, candlestick charts are only one of many other indicators and their interpretation is not trivial. It depends e.g. on commodity, stage of a given trend, etc. To efficiently perform correct prediction using this graphical means a system utilizing ability to process vague information being able of adaptation is necessary. In this paper a design of such an fuzzy adaptive knowledge-based classification system using evolutionary optimization is proposed for categorization of characteristic candlestick formations and verified by a number of experiments in the area of exchange rates.

1 Introduction

Although candlestick charts were invented by Japanese rice traders in the 18th century they have been used only since the middle of the previous century in a greater measure [2]. Nowadays they belong to fundamental means of *technical analysis*, which tries to forecast prices using historical market data in general. The main premise of *technical analysis* is belief that history is an always repeating series of occasions. So it would be principally possible to predict them if we know symptoms that characterize their occurrence. Experience confirms that prices are influenced by various factors as economical, political, seasonal (weather, climatic changes) as well as psychological. Mutual acting of these factors leads to repeated increasing and

Ján Vaščák · Peter Sinčák · Karol Prešovský
Center for Intelligent Technologies, Technical University of Košice,
Letná 9, 042 00 Košice, Slovakia
e-mail: {jan.vascak,peter.sincak,karol.presovsky}@tuke.sk

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 373–383.
DOI: 10.1007/978-3-319-00542-3_38

descending trends of prices, which are known in the area of finances as *bullish* and *bearish* ones, respectively.

However, there are also opposing theories like *efficient-market hypothesis*, which state about unpredictability of stock market prices because they are principally random or based on information not publicly available [8]. Mainly the psychological factor influenced by different natures and priorities of businessmen as well as unexpected situations act against the existence of any formulas for price predictions. Nevertheless, in last years, with help of huge data processing the experience shows there are at least some general patterns of behaviour and *technical analysis* can be used on certain level of credibility and at least its outcomes can determine trends of price changes.

Trend becomes a key notion in price prediction because it represents environment of price changes and sentiment of businessmen [12]. Basically, as the *Dow theory* states, each trend has three stages: *accumulation, absorbtion* and *distribution*, respectively. A new trend starts when a relatively small group of 'clever' businessmen starts to buy or sell a commodity against the majority on the market. During this stage this minority's *accumulates* a certain mass of stock without any significant price changes. After that the majority of businessmen starts to copy the minority activities in a huge mass, in other words the stock is *absorbed*, which causes significant price change. Finally, the 'clever' minority starts to *distribute* the stocks to the market so the pressure on price change ceases and the trend turns its direction, which is its finish.

The situation is still more complicated because principally there are three trends, which timely overlap and differ mainly by their length. There are *major, secondary* and *minor* trends. Their simultaneous existence causes there are lots of local extremes, which can temporally change prices up to two thirds of those determined by the major trend. Besides, the are further influences as e.g. volume of transactions, which are needed to be taken into consideration. The above mentioned remarks only confirm the need of processing vague information and adaptivity ability to get acceptable (although not fully reliable) advices for behaviour on the market.

Charts are main means of data representation for needs of *technical analysis*. Beside their user-friendly representation they also show possible trends and changes very clearly. A special kind of them is the so-called candlestick chart, where we can observe various formations (patterns) characteristic for various situations on the market. Hence the prediction task can be converted into a pattern recognition problem [11]. Further, for a more realistic model description of the market price changes a notion of *fuzzy candlesticks* was defined [6]. These approaches lead to constructing knowledge-based classification systems with the use of fuzzy logic, e.g. as in [9]. However, setting up the knowledge base is done still manually, which could be a problem, because of need of an experienced expert as well as a steady adaptation of the knowledge base for new conditions or commodities. Therefore, we present an adaptive fuzzy expert system using evolutionary algorithms for evaluation and classification of candlestick formations giving to resulting statements a certain weight of reliability and hereby a possibility to filter out potential false signals.

Based on the featured problem the paper describes basic notions and formations of a candlestick chart in Sect. 2. Section 3 deals with the design of the formations classification system and adaptation of its parameters. Some experiments and resulting experience with prediction of price trends are discussed in Sect. 4. Finally, some concluding remarks are contained in the last Sect. 5.

2 Candlestick Charts and Typical Formations

Beside other charts used for depicting price trends the candlestick chart has become very popular because of its user-friendly statement ability, where not only temporal price development but also price changes with their extreme deviations are clearly depicted. Information recorded e.g. in a simple line chart is summarized during a given period into a *candle*. Thus we get a compressed depiction as a set of candles, in other words a 'candlestick', where some formations of candles are especially well visible (see Fig. 1).

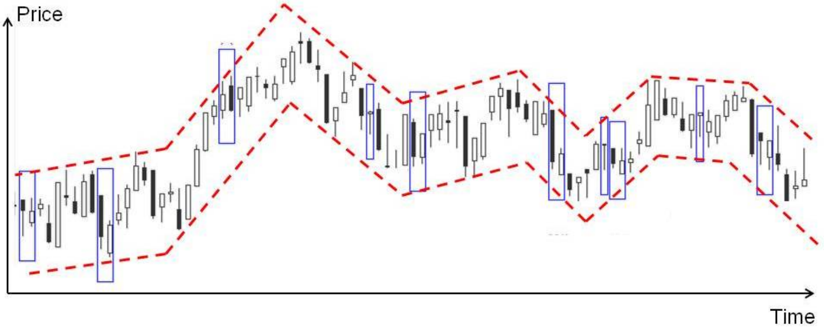


Fig. 1 An example of a candlestick chart with dashed trend lines and some selected typical formations indicated by rectangles

A candlestick chart is represented by white and black candles, which express increasing (*bullish*) and decreasing (*bearish*) price trends during one period (mostly one day), respectively. A candle depicts in total six values: trend (white or black), open (starting) price, close (finishing) price, body size (difference between open and close value), lowest price and highest price during the given period (see Fig. 2). The depictions of last two values are named as *shadows*, too.

Experience shows that some situations on the market may be followed by some characteristic formations of candles, which represent specific patterns having their special names. The first task in the analysis of a candlestick chart is to find these formations. Users try mainly to find trends, their tendency (*bullish* or *bearish*) and point of their reversal. This determines also the categorization of typical candlestick formations as they are depicted in Fig. 3. Sizes of bodies and shadows play also a significant role concerning the strength of such an indication. Besides, the white

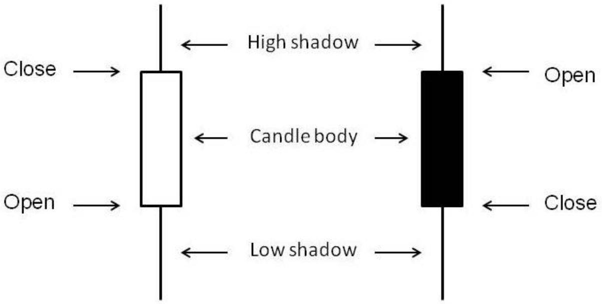


Fig. 2 Values represented by white and black candles

candle alone belongs also to a *bullish* trend as well as the black one to the *bearish* trend. The pattern *Cross* (in Japanese *Doji*) is very specific in comparison to others. The size of its body is zero or close to zero, i.e. the open and close values are approximately identical. The traverse line can be principally placed on any vertical position as it is indicated in the form of dashed lines in the figure. There are still further and more complex formations consisting of e.g. three candles. One representative of them is the formation *Three River Evening Star*, where it starts with a big white candle being followed by another smaller white one, whose open price is greater than the close price of the first candle. Thus a gap exists between these two candles, which is covered by the third but a black candle. More detailed information about further formations can be found e.g. in [13].

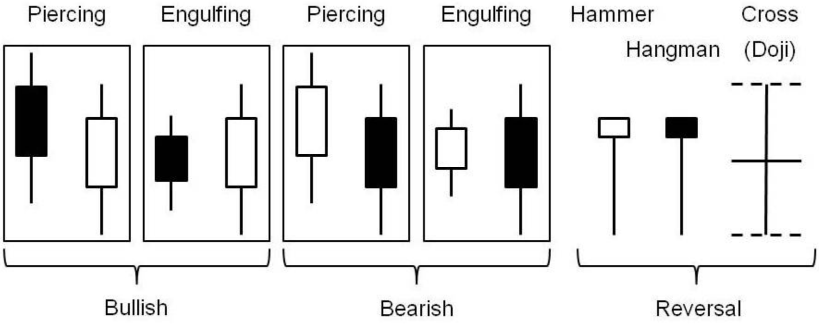


Fig. 3 Some typical candlestick formations (patterns) categorized by the type of trend

However, the existence of a given formation is not yet an unambiguous indicator that a situation will occur. Not situations but formations are consequences and produced indicators of situations. A reverse relation from a formation to a situation may not be valid. It seems that occurrence of a formation is only a partial indicator being connected only with a few parameters, which are indeed a part of significant changes but only these parameters do not have a greater influence. From this reason candlestick formations can act only as auxiliary indicators and it is necessary to combine them with other indicators as e.g. the state of a trend, whether a formation occurs in a highly developed trend or only in a transition phase. Therefore, after identification of a formation the second task is necessary, which is assigning a weight of reliability to such a formation, to filter out false signals, which occur among these formations relatively often. In our approach we try to describe an identified formation by a set of fuzzy production rules and to asses the weight, which is related with the probability of situation occurrence. In addition, we use an evolutionary algorithm for fine tuning descriptive rules of formations.

3 Design of a Weighting Classification System of Candlestick Formations

The description of candlestick formations is based only on experience of traders, which can be described in the form of rules. For instance, the *Bullish Engulfing* is described like:

IF the first candle is black and the second candle is white and the close value
of the first candle is greater than the open value of the second candle
and the open value of the first candle is less than the close value
of the second candle and the body size of the second candle is big
and shadows of the second candle are small and the trend is developed
and decreasing THEN it is a Bullish Engulfing formation. (1)

Thus we can obtain a rule base describing all significant formations. It is apparent such a task would be especially convenient for a knowledge-based system. However, we can see there are vague (fuzzy) values as *big, small*, etc. as well as crisp values as colour of candles. Therefore, we do a fictive fuzzification of crisp values in the form of singletons to be able to mix fuzzy and non-fuzzy values.

Another and more serious problems is (as already mentioned) that the only identification of formations is insufficient — the related situation may but also needs not occur. The final goals of the classification system are the *reliability weights* of found formations, which correspond to the *occurrence probability* of related situations, i.e. price increase or decrease and reversal points of trends. If we look at the inference process of a fuzzy system the result of premise evaluation is the so-called *rule strength* $\alpha \in [0;1]$, which represents the measure of rule fulfillment. Hence α is equivalent to the probability of the situation occurrence and hereby in this case the computation is finished. In other words, we evaluate premises of all rules describing significant candlestick formations. We get a vector of their strengths, i.e. occurrence probabilities telling us, which situations are the most probable in next future. The inference process of a fuzzy system is not a matter of this paper. There is a number of sources dealing with this topic, like for instance [4]. Similarly, there are numbers of methods how a trend can be evaluated. We used a method described in [3].

The rules contain a series of linguistic values described by fuzzy sets or more concretely by their membership functions. For the sake of correctness we should separate weights of individual inputs to rules from membership functions as e.g. body and shadow size, fuzzified differences between open and close values, etc. It is apparent these inputs have different influence on the indication of a formation. However, in reality we can simplify the computation if we do scaling of membership functions by these weights. The membership functions become indeed distorted but we get the same results. If we use some kind of automatic design of membership functions it will not be for us more interesting how ’pure’ membership functions and ’pure’ weights look. This task belongs to the area of adaptive systems and we choose a simple evolutionary algorithm. Using such a mix of these two principally distinct groups of parameters simplifies us the final optimized solution resulting in ’distorted’ membership functions.

3.1 Adaptation of Classification System Parameters

In the literature there are known many adaptation methods of fuzzy systems, which utilize basically either indirect methods, e.g. using neural networks [7] or direct adaptation methods and among them evolutionary algorithms [1]. A very interesting part of direct adaptation is also *fuzzy interpolation*, which is suitable in the case of a small number of rules or training data [5]. As our goal is to achieve the most reliable values of probabilities for designed rules, i.e. the computed *occurrence probability* of a situation if related formation is detected should be equivalent or at least approximate to the real probability so the task falls into the area of optimization. Further, as price trends are very dynamic processes we will need to take into consideration some special criteria, which are employed mainly in defining the fitness function [10].

As individual formations are mutually independent and not influencing we will optimise each rule separately and consecutively create only one rule base. From this point of view our optimization algorithm can be classified as a *Michigan-type* algorithm [1]. Besides, it means rules are individuals in this case, whose parameters are divided into groups of parameters describing membership functions of rule inputs. In one rule r_i $(i = 1,\dots,n)$ for each input only one membership function is created. In other words, the number of membership functions is equivalent to the number of rule inputs m_i. If there are necessary p parameters for description of a selected membership function then an individual representing the rule r_i will be of the length $m_i.p$. In our case we use the membership function in the form of a trapezoid, whose parameters are a, b, c, d, h as shown in Fig. 4.

Operators of mutation and crossover were modified in contrast to conventional definitions. Each parameter of the individual is modified by mutation with a certain probability. As the parameters are grouped into membership functions in an individual so the crossover is processed between these groups and not between individual parameters (genes), i.e. whole membership functions are randomly interchanged.

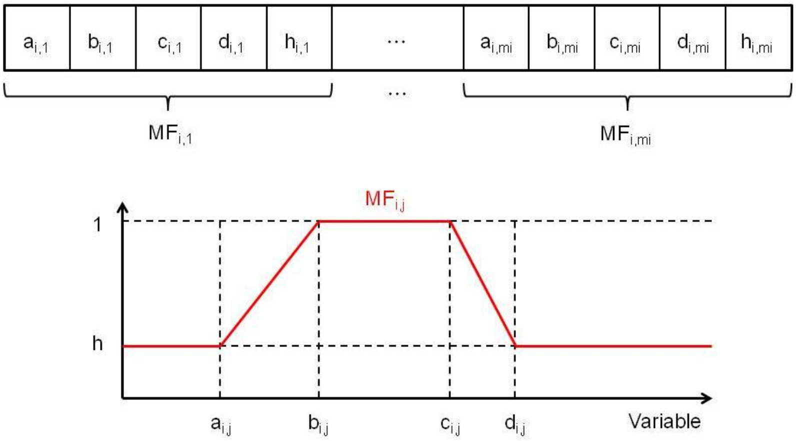


Fig. 4 Structure of the i-th chromosome representing the rule r_i with m_i inputs and structure of parameters for the j-th membership function $MF_{i,j}$ $(j = 1,\dots,m_i)$ in the rule r_i

From the final population of competing rules for a given formation the best one is selected and added to the resulting rule base of the fuzzy classifier.

The fitness function should depend on the difference between a given probability P (in the calculation process it is equivalent to α) of occurrence for the relation formation — situation and the real one P' obtained from data, i.e. calculating successful occurrences if a typical formation was detected. The smaller the difference is the better the fitness value for a given individual is achieved. For this purpose probability histograms are used, which are charts divided into bars representing given (reference) probabilities (equivalents to the mentioned notions are weights or reliability of a given rule). In our case there are 10 bars divided by 5% in the range 50-100%. The sizes of bars represent real measure of successfulness of the checked rule, where for each i-th bar ($i = 1,\dots,10$) with N_i detected formations its given probability P_i is compared to the real probability P'_i. In such a manner we can define the measure *Accuracy of Prediction* as:

$$AP = 1 - \frac{\sum_i N_i.|P_i - P'_i|}{\sum_i N_i} \quad . \tag{2}$$

The fitness value is influenced still by these other 5 factors:

- *Successfulness of Prediction* is the rate of successfully predicted price changes for all formations in histogram.
- *Weight Range* tries to prevent a tendency of membership functions to achieve their ranges analogous to the average successfulness of a formation. It is computed as an average between the minimum and maximum values of the grade of membership. The result is that narrow membership functions are suppressed.

- *Distribution in Probability Histogram* prevents creating membership functions of jump-wise forms. It prefers individuals, which occupy more bars in the probability histogram. This factor has mostly a small value, which can be calculated as:

$$DPH = \frac{1}{\max(N_i)} \quad \text{for } \forall i \ . \tag{3}$$

- *Number of Recognized Formations* prevents preferring individuals, which have a few very successful predictions but they are not able to detect further formations. This factor is set manually and it should not to be too high because we get individuals detecting many formations but with poor reliability. This factor is a counterpart to *Successfulness of Prediction*.
- *Best Prediction* motivates individuals to separate more successful formations from mean ones. It gives bonus for the best achieved prediction although with a smaller number of formations. It is set manually and its value is low.

All of these factors are in the range $[0;1]$ and their weighted product gives the final fitness value (also in $[0;1]$). For next generation one half of the best old and one half of the best new individuals are selected. The optimization process is finished when after a certain number of generations the fitness value is not enhanced by a minimum increment.

4 Experiments and Discussion

Three basic kinds of experiments using exchange rates of several currencies were performed to attest reliability of obtained results from our classification system of candlestick formations. Namely, we have mutually compared U.S. dollar ($), Japanese yen (¥) and Swiss franc (SF) based on four-years long historical data. We verified the reliability of three selected formations: *Engulfing, Cross* (both forms, i.e. *bullish* and *bearish*) and *Three River Evening Star*. The obtained results are summarized in Table 1, where beside the number of detected formations also the three most significant factors influencing the fitness value are listed in percentage (see Sect. 3.1), i.e. *Accuracy of Prediction*, *Successfulness of Prediction* and *Best Prediction*.

The goal of the first experiment No. 1 is to recognise the quality of the adaptation mechanism. The classification uses two kinds of membership functions definitions, namely the originally ones proposed by an expert, i.e. *before* and the adapted ones, i.e. *after* adaptation, as presented in identical columns for individual exchange rates, where e.g. in the table a notation €–$ means exchange rate between € and $. The experiment No. 2 resembles to conventional experiments in learning. The data were divided into training and testing parts by the ratio 3:1 (see columns *training* and *test* in the table). The aim is to investigate the sensitivity of classification parameters in the same commodity if they are adjusted only on training data. The number of detected formations does not play any role. The last experiment No. 3 is similar to the previous one, where the commodity is changed, i.e. parameters are adjusted on one exchange rate and tested on another rate. The goal is to analyze specificity of parameters. For the sake of simplicity the notation $–¥→$–SF means training on $–¥ and testing on $–SF. Again, the number of detected formations is irrelevant.

Table 1 Summary of experimental results (No. 1-3) for various combinations of exchange rates for U.S. dollar ($), Japanese yen (¥) and Swiss franc (SF). Symbols E, C, 3*, ↑ and ↓ stay for *Engulfing, Cross, Three River Evening Star, bullish* and *bearish*, respectively.

			Number [pcs]		**Accuracy [%]**		**Successfulness [%]**		**Best prediction [%]**	
No.	**Data**	**Form.**	**before**	**after**	**before**	**after**	**before**	**after**	**before**	**after**
1	€–$	C↓	0	195	0	97	0	73	0	100
		C↑	50	167	72	96	84	79	100	87
		E↓	83	183	70	98	72	75	72	80
		E↑	16	11	84	96	62	72	75	72
		3*	4	81	100	99	50	80	50	100
	$–SF	C↓	0	269	0	95	0	76	0	100
		C↑	50	229	61	93	72	75	100	85
		E↓	109	194	74	96	69	67	69	90
		E↑	36	36	69	98	75	75	76	75
		3*	4	107	100	99	50	69	50	100
			–	**–**	**training**	**test**	**training**	**test**	**training**	**test**
2	€–$	C↓	–	–	97	72	73	64	100	100
		C↑	–	–	96	74	79	84	87	100
	$–¥	E↓	–	–	98	88	74	82	100	100
		E↑	–	–	98	90	79	77	85	81
		3*	–	–	95	66	74	90	80	100
	$–SF	3*	–	–	99	87	69	74	100	100
			–	**–**	**training**	**test**	**training**	**test**	**training**	**test**
3	$–¥ ↓ $–SF	C↓	–	–	95	71	76	76	100	76
		C↑	–	–	93	90	75	74	85	76
		E↓	–	–	96	74	67	69	90	69
		E↑	–	–	98	81	75	75	75	90
		3*	–	–	99	82	69	66	100	100
	$–SF ↓ $–¥	C↓	–	–	93	92	74	74	78	88
		C↑	–	–	97	76	75	75	83	100
		E↓	–	–	98	82	74	73	100	100
		E↑	–	–	98	40	79	60	85	100
		3*	–	–	95	90	74	67	80	74

The results in the experiment No. 1 show that using an evolutionary algorithm for adaptation of membership functions brought significant improvement in all investigated indicators. Especially, a high improvement was achieved in detection of formations, which is a surprise that a human oversees so many parts of information. The other two experiments show decrease of classification quality, where 15% is no exception. This fact is a confirmation of earlier experience published in many sources that indicators contained in candlestick charts lack on robustness.

Summarizing obtained results we can see that the occurrence of a formation is able to predict price changes. However, the success rate of such predictions is approximately 75 - 85%. Therefore, this approach can be used only as an auxiliary means. Its exclusive use is insufficient. Main contribution of this approach is its ability to predict trends in the form of a decrease or increase and reversal point as well as to offer the *reliability weight* or *occurrence probability* of a situation if a certain formation is detected. However, candlestick charts are to weak for quantitative determining how strong a trend will be. This is the limit of its use.

5 Conclusions

In this paper we tried to show advantages of evolutionary optimization for description of typical candlestick chart formations, which can indicate some significant trend changes. Especially, we focused on tendencies and points of their reversal. The performed experiments showed that adjusted rules by such an evolutionary algorithm achieved considerably better results than those proposed by an expert.

However, signals generated by the candlestick formations can be still considered as only hints helping to make better decisions. For successful trading one has to consider much more factors. Conventional trading systems take into account only presence of a formation. Many of them offer possibility to set up shapes of candlestick formations thus a user can configure the system. Fuzzy logic is often used for this purpose. Such an approach assumes that a businessman has experience and knowledge of his own. Our adaptive fuzzy expert system brings advantage of collecting trading experience on historical prices. So when a formation appears on the candlestick chart, it assigns weight to this signal so a businessman knows how serious it is. Descriptions of formations in technical literature lack this kind of information. However, in spite of that this information can be considered only as an advice and cannot be used instead of a whole trade analysis because candlestick formations are results of local dynamics only, which could partially explain their high sensitivity to changes in time series.

Acknowledgements. This research is supported by the “Center of Competence of knowledge technologies for product system innovation in industry and service”, with ITMS project number: 26220220155 for years 20012-2015 and by the National Research and Development Project Grant 1/0667/12 “Incremental Learning Methods for Intelligent Systems” 2012-2015.

References

1. Cordón, O., Herrera, F., Hoffmann, F., Magdalena, L.: Genetic Fuzzy Systems — Evolutionary Tuning and Learning of Fuzzy Knowledge Bases. Advances in Fuzzy Systems — Applications and Theory, vol. 19. World Scientific (2001)
2. Cornelius, L.: Technical Analysis Application in the Global Currency Markets. Prentice-Hall, New York (2000)

3. Gavalec, M., Mls, K.: Trend evaluation in on-line decision making. In: Proc. International Conference on Operational Research, KOI 2008, Pula, Croatia, pp. 267–274 (2008)
4. Jantzen, J.: Foundations of Fuzzy Control. John Wiley & Sons, Inc., New York (2007)
5. Johanyák, Z.C., Kovács, S.: A brief survey and comparison on various interpolation-based fuzzy reasoning methods. Acta Polytechnica Hungarica 3(1), 91–105 (2006)
6. Lee, C.H.L., Liu, A., Chen, W.S.: Pattern discovery of fuzzy time series for financial prediction. IEEE Transactions on Knowledge and Data Engineering 18(5), 613–625 (2006), doi:10.1109/TKDE.2006.80
7. Lin, C.T., Lee, C.S.G.: Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems. Prentice-Hall PTR, New Jersey (1996)
8. Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.V.: Algorithmic Game Theory. Cambridge University Press, New York (2007)
9. Olej, V., Křupka, J.: Prediction of gross domestic product development by Takagi-Sugeno fuzzy inference systems. In: Kwasnicka, H., Paprzycki, M. (eds.) Proc. of the 5th International Conference on Intelligent Systems Design and Applications (ISDA), pp. 186–191 (2005)
10. Precup, R.E., Preitl, S.: Optimisation criteria in development of fuzzy controllers with dynamics. Engineering Applications of Artificial Intelligence 17(6), 661–674 (2004)
11. Roy, P., Sharma, S., Kowar, M.: Fuzzy candlestick approach to trade S&P CNX NIFTY 50 index using engulfing patterns. International Journal of Hybrid Information Technology 5(3), 57–66 (2012)
12. Schannep, J.: Dow Theory for the 21st Century: Technical Indicators for Improving Your Investment Results. John Wiley & Sons, Inc. (2008)
13. Teweles, R., Bradley, E.: The Stock Market, 7th edn. John Wiley & Sons, Inc., New York (1998)

Identification of Economic Agglomerations by Means of Accounting Data from ERP Systems of Business Entities

Petr Hanzal and Ivana Faltová Leitmanová

Abstract. The aim of this paper is to introduce the methodology of the identification of common economic agglomerations based on the accounting data gained from business entities in the Czech Republic. The identification of economic agglomerations by descriptive approach should reveal the geographical location and their extent within the country. Although the result of the examination cannot be fully generalised due to a limited data source, it should be considered as the contribution to regional studies of the Czech Republic in the broadest context.

Keywords: Economic agglomerations, Geostatistical methods, Inverse distance weighted interpolation, Enterprise Resource Planning.

1 Introduction

This article is intended as a contribution to the theory of ERP systems of business entities and their application to regional sciences as a result of a several-year research. The main goal is to demonstrate the potential of using accounting data from ERP systems for regional evaluation, e.g. for identification of economic agglomerations through a complex of accounting data obtained from business entities.The contribution is divided into two parts – a theoretical framework and a practical part. The theoretical framework contains theoretical grounds for application of accounting data from ERP to regional evaluation. It also contains a definition of economic agglomerations and the methodology for their identification. There is a further explanation of the statistical methods subsequently applied in the practical part there. The practical part then deals with identification of

Petr Hanzal
The Institute of Technology and Businesses in České Budějovice, Czech Republic
e-mail: hanzal@mail.vstecb.cz

Ivana Faltová Leitmanová
University of South Bohemia, České Budějovice, Czech Republic
e-mail: leitman@ef.jcu.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 385–399.
DOI: 10.1007/978-3-319-00542-3_39

economic agglomerations on an example of 2010 accounting data from 27 randomly selected business entities with countrywide operation in the Czech Republic regardless business subject classification.

2 Theoretical Framework

2.1 ERP Accounting Data

Accounting data represent an essential element of each ERP system. They bear the recorded facts related to company activities and are transferrable, interpretable and can be processed. They include all the facts of the organisation's micro and macro environment like recorded data of economic facts and other factors affecting the chain of the company value chain [1]. The basic function of accounting is to provide all its users with reliable information on the company economic situation. Accounting is particularly required to provide information on company assets and financial situation in the form of a balance sheet and performance in the form of a profit and loss statement within a particular period. Assessment of how the company management valorised the assets put in their trust in the previous period is not the only reason, there is still stronger interest in the financial situation prognosis as to whether and to what extent the company will be able to achieve positive results in the future. Accounting information is intended for both managers and various external users that are interested in the company for various reasons. We can thus differentiate between two basic groups of accounting information users [2]:

- Entities directly involved in company financing (company owners and creditors – e.g. bond holders, banks, suppliers);
- Entities that are somehow interested in the company financial results (revenue offices, state administration, employees, competitors, potential investors, the public).

Financial accounting data are formed by the facts in the field of asset situations, receivables from customers, debts to suppliers, and they also monitor incomes and costs from the point of view of the accounting unit as a whole. They include all transactions in the field of purchasing and selling goods, material, own products and services, including the stem data of suppliers and customers [3].

2.2 Definition of Economic Agglomerations

Economists' interest in activities and mutual transactions of companies, also from the point of view of locations where the companies are based, has been obvious for several decades. Allocation of resources and selection of a suitable place for a particular activity is a strategic question for each business entity [4]. Individual authors approach its solution with accent to different factors – foreign investments

[5], social capital [6], cluster evolution [7], institutional influences [8], urbanisation aspects [9], etc. Some alternative in monitoring performance at the regional level rather than the usual approach, using national, regional accounts, but using data "regional bodies" (in the sense of regional jurisdiction with regard to the seat of these entities) based on an analysis of direct (immediate) and mediated the relationship performance of these operators and power consumption along with selected segments of value added - personnel expenses and depreciation. Listed examining engaged [10], who, in addition to incorporate the sectoral differences investigated region and the economy as a whole.The basic concept of the economy of agglomerations presumes that the clustering of economic activities is caused by companies relying on some advantages brought to them by their physical closeness to other companies, upon their experience. A broader definition of agglomeration economies says that clustering occurs in such economies where a company may have an advantage from being at the same location as the other companies. Four basic sources supporting agglomeration are known [11]:

- Internal increase of economies of scale. This may occur to one company that manages to reduce costs as a result of controlling a bigger market. This concept involves no spatial effect except for the fact that one company creates a large local concentration of production factors;
- External economies, which are available to all local companies in the same industry – localisation economies;
- External economies, which are available to all local companies regardless the industry, growing from urban scale and density – urbanising economies;
- External economies, which are available to all local companies, separated from different sectors – Jacobs's externalities.

Economies with high agglomeration rates are those external economies a company localised at the same place as more companies can benefit from them. Agglomeration is traditionally supported by the two following mechanisms. At first it is a reduction of operation costs with the possibility to share certain social and physical infrastructure resources, and secondly it is the reduction of transport cost as a result of extended interaction between suppliers and customers that are located next to each other. Presence in an agglomeration may be expressed in different words as the possibility to increase company performance reducing the costs of sharing fixed and current capital. The importance of economic agglomerations is expected to rise as they reduce costs or increase yields (or both) to the companies participating in them. Basically there are two types of economies with high agglomeration rates distinguished: urbanised and localised [11], [12]. The term urbanised economy is linked to general economies, regional and urbanistic concentration of all companies and industries at a single location. These are the forces that lead to forming industrial “cores” and metropolitan regions. Localised economies are specific by their relation to companies involved in mutually linked activities leading to the formation of spatial agglomerations – industrial regions, localised industrial clusters, etc. [14]. Company agglomeration theories consider the internal increase of economies of scale and the size of internal and external markets as the essential factor explaining company agglomeration. Economy of scale is the basic component of all models emphasising the role of diversity of outputs and inputs. Companies having the possibility for a higher degree of economies of scale are looking for functional regions with high market potential where they can exploit their advantage. Some types of goods and services are connected with high geographic transaction costs, which determine whether production in the region will or will not be profitable. This is why products should be classified with regard to their sensitivity in relation to transaction costs. Specific product categories with development potential in small or medium or large functional regions may be defined based on this approach [15].The base of agglomeration activities includes numerous activities in the value added chain that are performed in a particular region. Economic activity of the particular region is in relation to the value and number of business transactions in the goods and services market undertaken with entities inside as well as outside the region. Concentration of economic activities to selected regions is derived from the impact of market forces that influence formation of economic agglomerations [16].

2.3 Dimension of Regional Agglomerations

With the large number of regional agglomerations a problem with their definition arises. Similar terminologies are used for agglomerations with widely different characteristics [16]:

- The geographic area of an agglomeration corresponds to the territorial extent of companies, customers, suppliers and supporting services and institutions that are involved in the existing mutual links and activities characterising the cluster;
- Agglomeration density corresponds to the number and economic weights (in the units of market shares of the specific industry) of companies within the agglomeration;
- The agglomeration activity base includes the quantity of activities in the value added chain that are performed in the region;
- The geographic size of sales provides indication of agglomeration scope;
- The agglomeration development stage may be embryonic, emerging or mature and agglomeration may be growing, stagnating or vanishing.

3 Methodology of Work

The main aim of the work is to demonstrate the possibility to use accounting data from company information systems in regional economy and statistics, e.g. for identification of economic agglomerations by means of a set of accounting data obtained from randomly selected business entities with countrywide operation. These partial goals were defined for fulfilment of the main goal:

- Definition of an indicator of intensity of economic relations based on business cases of business entities, which will be used in the next part of the work;
- Demonstration of the possibility to identify economic agglomerations upon ERP accounting data from selected business entities.

If we take into account that the geographic area of agglomeration corresponds to the territorial extent of companies, customers, suppliers and supporting services and institutions that are involved in the existing mutual links and activities characterising the cluster, while agglomeration density corresponds to the number and economic weights of companies within the agglomeration meaning that the identification of an economic agglomeration may not only be performed by increased economic activity at a particular location, but also by the higher density of the number of companies based within the agglomeration. It is thus possible to identify locations with increased occurrence of business transactions and thus localise economic agglomerations upon accounting operations describing business cases.

3.1 Spatial Interpolation by Means of the Method of Inverse Distance Weighted

Phenomena that are closer to each other in space tend to be more similar than those that are more distant in space [17]. This is the basic geostatistical principle. As the distance from the prediction point grows the influence of the location on the prediction decreases, and from a certain distance the influence of these distant places on the prediction at a given point is zero. Spatial analysis of data is based on application of measurement of particular phenomenon at particular locations and consequent estimation of that phenomenon within the whole area. This may be achieved by means of interpolation. The value of a phenomenon at all points of an area is thus determined from values measured at selected points in spatial analysis. Each point then has a value determined either by measurement or estimation. Interpolation by the Inverse distance weighted (IDW) method explicitly implements the presumption that objects that are closer to each other are more similar than those located at longer distances. IDW uses measured values surrounding the given location for prediction of values for any measured location. The measured values closest to the measured location have a stronger influence on the predicted values than distant values. IDW presumes that each measured point has influence, which weakens with growing distance. This gives larger weight to points that are closer to the measured location, and this weight decreases with the growing distance from the measured location. This is why the method is called inverse distance weighting.

As mentioned above weights decrease proportionally to the distance from the measured location.

The IDW method determines the value in each point by a weighted linear combination of the values measured at several of the nearest surrounding points according to the formula:

$$Z(S_0)\frac{\sum_{i=1}^{n}\frac{z\,(s_i)}{d_{oi}^{p}}}{\sum_{i=1}^{n}\frac{1}{d_{oi}^{p}}} \tag{1}$$

where $Z(S_0)$ is the value of the interpolated point, $Z(S_i)$ is the measured value in the i-th point, d_{0i} is the distance between the interpolated point and the measured point, p is the weight power parameter.

The weight power parameter p (power), expresses the relative weight declination slope. If p = 0, the relative weight has constant value, if p>0, only the immediately adjacent points influence the measured location – see Figure 1.

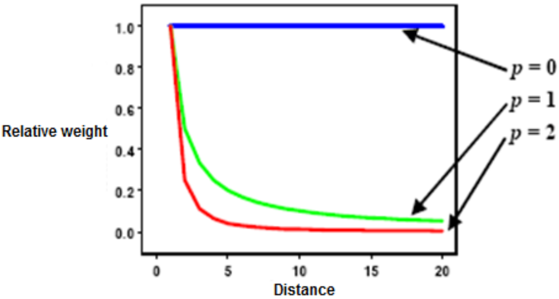


Fig. 1 Influence of *p* on relative weight declination slope (source: ArcGIS program documentation)

The optimum value of parameter p may be determined by means of the minimum value of the Root Mean Square Prediction Error (RMSPE), however p = 2 is usually used. The number of neighbours to be taken into account, from which the unmeasured value is to be calculated, is another parameter involved in the calculation. Graphic input of the method may be a three-dimensional model, where the latitude and longitude are the first two dimensions and the measured quantity e.g. intensity of economic relations based on business cases of business entities is the third dimension.

3.2 Methods Applied to the Research of Economic Agglomerations

After definition of economic agglomerations an indicator of intensity of economic relations based on business cases of business entities was chosen, which characterises the situation in regions from the viewpoint of purchases and sales of goods, own products and services transacted in the monitored regions and from the viewpoint of the number of active business entities in the particular region, from a sample of accounting data from 27 randomly selected business entities with countrywide operation.

Economic agglomerations were localised by means of a coordinated approach where the above indicator of intensity of economic relations based on business cases of business entities was calculated for each point, and the points were subsequently subject to research by means of statistic methods leading to identification of agglomerations.

Normalisation of the Calculated Data

Before application of all the methods a correction of values of selected quantities was performed. The selected quantities are actually expressed in incomparable values, which is why their normalisation had to be performed first. The following formula was applied to the transformation of all variables to normalised quantities:

$$\frac{x_i - \bar{x}}{S} \tag{2}$$

where x_i is the calculated quantity before normalisation, $\bar{x}$ is arithmetic average, S is standard deviation.

Indicator of Intensity of Economic Relations Based on Business Cases of Business Entities

The indicator of intensity of economic relations based on business cases of business entities I_r of a location, as hereinafter applied was defined as a sum of normalised values of partial intensities:

$$I_r = + I_{hnz_r} + I_{pnz_r} + I_{hns_r} + I_{pns_r} + I_{pas_r} + I_{hpz_r} + I_{ppz_r} + I_{hps_r} + I_{pps_r} \tag{3}$$

The definition itself, calculation and data sources of the individual partial intensities are summarised in Table 1, while the meanings of the individual variables are as follows:

Y_i – number of employees in a business entity i, where i = 1…n.
Z_r – number of inhabitants in region r, to which sale are directed or where a purchase is performed, where r = 1… number of regions.
n – number of business entities (here 27).

Table 1 Definitions, calculation and data sources of the individual items (internal resource)

Partial intensity	Mathematic definition	Description of variables	Data filter criteria	Data source
I_{hpz_r} Partial intensity of the values of sales of goods & own products	$\frac{\sum_{i=1}^{n}\frac{B_{ri}}{Y_i}}{Z_r}$	B_{ri} – total value of all sales of goods & own products to target region *r* by business entity *i*	Synthetic account 604 for goods 601 for own products	Own calculation

Table 1 (*continued*)

$\boldsymbol{I_{ppz_r}}$ Partial intensity of the number of sales of goods & own products	$\boldsymbol{\frac{\sum_{i=1}^{n}\frac{C_{ri}}{Y_i}}{Z_r}}$	$\boldsymbol{C_{ri}}$ – total number of all sales of goods & own products to target region *r* by business entity *i*	Synthetic account 604 for goods 601 for own products	Own calculation
I_{hps_r} Partial intensity of the values of sales of services	$\frac{\sum_{i=1}^{n}\frac{D_{ri}}{Y_i}}{Z_r}$	D_{ri} – total value of all sales of services to target region *r* by business entity *i*	Synthetic account 602 for services	Own calculation
I_{pps_r} Partial intensity of the number of sales of services	$\frac{\sum_{i=1}^{n}\frac{E_{ri}}{Y_i}}{Z_r}$	E_{ri} – total number of all sales of services to target region *r* by business entity *i*	Synthetic account 602 for services	Own calculation
I_{hnz_r} Partial intensity of the values of purchases of goods & material	$\frac{\sum_{i=1}^{n}\frac{F_{ri}}{Y_i}}{Z_r}$	F_{ri} – total value of all purchases of goods & material from source region *r* by business entity *i*	Synthetic account 131 for goods 111 for material	Own calculation
I_{pnz_r} Partial intensity of the number of purchases of goods & material	$\frac{\sum_{i=1}^{n}\frac{G_{ri}}{Y_i}}{Z_r}$	F_{ri} – total number of all purchases of goods & material from source region *r* by business entity *i*	Synthetic account 131 for goods 111 for material	Own calculation
I_{hns_r} Partial intensity of the values of purchases of services	$\frac{\sum_{i=1}^{n}\frac{H_{ri}}{Y_i}}{Z_r}$	H_{ri} – total value of purchases of services from source region *r* by business entity *i*	Synthetic account 518 for services	Own calculation
I_{pns_r} Partial intensity of the number of purchases of services	$\frac{\sum_{i=1}^{n}\frac{I_{ri}}{Y_i}}{Z_r}$	I_{ri} – total value of sales of services from source region *r* by business entity *i*	Synthetic account 518 for services	Own calculation
I_{pas_r} Partial intensity of the number of active business entities	$\frac{\frac{K_{ri}}{Y_i}}{Z_r}$	$K_{ri} -$ number of customers/suppliers in region *r*, with at least one business case in relation to business entity *i*	Purchase or sale value <> 0	Own calculation

4 Research Input Data Sources

4.1 Data Model Applied to the Research

Figure 2 shows a schematic model of the data basis designed for regional evaluation, including relations between the databases. The individual databases correspond with the sources listed in the next part of the chapter.

Database of Business Entities

A set of 27 randomly selected unnamed monitored companies from the Czech Republic with countrywide operation regardless the business subject (NACE) is the primary data source. All the involved business entities were uniregional, i.e. all their units were located in the same region as the administration unit.

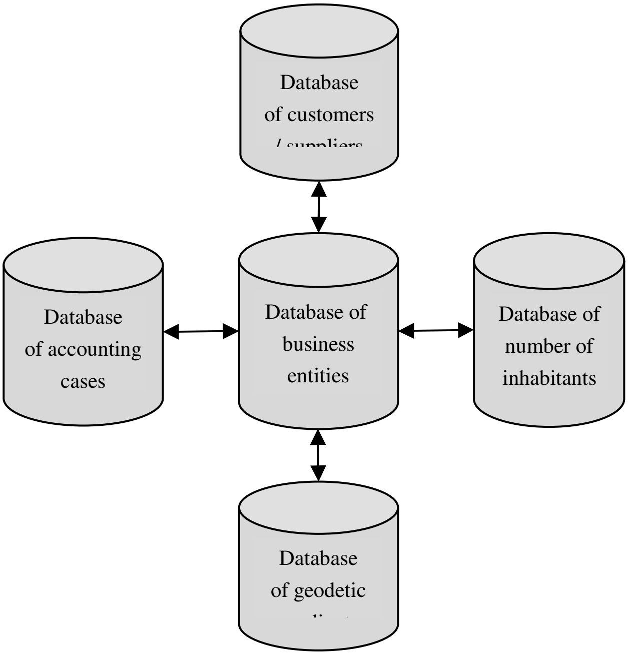


Fig. 2 Model of the data basis designed for evaluation with relations between the databases (internal resource)

Database of Accounting Cases

Consolidated accounting file for 2010, containing 4.35 million sentences, is the key data source for the research. Each accounting case includes the following information:

Account number, CREDIT and DEBIT sides, date and number of a document, accounting period, customer/supplier number, accounting text information, document symbol and a code specifying to which of the 27 monitored business entities each transaction belongs.

Supplier accounting cases may be chosen easily by means of the CREDIT side of synthetic account 321 – trade payebles.

The following prevailing main types of transactions may be distinguished by synthetic classification of accounts on the DEBIT side:

111 – cost of material (or also 112),
131 – cost of merchandise (or also 132),
518 – rendered services.

As all the above business entities apply a method on booking store operations, the accounting cases where the purchase of goods is booked to accounts 504/321 may be practically excluded.

Customer accounting cases may be chosen by means of the DEBIT side of synthetic account 311 – trade receivables.

The following prevailing primary types of transactions may be distinguished by accounting synthesis of the given operation on the CREDIT side:

601 – manufactured goods revenue,
602 – service revenue,
604 – merchandise revenue.

Database of Geodetic Coordinates, NUTS, LAU

The database of all communities in the CR with 15,500 records containing information on LAU1, geographic latitude and longitude, is another source of high importance.

Database of Customers and Suppliers

A data file of 27,900 items of legal entities and individuals, each with the number and address of the customer/supplier, country and post code, was gained by combining lists of customers and suppliers of all the 27 monitored business entities.

5 Research Results

Identification of economic agglomerations was performed by means of the ArcGIS program applying the Inverse distance weighted (IDW) interpolation method.

The grade of weight power parameter p was chosen as 1.8 and a maximum 15 of the nearest neighbouring points were taken into account. The indicator of intensity of economic relations based on business cases of business entities defined in Chapter 3.2 was used for calculation in the first case. 2,673 points characterised by latitude and longitude coordinates and by intensity of economic relations based on business cases of business entities were obtained in the Czech Republic in total.

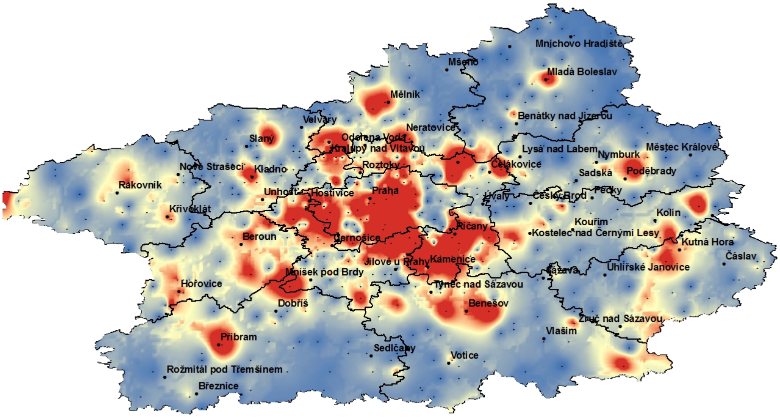


Fig. 3 Economic agglomerations in Prague and Central Bohemia (internal resource, data of selected entities)

In the event that the economy was made up of only these 27 randomly selected subjects, it would be possible according to their economic activities, such as those identified agglomerations: Prague with the nearest environs represents one large economic cluster covering the districts Capital Prague, Prague-East and Prague-West, see Figure 3.

6 Evaluation of the Research Results

Several conclusions may be drawn from the research results mentioned in the previous chapter.

One dominant cluster is obvious in Figure 3, represented in the CR by Prague and its environs surrounded with numerous smaller clusters, The method used allows to define not only large and traditional economic agglomeration, but also other smaller regional agglomeration, which may be classified as regional clusters, each having the regional metropolis as its natural centre. Prague represents the top of the organisational and thus functional hierarchy of the regional territorial structure. This is why there is an important concentration of public and private institutions, company headquarters, central administration bodies, universities, etc., which attract one another and as a whole form a natural control centre. With regard to this function Prague also represents the link between domestic and foreign economies. Its dominance among the regions of the CR results from a greater involvement in the global economy. Here we should say that from the statistical point of view there has just been an artificial separation of the core from its background. Prague itself does not represent a compact functional region, so its comparison with the other regions is not methodologically correct. It has to be taken as a whole with its background, i.e. the Central Bohemian Region. This will substantially reduce the difference between Central Bohemia and the other regions compared to the difference between Prague itself and the other regions. The other regional metropolises always remain metropolises of their regions apart from their efforts to sit higher in the hierarchy of the global network of metropolises.

Less dominant clusters may in some cases also be identified in the environs of some regional metropolises, for example:

- East Bohemian agglomeration formed by two main centres – Hradec Králové and Pardubice, complemented by some smaller towns as for example Chrudim. This complex forms the basic development axis of Eastern Bohemia;
- North Bohemian agglomeration formed by the functionally linked cities of the Krušné Hory region. The main centres are particularly Ústí nad Labem, Teplice, Most, Děčín, Litvínov, Chomutov;
- North-East agglomeration formed by the cities of Liberec and Jablonec nad Nisou.
- Ostrava agglomeration formed mainly by dominating Ostrava, the territory of which is directly linked to Bohumín, Frýdek-Místek, Karviná, Havířov and Orlová;
- Brno agglomeration formed by Brno, surrounded by smaller centres including Blansko and Vyškov. The position of Brno as the second largest centre of the CR is growing stronger;
- Plzeň is the main development centre of south-western Bohemia, strongly supported by the substantially smaller towns of Rokycany and Starý Plzenec;
- České Budějovice, the dominant natural centre of South Bohemia;
- Karlovy Vary, the natural centre of western Bohemia;
- Central Moravian cluster formed mainly by Olomouc and further smaller centres of the central Moravian region, represented by Prostějov and Přerov;
- Zlín forming a natural functional complex with Otrokovice.

The second conclusion deals with polarisation and density of clusters in relation to their geographical position. Visualisation of economic agglomerations in the CR shows an obvious decrease of size and density of clusters in the west-east direction, which is typical for the whole of Europe. Development of regional disparities in the CR is strongly influenced by geographical position, namely by the direct neighbourhood of developed countries, Germany and Austria, which are big sources of development impulses through investments that are geographically concentrated on the area close to the borders with these countries. Higher cooperation of regions on the western border particularly with cross-border Bavaria and Austria, as well as their better link to the whole European space, thus forms certain competitive advantages for these regions against the whole territory of Moravia and Silesia. The attractiveness of the western border area is mainly due to its physical closeness to developed regions, and it moreover reflects the cultural and economic relationship of these regions.

The third conclusion deals with mutual geographical accessibility of economic agglomerations, which is indisputably an important aspect influencing the intensity and orientation of economic relations among them. We may observe that the interactive spatial relations between the individual spatial system elements based on the intensity of economic relations in business cases of business entities are directly proportional to the value of intensity of economic relations and indirectly proportional to their distance. Market proximity in general reduces the total costs of entrance to external markets, however on the other hand it usually represents larger competition in the market, which presses companies to higher productivity. From the business entity point of view the closeness of the market in the basic spatial framework represents the potential for market expansion. The shorter the mutual distance between individual agglomerations the higher the potential. In activities with more intensive cooperation in the manufacturing industry the aspect of market closeness is comparable with the aspect of the closeness of key customers. The basic difference in the informational potential of the aspect of market closeness and the aspect of closeness of key customers is given by the fact that the first aspect emphasises the long-term stability of business location structure determining the territorial potential of final consumption of products and services by inhabitants, while the second aspect mainly emphasises the interaction relations arising within the business sphere.

The fourth conclusion deals with the possibility of further research that might be performed if the accounting data serving for the case study were not only limited to 2010 and were provided by all business entities from the territory of the CR. The whole case study of identification of economic agglomerations and spatial dependences based on accounting data of selected business entities is elaborated statically at present, i.e. without monthly or even multiannual comparison. If we took the point of view of time into account we could also monitor, for example, trends in formation and extinction of agglomerations, the influence of economic crisis on sizes and relations between agglomerations and other dependences. We can assume that if we had accounting data available from all entities based within the territory of the CR, not only identification of economic agglomerations, but also that of regional clusters as well as of geographical concentrations of mutually linked companies and institutions in a specific field, we would get closer to reality to a large extent.

7 Conclusion

After completion of the research based on accounting data obtained from 27 randomly selected business entities we can evaluate the results. We have confirmed that accounting data enable economic agglomerations to be identified and characterised despite the fact that the research data were only obtained from a limited number of business entities.

If data from more periods were available the development of individual agglomerations would also be visible. Visualisation of development stages of the individual agglomerations, their formation, development but also their extinction would be possible. Upon these data attention could be concentrated on the regions where investment would be advantageous.

The number of studies dealing with various aspects of geographic regional economy, which are gradually becoming an integral part of modern conceptions of regional development, has been growing recently in connection with regional economy. They thus widen the system of economic theories, which was characterised by insufficient reflection of globalisation processes, resulting from the idealised conception of lack of spatial dimensionality in the recent past. Increasing mutual dependence is a characteristic trait of global economy development. These phenomena may be described both by means of economic-statistical indicators obtained from statistic research, and by means of data from company information systems, which bear a wide spectre of information important for regional economy and development. Company information systems provide a large number of tools for effective mediation of information. Collection and distribution of information in the field of regional economy integrate both sides of information society development: creation of information and its transfer for further purposes.

References

1. Sodomka, P., Klčová, H.: Informační systémy v podnikové praxi. Computer Press, Brno (2010) ISBN 978-80-251-2878-7
2. Kovanicová, D.: Finanční účetnictví: světový koncept. Polygon, Praha (2005) ISBN 80-7273-129-7
3. Hanzal, P.: Výpočet HDP regionů a ČR produktovou metodou pomocí účetních dat podnikatelských subjektů. Improforum, 97–103 (2009)
4. Ketels, C.H.M., Mamedovic, O.: From clusters to cluster-based economic development. International Journal of Technological Learning, Innovation and Development 1, 375–392 (2008)
5. Majocchi, A., Presutti, M.: Industrial clusters, entrepreneurial culture and the social environment: The effects on FDI distribution. International Business Review 18(1), 76–88 (2009)
6. Huber, F.: Social capital of economic clusters: towards a network-based conception of social resources. Tijdschrift voor economische en sociale geografie 100(2), 160–170 (2009)
7. He, J., Fallah, M.H.: Is inventor network structure a predátor of cluster evolution? Technological Forecasting & Social Change 76, 91–106 (2009)
8. Jellema, J., Roland, G.: Institutional clusters and economic performance. Journal of Economic Behavior & Organization 79, 108–132 (2011)
9. Henderson, J.V.: Cities and development. Journal of Regional Science 50(1), 515–540 (2010)
10. Leitmanová, F., Krutina, J.: Sledování výkonnosti regionu - využití přidané hodnoty (zaměření na Jihočeský kraj). Ekonomický časopis: časopis pre ekonomickú teóriu, hospodársku politiku, spoločensko-ekonomické prognózovanie 57(10), 1018–1037 (2009)

11. Frenken, K., Van Oort, F., Verburg, T.: Related variety, unrelated variety and regional economic growth. místo neznámé. Regional Studies 41, 685–697 (2007) ISSN 1360-0591
12. Hoover, E.M.: The Location Theory and the Shoe and Leather Industries. Hardvard University Press, Cambridge (1937)
13. Dicken, P., Lloyd, P.E.: Location in Space: Theoretical Perspectives in Economic Geography, vol. 3. Harper and Row, New York (1990)
14. Malmberg, A., Malmberg, B., Lundequist, P.: Agglomeration and firm performance: economies of scale, localisation and urbanisation. Environment and Planning 32, 305–321 (2000)
15. Karlsson, C.: Clusters, Functional Regions and Cluster Policies. Elektronic Working Papaer Series. CESIS, 2007 Paper No. 84, http://www.infra.kth.se/comcen/cesis/documents/WP84.pdf (citace: September 30, 2011)
16. Enright, M.J.: Regional clusters: What we know and chat we should know. The University of Hong Kong (2001)
17. Tobler, W.: A computer simulating urban growth in the Detrit region. Economic Geography 46, 234–240 (1970)

Nonlinear Spatial Analysis of Dynamic Behavior of Rural Regions

Yi Chen, Guanfeng Zhang, Bin Zheng, and Ivan Zelinka

Abstract. We aim to examine the impact of economic growth with a carbon dioxide (CO_2) emission factor, particularly, our investigation focuses on the dynamic behaviors of the functional regions of a rural area. A spatial analysis approach that incorporates three components and CO_2 emission factor has been developed to evaluate the dynamic behaviors of the rural areas at an administrative county level. We adopt the Mendel genetic algorithm (Mendel-GA) to implement the technical computation, in which a Mendel genetic operator implies the random assignment by using the Mendel's principles and the data of gross domestic product (GDP) has been utilised to estimate the CO_2 emission of productive activities in the rural area. A functional region affecting index (FRAI) has been used to construct the fitness function

Yi Chen
Department of Mechanical, Electrical and Environmental Engineering,
School of Engineering and Built Environment, Glasgow Caledonian University,
Glasgow G4 0BA, UK
e-mail: leo.chen@gcu.ac.uk

Guangfeng Zhang
GREQAM, School of Economics, Aix-Marseille University, Marseille 13236, France
Department of Economics, Adam Smith Business School, University of Glasgow,
Glasgow, G12 8RT, UK
e-mail: guangfeng.f.zhang@gmail.com

Bin Zheng
Chongqing Institute of Green and Intelligent Technology,
Chinese Academy of Sciences, Chongqing 401120, China
e-mail: zhengbin@cigit.ac.cn

Ivan Zelinka
Department of Computer Science, Faculty of Electrical Engineering and
Computer Science, VŠB-TUO, 17 listopadu 15 708 33, Ostrava-Poruba,
Czech Republic
e-mail: ivan.zelinka@vsb.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 401–412.
DOI: 10.1007/978-3-319-00542-3_40

in the Mendel genetic algorithm evaluation. The real data simulation for Chongqing rural region indicates that the index of FRAI works well in the modeling process and suggests its potential as a technical indicator for the rural policy-making.

1 Introduction

In the past decade, China's fast economic growth has been contributing to the global economic growth whilst the economic growth has brought environmental problems as a serious issue [1, 2, 3]. Particularly, the effects of the large population, the rapid urbanization, the increasing householders wealth demands and the local economic reconstruction have influenced the regional development and caused low efficient environmental recycle. Being one of the representative research issues, the impacts of various air pollutants on the regional economic development, such as Carbon Monoxide (CO), Nitrogen Oxides (NO_x), Silicon Dioxide (SO_2), have been discussed.

As has been widely accepted, CO_2 emission is connected with most of the human activities through mainly fossil fuels energy consumption, for example a fossil-fuelled power station. To evaluate the dynamic behaviors of the particular regions concerning energy consumption and the CO_2 emissions, some interdisciplinary studies have been proposed as reviewed as follows. Berlingwolff and Wu [4] reviewed the historical development of urban growth models, and they summarized that the different disciplines and diverse theories could be brought together to generate new models, such as fuzzy logic theory and neural network theory. Nejadkoorki et al.[5] proposed a model to estimate the road traffic CO_2 emissions of an urban area with three components. Weiss et al. [6] presented a bottom-up model for estimating the non-energy use of fossil fuels and the CO_2 emissions. Briefly, these studies suggest the researches of the dynamic models need to be considered to integrate with the interdisciplinary factors, especially the interaction between the economic activities and the environmental factors. In this context, it is aim to examine the dynamic behaviors of the rural areas, particularly focus on the spatial features of the CO_2 emission and economic growth.

Our study adopts genetic algorithms (GA) as the technical methodology. As been widely proved, the GA is an ideal search tool for many computational applications, for example, determining optimal parameters, which was introduced in the 1960s by John Holland, inspired by Darwin's theory of evolution [7]. Compared with the other search methods like hill-climbing method, the GA is more robust and efficient in computational process, that is, the GA has the advantage of robustness over the other traditional search methods, which allows the GA to work efficiently in a lot of different practical applications[8, 9, 10, 11, 12, 13].

Farmers and herders used to breed their plants and animals to produce more useful hybrids, and the knowledge of the genetic mechanisms has been built up. In 1860s, Mendel studied and published his field experimental research on the selective cross-breeding of common pea plants over many given generations [14], in which Mendel discovers that certain traits show up in offspring without any blending of parental characteristics. For example, the pea flowers are either purple or white, intermediate colors do not appear in those offspring of cross-pollinated pea plants. Mendel's research only refers to plants, however the underlying principles of heredity also apply to human and animals because the mechanisms of heredity are essentially the same for all complex life forms on this planet. Another case is, Fig. 1[1] shows daffodils with different hybrid characteristics of color (white or whiteish, green, yellow, pink, orange, etc.), shape (flat, slim, wrinkled, etc.) and size.

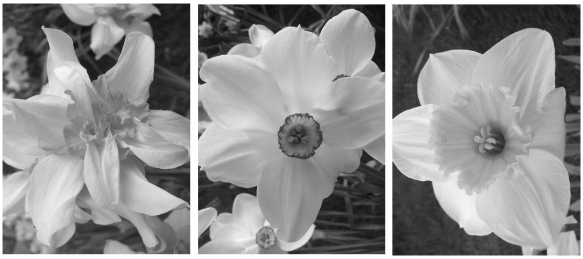


Fig. 1 Daffodils with different hybrid characters

Specifically, in this paper we utilize a GA method applying Mendel's principles (Mendel-GA)[8] for a spatial analysis that incorporates three economic components and a carbon dioxide (CO_2) emission factor. Our empirical analysis evaluates the shaped dynamic behaviors of the rural regions of Chongqing at the administrative county level.

2 CO_2 Emission Estimation of Productive Activities

According to the 'Revised 1996 IPCC Guidelines (Reference Manual, Volume 3)'[15], the estimation of the CO_2 emissions, which relates to the productive activity, requires three major factors for each of fuel type: the heat conversion factors (HCF), the carbon emission factor (CEF) and the fraction of carbon oxidized (FOC) in each sector.

Subject to the requirements outlined above and intended to ensure the comparability of country inventories, the IPCC approach to the calculation of emission encourages the use of fuel statistics collected by an officially recognized national body. However, practically, this IPCC approach can hardly be applied to the CO_2 emission of a local area for two reasons, firstly, recent

[1] Daffodils photos were taken by the first author at University of Glasgow, 17-Mar-2008.

satellite data has shown that the drop in coal consumption is probably unrealistic and the coal consumption data should not be used [2, 3]; secondly, the CO_2 emissions data by fuel combustion of each local district are not directly available.

To ensure the comparability for the local districts, an alternative of TCE/GDP based CO_2 emission estimation ($\hat{E}$) is proposed in this context as introduced by equation (1).

$$\hat{E} = \sum_{j=1}^{n_2} \kappa_j = \sum_{j=1}^{n_2} \tau\gamma_0\alpha_0\beta_0 \left(\hat{A}_j + \nu_0\hat{B}_j\right)\times 10^{-3} = \sum_{j=1}^{n_2} \tau\gamma_0\alpha_0\beta_0 \left(e_j g_j + \nu_0 f_j g_j\right)\times 10^{-3} \tag{1}$$

Where, $\hat{E}$ is the total TCE CO_2 emission estimation of the CQ rural area, $10^3 t$; κ_j is the CO_2 emission estimation of each district j by GDP data as listed in Table 1. g_j is the gross domestic product (GDP) of district j, $10^4 CNY$; e_j is the TCE energy consumption per GDP unit of district j, TCE/$10^4 CNY$; f_j is the electricity consumption per GDP unit of district j, $kwh/10^4 CNY$; ν_0 is the ratio of electricity to TCE, which is 0.01182 in this case[2]; $\hat{A}_j$ is the CO_2 estimation of the equivalent *energy* consumption of district j, t. $\hat{B}_j$ is the CO_2 estimation of the equivalent *electricity* consumption of district j, t; τ, γ_0, α_0 and β_0 are the molecular weight ratio of CO_2 to C, the TCE fraction of carbon oxidized, the TCE heat conversion factor and the TCE carbon emission factor, as given in IPCC Guidelines, $j = 1,...,n_2$.

As a study case, the TCE/GDP energy statistics of the rural area of Chongqing (CQ) is listed in Table 1. $j = 1,2,...n_2$, $n_2 = 31$. The energy consumption data constructed by the gross domestic product (GDP) g_j, the GDP related energy consumption e_j and the electricity consumption f_j, which are officially provided by the 'Chinese Energy Statistical Yearbook' and the 'Chongqing Statistical Yearbook'[16].

3 Nonlinear Modeling of the Functional Region with CO_2 Emission Uncertainty

The city of Chongqing is one of the largest four direct-controlled municipalities of China, which is the only such municipality in south west China. The rural area of CQ spans over 80,000 km^2, which covers 31 out of 40 district-level divisions [16]. Fig. 2 shows the location of CQ and the rural districts. In order to estimate the rural-urban (R-U) spatial interactions, a functional region affecting index (FRAI) with a 'law-of-gravity' interpretation can be characterized as equation (2) by using the basic form of a Cobb-Douglas (C-D) production function[9], which can provide a qualitative tool to perform a

[2] Electricity is converted to TCE by the equation 104 kwh = 1.229 TCE, that is, $\nu_0 = 1.229/104 = 0.01182$ [17]

Table 1 Chongqing energy statistical data [16]

j	District	e_j	f_j	g_j	$\hat{A}_j$	$\hat{B}_j$	κ_j
1	Wansheng	3.662	2578.51	323630	1185133.06	9442.50362	2,832.3
2	Shuangqiao	1.440	2214.30	300008	432011.52	3188.592	1,032.5
3	Fuling	1.610	1136.81	2534758	4080960.38	1830.2641	9,752.2
4	Changshou	2.669	1795.25	1438173	3838483.737	4791.52225	9,172.8
5	Jiangjin	2.523	1574.31	2192439	5531523.597	3971.98413	13,218.6
6	Hechuan	1.278	774.00	2034734	2600390.052	989.172	6,214.1
7	Yongchuan	1.093	604.35	1920729	2099356.797	660.55455	5,016.8
8	Nanchuan	1.560	941.99	1004876	1567606.56	1469.5044	3,746.1
9	Qijiang	1.061	1030.67	1252081	1328457.941	1093.54087	3,174.6
10	Tongnan	0.832	513.83	863084	718085.888	427.50656	1,716.0
11	Tongliang	0.849	656.26	1109987	942378.963	557.16474	2,252.0
12	Dazu	0.856	555.19	1018873	872155.288	475.24264	2,084.2
13	Rongchang	2.138	900.14	1100317	2352477.746	1924.49932	5,621.7
14	Bishan	1.170	1304.30	1134928	1327865.76	1526.031	3,173.2
15	Wanzhou	1.272	967.26	2560553	3257023.416	1230.35472	7,783.2
16	Liangping	0.967	684.99	754187	729298.829	662.38533	1,742.8
17	Chengkou	2.756	4384.73	195382	538472.792	12084.31588	1,287.1
18	Fengdu	0.766	623.26	574684	440207.944	477.41716	1,052.0
19	Dianjiang	0.853	669.16	815592	695699.976	570.79348	1,662.5
20	Zhongxian	0.848	415.53	778005	659748.24	352.36944	1,576.6
21	Kaixian	2.102	883.94	1106848	2326594.496	1858.04188	5,559.8
22	Yunyang	0.890	599.86	664771	591646.19	533.8754	1,413.8
23	Fengjie	0.875	525.84	753320	659155	460.11	1,575.2
24	Wushan	1.108	888.53	337123	373532.284	984.49124	892.6
25	Wuxi	1.172	828.84	235560	276076.32	971.40048	659.8
26	Qianjiang	0.963	876.09	604842	582462.846	843.67467	1,391.9
27	Wulong	1.159	839.29	498120	577321.08	972.73711	1,379.6
28	Shizhu	0.989	704.96	439487	434652.643	697.20544	1,038.7
29	Xiushan	2.464	4241.37	500364	1232896.896	10450.73568	2,946.5
30	Youyang	1.280	1748.30	329180	421350.4	2237.824	1,006.9
31	Pengshui	1.026	936.46	500474	513486.324	960.80796	1,227.1
	$\hat{E}$						103,203.1

distance-related interactive analysis and can be utilized as the fitness function for the Mendel-GA optimization.

Taking the district *Wulong* as an example, as shown in Fig. 2, O is the CQ city center, r_1 and r_2 are the shortest and longest distances[3] from O to *Wulong*, A and B are the closest and farthest locating-points of *Wulong*.

$$FRAI = \sum_{j=1}^{n_2} \left(\alpha_j \frac{y_j}{\kappa_j} \frac{z_j}{x_j r_j^{\beta} + \epsilon} \right)^{\eta_j} \tag{2}$$

[3] The distances r_{1j} and r_{0j} are provided by the GIS system *China Map*, `www.51ditu.com`

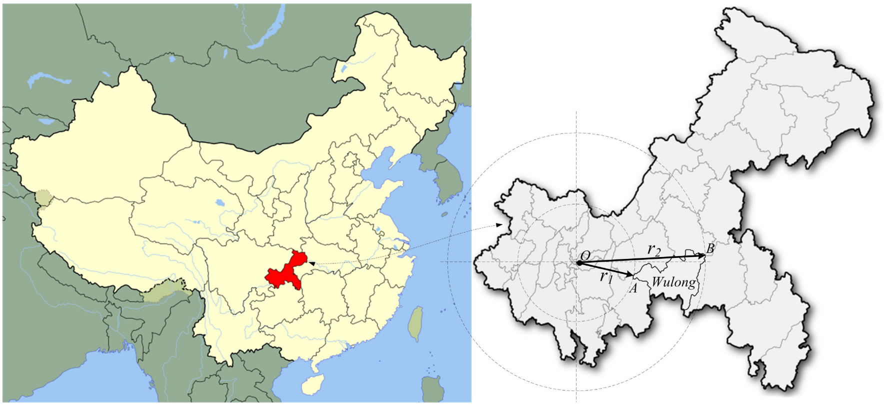


Fig. 2 Functional distances of Chongqing's districts

$$\eta_j = \begin{cases} 1 & \text{if } g_j \geq \bar{g} \\ -1 & \text{if } g_j < \bar{g} \end{cases} \tag{3}$$

$$\bar{g} = \frac{\sum_{j=1}^{n_2} g_j}{n_2} \tag{4}$$

Where, $FRAI$ is the functional region affecting index; x_j is the number of enterprises in the district j, n_2 is the number of the rural districts; y_j is the total profits of district j, $10^4 CNY$; z_j is the non-agricultural employment of district j, (10^4persons); $\alpha_j = \alpha_{1j}/\alpha_{0j}$ is the population effect coefficient of district j, α_{1j} is the local non-agricultural population, α_{0j} is the population of district j, (10^4persons); κ_j is the CO_2 emission of district j, as listed in Table 1; η_j is the output elasticity of district j, as given in equation (3), in which $\bar{g}$ is the mean of all the g_j in equation (4); β is the factor of the functional distance r_j, $\beta \in [0.5,3]$; ϵ is the floating-point relative accuracy, which prevents the singularity in case of the r_j^β is approaching to 0 and $FRAI$ is approaching to ∞ numerically; r_j is the mean value of point-to-point distance from the CQ city center to the town center of district j, as defined $r_j = (r_{j1} + r_{j2})/2$, kilometre (km).

4 Fitness Function Definition

In order to evaluate the dynamic behaviors relating to the functional distance, CO_2 emission and population in the economic production of CQ rural area, the $FRAI$ index has been adopted as the fitness function, which is defined as in equation (5) by the form of equation (2). $FRAI$ index is a hybrid measurement for the functional region and the CO_2 emission factor, which is a spatial determination of the agreement between the statistical data and the Mendel-GA driven data pool. All the statistical data for the parameters of equation (5) are listed in Table 2. The data sources are the 'Chongqing Statistical Yearbook 2009' [16] and the 'China Statistical Yearbook 2009' [18].

$$Fitness = F(X, Y, Z) = \sum_{j=1}^{n_2} \left(\alpha_j \frac{y_j}{\kappa_j} \frac{z_j}{x_j r_j^{\beta} + \epsilon} \right)^{\eta_j} \tag{5}$$

Table 2 Chongqing rural area statistical data [16, 18]

No. j	District	r_{j1}	r_{j2}	α_{0j}	α_{1j}	μ_{x_j}	μ_{y_j}	μ_{z_j}
1	Wansheng	58.3	97.4	26.82	13.23	578	-7220	2.90
2	Shuangqiao	70.5	79.5	5.01	2.82	299	5636	1.01
3	Fuling	38.4	108.3	113.80	33.04	3094	121250	10.42
4	Changshou	45.0	99.0	89.87	21.46	2467	117276	7.21
5	Jiangjin	27.2	117.0	148.65	40.02	2488	78214	10.87
6	Hechuan	41.1	104.7	153.89	32.04	2950	70417	6.36
7	Yongchuan	45.8	101.2	110.18	29.72	3309	130936	6.02
8	Nanchuan	38.8	111.9	66.09	11.71	2376	66422	2.33
9	Qijiang	44.6	124.9	95.00	22.18	1360	18777	4.72
10	Tongnan	69.4	119.4	93.26	11.99	1216	9372	2.09
11	Tongliang	69.3	84.7	82.42	14.95	1857	31717	2.33
12	Dazu	53.8	105.9	95.02	17.72	1890	40290	3.62
13	Rongchang	81.6	122.5	83.07	18.51	1585	60004	3.29
14	Bishan	29.8	56.1	62.14	15.92	1736	83208	3.39
15	Wanzhou	181.3	257.6	172.54	51.06	4123	96974	11.64
16	Liangping	133.8	194.3	91.07	11.81	559	47499	3.12
17	Chengkou	301.3	359.2	24.15	2.95	331	14344	0.96
18	Fengdu	100.8	162.0	82.44	15.05	883	25533	2.34
19	Dianjiang	86.6	146.7	93.94	14.43	1938	25829	3.65
20	Zhongxian	119.6	190.9	99.22	16.01	2272	17161	1.93
21	Kaixian	202.4	327.2	159.72	21.29	1590	6861	4.71
22	Yunyang	240.9	312.8	133.58	18.40	732	528	3.14
23	Fengjie	268.2	356.7	104.76	13.22	898	7961	2.49
24	Wushan	325.6	399.6	62.35	10.29	453	2902	1.71
25	Wuxi	294.4	393.5	53.52	6.29	571	3111	1.73
26	Qianjiang	186.5	235.9	52.17	10.36	576	52422	2.45
27	Wulong	68.7	144.1	41.01	5.92	650	3362	1.84
28	Shizhu	145.8	209.4	53.34	8.83	720	11305	2.27
29	Xiushan	241.5	291.0	64.35	8.62	760	32654	1.70
30	Youyang	182.1	274.8	80.81	8.56	958	1712	2.15
31	Pengshui	138.3	200.4	67.38	6.67	614	36854	1.97

With the following assumptions of the parameter uncertainties, where

○ $X = [x_1, x_2, ..., x_j, ..., x_{n_2}]$ is the vector of the enterprise number, $X \sim N(\mu_X, \sigma_X^2)$. Let $\mu_X = [\mu_{x_1}, \mu_{x_2}, ..., \mu_{x_j}, ..., \mu_{x_{n_2}}]$ and $\sigma_X = [\sigma_{x_1}, \sigma_{x_2},$

..., σ_{x_j}, ..., $\sigma_{x_{n_2}}$] be the mean and the standard deviation (STD) vectors of X, respectively, that is, $x_j \sim N(\mu_{x_j}, \sigma^2_{x_j})$, μ_{x_j} and σ_{x_j} are the mean and the standard deviation of x_j;

∘ Y = [y_1, y_2, ..., y_j, ..., y_{n_2}] is the vector of the total profits, $Y \sim N(\mu_Y, \sigma^2_Y)$. Let μ_Y = [μ_{y_1}, μ_{y_2}, ..., μ_{y_j}, ..., $\mu_{y_{n_2}}$] and σ_Y = [σ_{y_1}, σ_{y_2}, ..., σ_{y_j}, ..., $\sigma_{y_{n_2}}$] be the mean and the standard deviation vectors of Y, respectively, such that $y_j \sim N(\mu_{y_j}, \sigma^2_{y_j})$, μ_{y_j} and σ_{y_j} are the mean and the standard deviation of y_j;

∘ Z = [z_1, z_2, ..., z_j, ..., z_{n_2}] is the vector of the non-agricultural employment, $Z \sim N(\mu_Z, \sigma^2_Z)$. Let μ_Z = [μ_{z_1}, μ_{z_2}, ..., μ_{z_j}, ..., $\mu_{z_{n_2}}$] and σ_Z = [σ_{z_1}, σ_{z_2}, ..., σ_{z_j}, ..., $\sigma_{z_{n_2}}$] be the mean and the standard deviation vectors of Z, respectively, such that $z_j \sim N(\mu_{z_j}, \sigma^2_{z_j})$, μ_{z_j} and σ_{z_j} are the mean and the standard deviation of z_j;

∘ δ_{x_j}, δ_{y_j} and δ_{z_j} are the coefficients of variation (CV) of x_j, y_j and z_j respectively, with the definition in equation (6). The rest parameters α_j, β, r_j, κ_j, ϵ and η_j are as introduced in section 3;

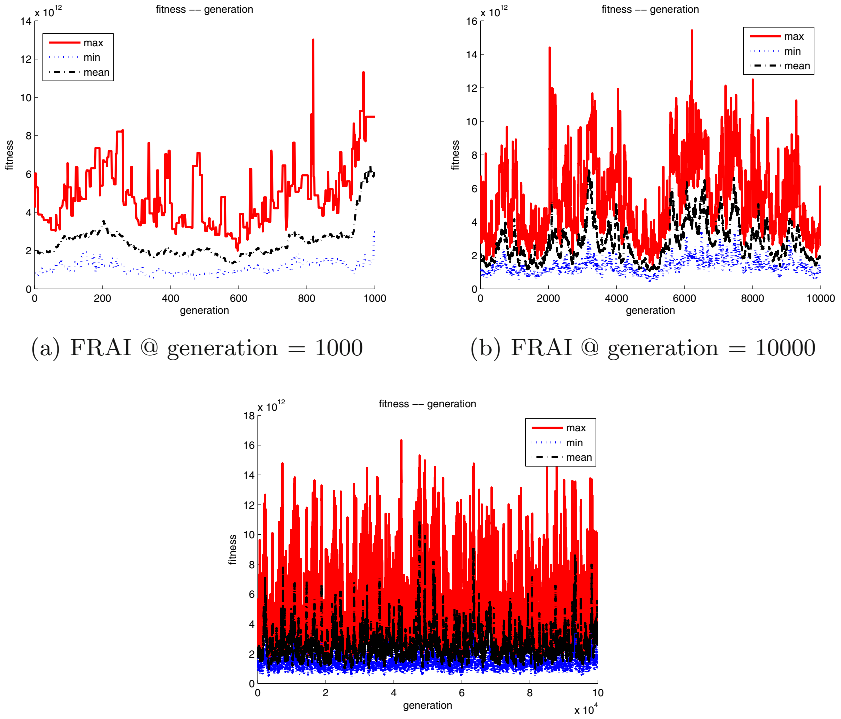
$$\delta_j = \frac{\sigma_{x_j}}{\mu_{x_j}} \tag{6}$$

5 Empirical Results and Discussions

The empirical results for the Mendel-GA driven hybrid modeling of the CQ functional region is performed by the *SGALAB* [19] and SECFLAB[20], the parameters of Mendel-GA calculation are reported in Table 3, in which the selection operator is tournament method; the crossover and mutation operators are single point method; the encoding is binary method; the Mendel percentage is with full chromosome length; the coefficients of variation δ_j for all x_j, y_j and z_j are set to 15%; the experiment number is 100.

Table 3 Parameters for the simulations

max generations	1000/10000/100000
crossover probability	0.8
mutation probability	0.001
population	50
selection operator	tournament
crossover operator	single point
mutation operator	single point
encoding method	binary
experiment number	100
CV (δ_j)	15 %
Mendel percentage	1 (full chromosome length)



(a) FRAI @ generation = 1000

(b) FRAI @ generation = 10000

(c) FRAI @ generation = 100000

Fig. 3 FRAI fitness plot @ (a) generation = 1000, (b) 10000 and (c) 10000

Table 4 Fitness uncertainty analysis

measures	$\mu_{F(X,Y,Z)}\|_{max}$	$\mu_{F(X,Y,Z)}\|_{mean}$	$\mu_{F(X,Y,Z)}\|_{min}$
Max	1.6332	1.1008	0.45981
Min	0.1395	0.0974	0.04727
$Mean$	0.4994	0.2582	0.12872
$Median$	0.4404	0.2345	0.12222
$Mode$	0.5695	0.1193	0.09740
STD	0.2267	0.1034	0.03863
Q_1	0.3340	0.1892	0.10253
Q_2	0.4404	0.2345	0.12222
Q_3	0.6089	0.2979	0.14697

Figs. 3(a), 3(b) and 3(c) are the fitness plots for the generations g = 1000, 10000 and 10000, whose ‘x-axis’ is the evolution generations of Mendel-GA and ‘y-axis’ is the fitness values. These figures demonstrate the short-term (g = 1000), middle-term (g = 10000) and long-term (g = 100000) dynamic behaviors of the functional regions of CQ rural area, in which the solid line, dotted line and dash-dot line are respectively the ‘max’, ‘min’ and ‘mean’ values of

Table 5 An estimation $\hat{X}$, $\hat{Y}$ and $\hat{Z}$ optimized by the Mendel-GA

No. j	District	$\hat{x}_j$	$\Delta x_j(\%)$	$\hat{y}_j$	$\Delta y_j(\%)$	$\hat{z}_j$	$\Delta z_j(\%)$
1	Wansheng	454.1	-21.4	-5873.9	-18.6	1.0	-65.3
2	Shuangqiao	183.2	-38.7	5411.9	-4.0	0.7	-28.1
3	Fuling	527.6	-82.9	73545.4	-39.3	8.9	-14.6
4	Changshou	1678.9	-31.9	97562.5	-16.8	7.2	-0.7
5	Jiangjin	2251.1	-9.5	38926.7	-50.2	4.1	-62.2
6	Hechuan	1214.2	-58.8	58146.4	-17.4	2.5	-60.9
7	Yongchuan	1442.9	-56.4	101450.4	-22.5	3.6	-39.6
8	Nanchuan	1723.4	-27.5	35494.6	-46.6	2.2	-4.0
9	Qijiang	1209.3	-11.1	8935.8	-52.4	4.4	-6.2
10	Tongnan	785.2	-35.4	2930.0	-68.7	1.7	-17.3
11	Tongliang	946.9	-49.0	21328.0	-32.8	1.6	-30.3
12	Dazu	988.1	-47.7	8948.8	-77.8	3.4	-5.5
13	Rongchang	1149.3	-27.5	9618.9	-84.0	1.3	-60.2
14	Bishan	1608.0	-7.4	42294.6	-49.2	1.2	-65.7
15	Wanzhou	1276.9	-69.0	36627.3	-62.2	9.6	-17.4
16	Liangping	518.2	-7.3	8379.1	-82.4	1.3	-59.5
17	Chengkou	188.7	-43.0	4284.6	-70.1	0.2	-80.3
18	Fengdu	655.1	-25.8	5916.4	-76.8	2.3	-3.3
19	Dianjiang	1541.9	-20.4	9170.7	-64.8	0.6	-82.3
20	Zhongxian	1301.1	-42.7	6333.3	-63.1	0.3	-84.7
21	Kaixian	746.1	-53.1	2586.9	-62.3	2.4	-49.9
22	Yunyang	638.6	-12.8	521.9	-1.1	3.0	-5.1
23	Fengjie	366.1	-59.2	7509.2	-5.7	2.3	-7.0
24	Wushan	173.0	-61.8	1324.4	-54.4	1.1	-35.3
25	Wuxi	397.2	-30.4	2603.2	-16.3	1.6	-9.3
26	Qianjiang	100.5	-82.6	19219.0	-63.3	1.7	-30.7
27	Wulong	597.5	-8.1	1630.7	-51.5	1.1	-39.3
28	Shizhu	200.6	-72.1	8374.4	-25.9	0.5	-76.3
29	Xiushan	230.4	-69.7	28798.1	-11.8	1.7	-2.0
30	Youyang	779.2	-18.7	484.5	-71.7	0.8	-64.0
31	Pengshui	347.31	-43.4	16500.8	-55.2	0.8	-60.7

fitness evolution process. Different from the fitness shape for engineering applications, the fluctuations of the fitness curves indicate the non-monotonic behaviors of the CQ rural area.

As shown in Table 4, nine measures of the dynamic behaviors of fitness plots are devised to describe the uncertainties of the ‘max’ ($\mu_{F(X,Y,Z)}|_{max}$), ‘mean’ ($\mu_{F(X,Y,Z)}|_{mean}$) and ‘min’ ($\mu_{F(X,Y,Z)}|_{min}$) fitness plots, which includes: Max, Min, $Mean$, $Median$, $Mode$, STD, 25th percentile (Q_1), 50th percentile (Q_2) and 75th percentile (Q_3).

Table 5 presents one of the set of estimation $\hat{X} = [\hat{x}_1, \hat{x}_2, ..., \hat{x}_j, ..., \hat{x}_{n_2}]$, $\hat{Y} = [\hat{y}_1, \hat{y}_2, ..., \hat{y}_j, ..., \hat{y}_{n_2}]$ and $\hat{Z} = [\hat{z}_1, \hat{z}_2, ..., \hat{z}_j, ..., \hat{z}_{n_2}]$ for each district in CQ rural area. Compared with the year book data μ_{x_j}, μ_{y_j} and μ_{z_j} in Table 2, the negative percentage errors of $\Delta x_j(\%)$, $\Delta y_j(\%)$ and $\Delta z_j(\%)$

indicate that the scale of the non-agricultural activities in CQ rural area are in a decreasing trend, which matches the practical situation, that is, the rural area of CQ is designed as an agriculture oriented functional region, administratively, meanwhile, the urban area of CQ is the industry oriented functional region [16].

6 Conclusions and Future Studies

Technical analysis is frequently adopted to assist the region planning, which aims to take good advantage of the available statistical data. This paper provides an evolutionary alternative of dynamic behaviors modeling of functional region using the distance factor and the CO_2 emission factor. In this paper, the Mendel-GA is firstly introduced, the GDP based TCE method is applied to the CO_2 emission estimation is discussed. For the hybrid modeling of functional region of the CQ rural area, the FRAI index is proposed to describe the interaction between the production inputs and outputs with the source data uncertainties, which indicates a sort of distance-based relationship between a FRAI combination of the X, Y and Z inputs and the possible maximum output under current productive capability, with which the experience of a policy-makers can be incorporated in a natural or artificial way. It is clear that the CO_2 emission is related to the complicated dynamic behaviors of the functional region, the GDP/TCE method for the CO_2 emission estimation is a practical way to deal with the lack of direct fossil fuel data source.

Our empirical simulation demonstrated the practical behaviors of functional region of CQ rural area, which suggested some interesting issues for further investigation on the real-time public sector policy making, in which the consumer price index (CPI) will be one of the dynamic decision criteria for the further reliability analysis of a multi-state FRAI modeling or the analysis of the regional system maintenance and sustainability.

Acknowledgements. The authors would like to acknowledge the partial supports provided by the National Natural Science Foundation of China No. 61179059, 51105061 and 50905028. This paper was also supported by the Development of human resources in research and development of latest soft computing methods and their application in practice project, reg. no. CZ.1.07/2.3.00/20.0072 funded by Operational Programme Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic

References

1. Liu, J., Diamond, J.: China's Environment in A Globalizing World. Nature 435, 1179–1186 (2005)
2. Peters, G., Weber,C., Liu, J.-R.: Construction of Chinese Energy and Emissions Inventory, Report No.4/2006, Norwegian University of Science and Technology (NTNU) Industrial Ecology Programme (2006)

3. Akimoto, H., Ohara, T., Kurokawa, J.-I., Horiid, N.: Verification of Energy Consumption in China during 1996-2003 by using Satellite observational data. Atmospheric Environment 40(40), 7663–7667 (1999)
4. Berling-Wolff, S., Wu, J.-G.: Modeling Urban Landscape Dynamics: A Review. Ecological Research 19(1), 119–129 (2004)
5. Nejadkoorki, F., Nicholson, K., Lake, I., Davies, T.: An Approach for Modeling CO2 Emissions from Road Traffic in Urban Areas. Science of The Total Environment 406(1-2), 269–278 (2008)
6. Weiss, M., Neelis, M., Blok, K., Patel, M.: Non-energy Use of Fossil Fuels and Resulting Carbon Dioxide Emissions: Bottom-up Estimates for the World as A Whole and for Major Developing Countries. Climatic Change 95(3-4), 369–394 (2009)
7. Holland, J.H.: Adaptation in Natural and Artificial Systems. The University of Michigan Press (1975)
8. Chen, Y., Zhang, G.-F.: Exchange Rates Determination Based on Genetic Algorithms using Mendel's Principles: Investigation and Estimation under Uncertainty. Information Fusion 12(8), 2631–2637 (2012)
9. Chen, Y., Song, Z.-J.: Spatial analysis for functional region of suburban-rural area using micro genetic algorithm with variable population size. Expert Systems with Applications 39(7), 6469–6475 (2012)
10. Chen, Y., Ma, Y., Lu, Z., Peng, B., Chen, Q.: Quantitative Analysis of Terahertz Spectra for Illicit Drugs using Adaptive-range Micro-genetic Algorithm. Journal of Applied Physics 110(4), 44902–44910 (2011)
11. Chen, Y., Ma, Y., Lu, Z., Qiu, L.X., He, J.: Terahertz Spectroscopic Uncertainty Analysis For Explosive Mixture Components Determination Using Multiobjective Micro Genetic Algorithm. Advances in Engineering Software 42(9), 649–659 (2011)
12. Chen, Y., Ma, Y., Lu, Z., Xia, Z.-N., Cheng, H.: Chemical Components Determination via Terahertz Spectroscopic Statistical Analysis using Micro Genetic Algorithm. Optical Engineering 50(3), 34401–34412 (2011)
13. Chen, Y., Cartmell, M.P.: Multi-objective Optimisation On Motorised Momentum Exchange Tether for Payload Orbital Transfer. In: Proceedings of IEEE Congress on Evolutionary Computation, pp. 987–993 (2007)
14. Mendel, Gregor: Experiments in Plant Hybridization (1865), www.mendelweb.org/Mendel.html
15. IPCC,Revised 1996 IPCC Guidelines for National Greenhouse Gas Inventories: Reference Manual, vol. 3. NGGIP Publications (1997)
16. National Bureau of Statistics of China, Chongqing Statistical Yearbook 2009. China Statistics Press (2009)
17. National Bureau of Statistics of China, Chinese Energy Statistical Yearbook 2009. China Statistics Press (2009)
18. National Bureau of Statistics of China, China Statistical Yearbook 2009. China Statistics Press (2009)
19. Chen, Y.: SGALAB - Simple Genetic Algorithm Laboratory Toolbox (2009), http://www.mathworks.co.uk/matlabcentral/fileexchange/5882
20. Chen, Y.: SECFLAB - Simple Econometrics And Computational Finance Laboratory Toolbox (2012), http://www.mathworks.com/matlabcentral/fileexchange/38120

Complex System Simulation Parameters Settings Methodology

Michal Janošek, Václav Kocian, and Eva Volná

Abstract. This article extends (Janošek and Kocian 2013) and deals with simulation parameters setting methodology proposal for complex system behaviour adaptation. Therefore the article focuses on system adaptation where there is an effort to find such means of mediating the system's behaviour that would make it possible to adapt to the current state of the system and the environment and react to the changes so that the desired behaviour of the system is kept in specified limits or patterns of behaviour. The instruments of regulating the system's behaviour are its parameters. In recognizing the parameters' importance, this work is inspired by the leverage point theory (Meadows 1999) and builds on its approach to the system cognizance. The adaptation of the system's behaviour itself consists of recognizing these characteristic patterns using neural networks and the subsequent mediation of the system's behaviour through selected parameters and their action ranges based on pre-prepared expectations of what will happen if the system's behaviour exhibits a known characteristic pattern.

1 Complex System

Every real system can be seen as a complex phenomenon. It consists of many interconnected parts mutually reacting and it is very complicated to separate them. The system's behaviour is impossible to deduce from the behaviour of its individual components. The more parts and more interactions the system consists of, the more complicated it is (Gershenson 2002).

The complexity of a system C_{sys} increases with the number of its elements $\#\bar{E}$, interactions $\#\bar{I}$ between elements, complexities of the elements C_e and complexities of the interactions C_i (1).

$$C_{sys} \sim \begin{cases} \#\bar{E} \\ \#\bar{I} \\ \sum_{j=0}^{\#\bar{E}} C_{e_j} \\ \sum_{k=0}^{\#\bar{I}} C_{i_k} \end{cases} \tag{1}$$

Michal Janošek · Václav Kocian · Eva Volná
University of Ostrava, 30. dubna 22, 70103 Ostrava, Czech Republic
e-mail: {michal.janosek,vaclav.kocian,eva.volna}@osu.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 413–422.
DOI: 10.1007/978-3-319-00542-3_41

It is not possible to simply and without the context to decide if the system is complex or not. Dynamic complex systems consist of many interconnected positive and negative feedback loops where a change in one part of the system leads to a cascade change in a number of connected components which (in part feeding back) positively or negatively affect the initial component. If there is a variable time delay between these effects, it becomes nearly impossible to make predictions, because we do not know who will affect whom first and thus whether the effect is dampened before it has had the chance to get amplified or not (Gershenson 2007).

We will never be able to predict such systems' behaviour completely. There will always be mistakes, surprises and unexpected problems. How can we deal with that then? One of the solutions is to adapt to any changing conditions of the system's behaviour by reflecting the current situation. To adapt to any change (expected or unexpected), it should be sufficient to compensate any deviation from the desired course. In case the response to a deviation comes quickly enough, the way of regulation (feedback-based) can be very effective. It does not matter how complicated the system is (how many factors and interactions it has) in case we have efficient means of control (Volna et. al 2012).

2 Leverage Points

If we want to adapt to changing system's behaviour, it is crucial to find certain particular places inside the system. By "pushing" on these places, we can change the system's behaviour in the desired way. Leverage points are places in the system where a small change in one small thing can produce a big change in everything (Pelánek 2011). A leverage point is an amount of intervention to a place in the system (input) which is sufficient to affect the system's behaviour (output). The higher the leverage point is, the less effort to affect the system's behaviour is needed. If we induce a relatively low amount of input power, the low leverage point will develop a small change in the system. On the contrary, a high leverage point will develop a huge change in a system. The magnitude of the system's behaviour change is in direct correlation with the leverage point level and the induced input power (2).

$$leverage\ point \times power = change \tag{2}$$

This equation (2) illustrates how to succeed in a complex problem solution. It is necessary to study the system and find its leverage points. They will help us, using a low amount of power, to change the system's behaviour in the desired way.

Simple problems can be solved by inducing to some low leverage point. To solve more complex problems, it is necessary to find a higher leverage point. Usually we try to solve problems around us using lower leverage points only. But this solution doesn't have to work in a longer term because the main cause of the problem is omitted.

There is no straight recipe how to find these points. It is always crucial to analyse the particular system and the problem we are working on.

But there is a tool which can help us to find them. Donella H. Meadows developed, with the help of other colleagues from the branch, the leverage points' hierarchy (Meadows 1999). Even though this hierarchy is made for system dynamics (top-down approach), we can find useful points for agents-based modelling (bottom-up approach) as well.

Leverage points' hierarchy
(Places to intervene in a system in increasing order of effectiveness)

12. Constants, parameters, numbers (such as subsidies, taxes, standards).
11. The sizes of buffers and other stabilizing stocks, relative to their flows.
10. The structure of material stocks and flows (such as transport networks, population age structures).
9. The lengths of delays, relative to the rate of system change.
8. The strength of negative feedback loops, relative to the impacts they are trying to correct against.
7. The gain around driving positive feedback loops.
6. The structure of information flows (who does and does not have access to information).
5. The rules of the system (such as incentives, punishments, constraints).
4. The power to add, change, evolve, or self-organize system structure
3. The goals of the system.
2. The mindset or paradigm out of which the system its goals, structure, rules, delays, parameters arises.
1. The power to transcend paradigms.

3 Methodology Requests

The primary request of this methodology is to effectively adapt complex system's parameters in order to mediate its behaviour. That means that system's adaptation is done relatively fast in contrast to the delays in the system and is adequate to the changes in the system, e.g. with a pre-defined deviation from the expected behaviour.

What is expected from the methodology? Its usage can enable to mediate particular system behaviour based on our goals and expectations. What should be input conditions, inputs and outputs of the methodology?

The necessary precondition is an already working simulation made in one of available simulation software. It means that this particular simulation fulfils the behaviour of a real system or reflects an abstract design of a non-existing system.

The input to this methodology should be all input parameters of the simulation which are relevant. Next, the requested values of output indicators (parameters) of the system's behaviour are desired. The pre-defined characteristic behaviour patterns of output system indicators must also be considered.

The output should be action parameters from the chosen input parameters with the magnitude of their influence on the output indicators together with the effective action range of these action parameters.

3.1 Methodology

The methodology consists of several main phases (Figure 1). Agents and Parameters Identification, Parameters' Hierarchy, Simulation, Behaviour Pattern Recognition, Adaptation of Parameters. It is possible to return back to any previous phase if next phases are altered accordingly.

Agents and Parameters Identification. In the first phase it is crucial to identify all agents and their parameters in the simulation model. Agents are categorised by their kind to the environment agents, common agents and mediators. Its parameters are identified for each agent.

Parameters' Hierarchy. Now, it is known what kind of agents and parameters are presented in the simulation model. Next, it is crucial to find out what influence each parameter has on the corresponding output indicator. That is the part where an expert assigns appropriate leverage points to each parameter.

Simulation. In this phase we have to conduct a necessary amount of experiments when a behaviour of the system is observed. Characteristic patterns of the system behaviour are searched in the output indicators. These patterns are saved and described. Particular patterns are determined. These patterns are split into several categories, each of which has similar properties. A training set is created by this procedure. For every category, it is necessary to define what will happen if any pattern from any category emerges in the system's behaviour.

Behaviour Pattern Recognition. In this phase the neural network is adapted by the prepared training set. This task makes use of our own developed tool called EDU Sandbox[1]. Then the neural network is able to generalise the learned training set. After that, when the simulation is running, the neural network can recognize similar behaviour pattern it already knows and returns that information back to the simulation software.

Adaptation of Parameters. Based on the recognised patterns in the system's behaviour it is possible to adapt action parameters' values according to the prepared expectations. Adaptation logic has to be programmed in the simulation software.

The flowchart of the methodology (Figure 1) represents a more detailed step-by-step procedure with each above-mentioned phase.

[1] http://sourceforge.net/projects/esbox/ - Václav Kocian

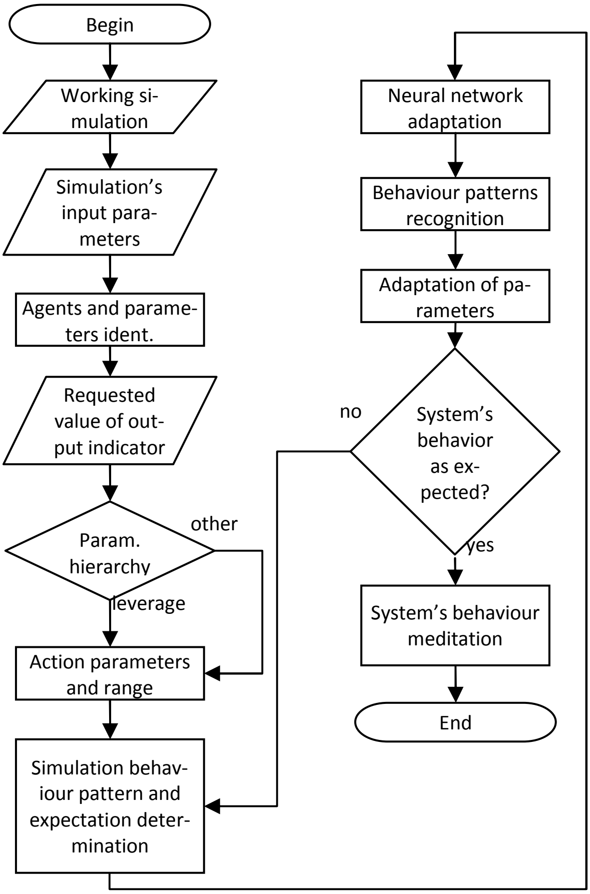
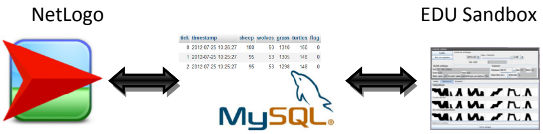

Begin
Working simulation
Simulation's input parameters
Agents and parameters ident.
Requested value of output indicator
Param. hierarchy
other
leverage
Action parameters and range
Simulation behaviour pattern and expectation determination
Neural network adaptation
Behaviour patterns recognition
Adaptation of parameters
System's behavior as expected?
no
yes
System's behaviour meditation
End

Fig. 1 Methodology flowchart

4 Experiment

For our experiment we use a simulation prepared by Carlos Gershenson (Gershenson 2007) which is available at http://turing.iimas.unam.mx. We have modified this simulation to able to connect it to our pattern recognition utility. Adaptation logic was added as well. The model simulates city traffic where traffic lights try to "self-organise" to improve the traffic flow in the city. It consists of an abstract traffic grid with intersections between cyclic single-lane arteries of two types: vertical or horizontal. Cars only flow in a straight line, either eastbound or southbound. Each crossroad has traffic lights that allow traffic flow in only one of the intersecting arteries with a green light.

The experiment used the following tools: NetLogo simulation software (Wilensky 1999), MySQL database, EDU Sandbox. Communication between these tools is following (Figure 2).



tick	timestamp	sheep	wolves	grass	turtles	flag
0	2012-07-25 10:25:27	100	50	1310	150	0
1	2012-07-25 10:25:27	95	53	1305	148	0
2	2012-07-25 10:25:27	95	53	1298	148	0

Fig. 2 Communication between NetLogo and EDU Sandbox

The simulation runs in the NetLogo simulation software. The simulation consists of discrete steps. Information about each simulation step is stored into the MySQL database. This information is read by the EDU Sandbox utility. This utility is able to learn patterns beforehand. EDU Sandbox reads in real-time data from the database and reports back to NetLogo if any known patterns have been recognized. Based on the recognised pattern, the adaptation logic in NetLogo adapts the chosen action parameters to mediate simulation's behaviour.

As we can see (Figure 3) there are different strategies of self-organising traffic lights. Each strategy has its advantage and disadvantage. The most significant difference is with about *1000* vehicles. The *sotl-platoon* strategy begins to be ineffective, while the *sotl-phase* strategy tends to be better for more than about *1000* vehicles. Gershenson's work does not implement any adaptation strategy where it would be possible to switch between strategies based on the current traffic status. Our experiment tries to do that.

We use one output indicator called *Average Speed of Cars* which tells us what the average speed of all vehicles in the simulation is. The goal will be to allow the vehicles to pass through this grid system as fast as possible regardless the traffic density.

Simulation input parameters

The first step is to choose all relevant parameters from all available. We have chosen three parameters (Table 1).

Table 1 Input parameters

Parameter	Purpose	Default value	Range
number	number of vehicles	320	1-3000
speed-limit	maximum speed	1.0	0.0-1.0
control	mediator's (tradic lights) strategy	sotl-platoon	marching optim cut-off sotl-request sotl-phase sotl-platoon no-corr

Figure 3 shows the influence of every parameter to the *Average Speed of Cars* output indicator.

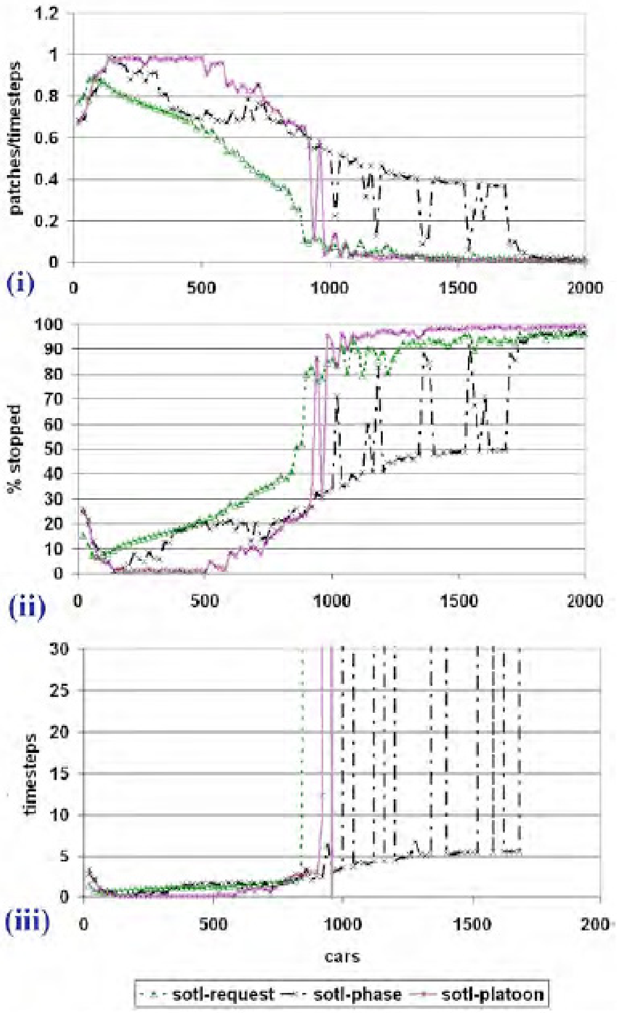


Fig. 3 Traffic lights SOTL strategies comparison (Gershenson 2007)

Agents and Parameters Identification

Dividing the parameters into three types (environment, common and mediator) gives us information about what part of the simulation belongs to each particular parameter. The parameters are divided based on the agent type:

Environment – *speed-limit;*
Common – *number;*
Mediators – *control.*

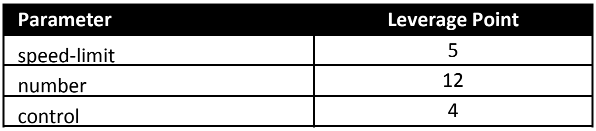
Desired value of output indicator

For our purpose, there is one key indicator which is *Average Speed of Cars*. We require that its value would be as high as possible.

Parameters' Hierarchy

Now, we need to find out what influence on the corresponding output indicator each parameter has. We can evaluate this influence using leverage points hierarchy (Table 2).

Table 2 Input parameters



Parameter	Leverage Point
speed-limit	5
number	12
control	4

Parameter *numbers* refers to leverage point 12 because it is just a constant. Parameter *speed-limit* refers to leverage point 5 because it is a sort of constraint. Parameter *control* refers to leverage point 4 because it can self-organise the structure of the system.

Action parameters and their range

For the choice of action parameters it is crucial to define their range where behaviour of the output indicator is desirable. It includes the minimum and maximum values of the output indicator as well as its stability and possible oscillations.

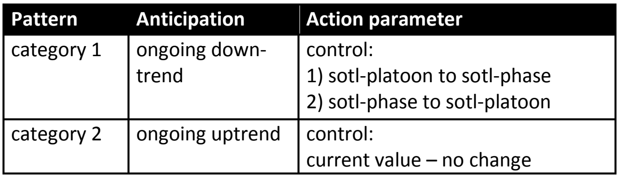
Simulation

In this phase we have to conduct a necessary amount of experiments with action parameters' candidates and their range to observe characteristic system's behaviour observable on the output indicator.

Pattern and expectation determination

In this step, we should have characteristic behaviour patterns split into several categories. Each category represents one type of behaviour and an anticipation what would happen after that behaviour emerges. In (Table 3) we have two categories with corresponding anticipations.

Table 3 Patterns and their anticipations



Pattern	Anticipation	Action parameter
category 1	ongoing downtrend	control: 1) sotl-platoon to sotl-phase 2) sotl-phase to sotl-platoon
category 2	ongoing uptrend	control: current value – no change

Neural network adaptation

In this step, all patterns are presented to the neural network to learn them. Here are some examples of training sets (Figure 4).

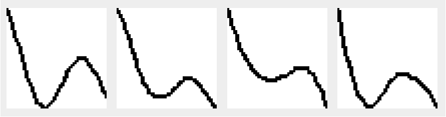


Fig. 4 Category 1 patterns example

Behaviour Pattern Recognition

When all patterns have been adapted, we can launch the simulation again and see how patterns are recognised. In our case, the recognized patterns appear in our EDU Sandbox utility and are transferred via MySQL to NetLogo according to adaptation logic.

Adaptation of Parameters

In case we had chosen correct patterns and their adaptation had been held successfully, we can adapt the chosen input action parameters of the simulation using application logic implemented in NetLogo. The system should behave according to our expectation. In our case, it should switch between two strategies in order to allow the vehicles to pas as fast as possible (Figure 5).

Figure 5 shows the experiment's result. First, original experiments for *sotl-phase* and *sotl-platoon* control mechanism were repeated. Two other lines, *pattern*

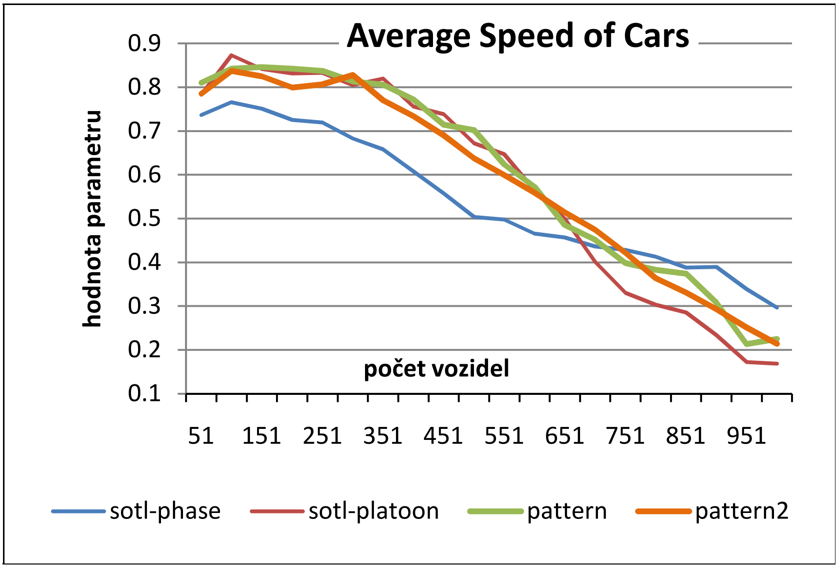


Fig. 5 Results of adapting system's behaviour

and *pattern2* represent our adaptation strategies, each with slightly different patterns. These two strategies adaptively combine both original strategies *sotl-phase* and *sotl-platoon* based on the traffic density.

5 Conclusion and Future Work

In this article we have presented our methodology proposal for the system's behaviour adaptation using its so-called action parameters. The methodology consists of five main phases: Agents and Parameters Identification, Parameters' Hierarchy, Simulation, Behaviour Pattern Recognition, Adaptation of Parameters. Each phase is designed as an independent part so it is possible to return to any previous phase. A detailed methodology is presented in the flowchart (Figure 1).

In the Experiment chapter we have verified that the methodology works as expected and is able to adapt system's behaviour in the desired way. This experiment was only one of possible scenarios. If we compare the results without and with our methodology, our results are better than the original experiment (Gershenson 2007) because we were able to mediate the system's behaviour by adaptation of input parameters of the simulation.

Acknowledgments. The research described here has been financially supported by University of Ostrava grant SGS23/PřF/2013. 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.

References

Gershenson, C.: Complex philosophy. In: La Habana, C. (ed.) Proceedings of the 1st Biennial Seminar on Philosophical, Methodological & Epistemological Implications of Complexity Theory (2002), `http://uk.arxiv.org/abs/nlin/0109001v3` (cit. February 212, 2012)

Gershenson, C.: Design and Control of Self-organizing Systems. CopIt ArXives, Mexico (2007) ISBN: 978-0-9831172-3-0

Janošek, M., Kocian, V.: Simulation Parameters Settings Methodology Proposal Based on Leverage Points. In: Stavrinides, G., Banerjee, S., Caglar, H., Ozert, M. (eds.) Proceedings of the 4th International Interdisciplinary Chaos Symposium Chaos and Complex Systems. Complexity series, pp. 411–414. Springer (2013) ISBN 978-3-642-33913-4

Meadows, D.H.: Leverage points: Places to intervene in a system. The Sustainability Institute (1999)

Pelánek, R.: Modelování a simulace komplexních system, 236 p. Masaryk University (2011) ISBN: 978-80-210-5318-2

Volná, E., Janošek, M., Kocian, V., Kotyrba, M., Oplatková, Z.: Methodology for System Adaptation Based on Characteristic Patterns. In: Dutta, A. (ed.) Robotic Systems - Applications, Control and Programming. InTech (2012) ISBN: 978-953-307-941-7

Wilensky, U.: NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL (1999), `http://ccl.northwestern.edu/netlogo/`

Simulation Analysis of the Complex Production System with Interoperation Buffer Stores

Bronislav Chramcov, Robert Bucki, and Sabina Marusza

Abstract. The article highlights the problem of mathematical modelling and subsequent simulation of the highly complex synthetic environment illustrating the real production system consisting of parallel manufacturing plants equipped with interoperation buffer stores. The system can be arranged optionally by means of the simulator which was created on the basis of the presented assumptions in the C# programming language. The discussed system realizes orders set by defined customers. Production control is based on heuristic algorithms which choose an order to be realized and a manufacturing plant in which the production process is carried out. The criterion is to minimize the total time of realizing orders however, as seen in the case study, also either the remaining capacity of tools after realizing all orders or the total tool replacement time can be taken into account while dealing with the problem. The modelling and projecting stages are followed by the simulation study. This simulation study is realized for the specific list of orders. It all leads to the thorough analysis of the obtained results which are later compared with the results obtained for the system without interoperation buffer stores.

1 Introduction

Manufacturing companies are currently facing very strong pressures in terms of cost, quality, flexibility, customization and a product delivery time to the defined market. Production systems which transform raw materials into high quality and highly reliable products are being developed and improved to address these needs. These systems have to be flexible and able to react to changing production capacity

Bronislav Chramcov
Tomas Bata University in Zlin, Faculty of Applied Informatics, nam. T.G. Masaryka 5555,
760 01 Zlin, Czech Republic
e-mail: chramcov@fai.utb.cz

Robert Bucki · Sabina Marusza
Institute of Management and Information Technology in Bielsko-Biała, ul. Legionów 81,
43-300 Bielsko-Biała, Poland
e-mail: rbucki@wsi.edu.pl

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 423–434.
DOI: 10.1007/978-3-319-00542-3_42

requirements. Design for manufacturing environments and production planning and scheduling in cellular manufacturing environment are critical areas [1].

Methodologies of industrial production control and production management can support today's companies in addressing the aforementioned challenges. A critical review of popular production management methodologies is presented for example in [2]. These methodologies involve frequently different optimization methods. Many optimization problems in the field of production control may be approached using heuristic and meta-heuristic techniques. Heuristics were first introduced by G. Polya in 1945 [3] and were developed later in the 70's when various heuristics were also introduced for specific purpose problems in different fields of science and techniques [4]. Heuristic algorithms are used to control the production process which is described in detail in the work [5].

It is possible to solve the problem of production design and scheduling with the use of modelling and simulation methods of such production systems. One of the most useful tools in the arsenal of an operations research (industrial engineering) management science analyst consists in computer simulation. Some common application areas of computer simulation or simulation optimization are service stations such as airports [6], call centers and supermarkets; road and rail traffic; industrial production lines [7] and logistical operations like warehousing and distribution [8, 9]. The initial specification and consequent modelling of the discussed manufacturing system is described and consequently expanded to the simulation form enabling us to carry out a simple simulation process [10]. Each simulation process is carried out by means of the simulation tool built on the basis of the specification details presented hereby and the subsequent model [11].

2 General Mathematical Model of the Complex Production System

Let us model the complex production system consisting of several manufacturing plants arranged in a parallel way. The continuous production process is carried out in each manufacturing plant in J workstations arranged in a series. We assume that there is a robot centre in each work station which performs various operations with use of I tool types. This system realizes orders set by M customers. However, further we assume that only one tool can be determined to perform the operation on the order unit in each robot centre. Operations on order elements are carried out in each robot centre in sequence. We assume there are buffer stores

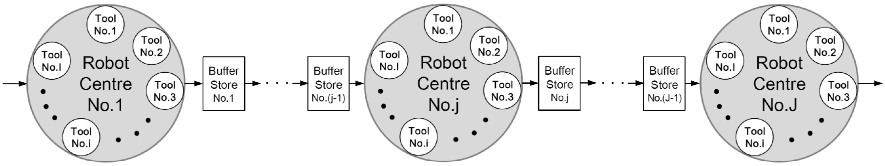


Fig. 1 The scheme of the semi-products flow in the manufacturing plant

between robot centres. The capacity of each buffer store is limited. Moreover, let us assume that there are *A* identical manufacturing plants described according to the scheme in Fig.1.

This system requires *K* stages to realize the order. The matrix of orders at the *k*-th stage is considered in the form $Z^k = \left[z^k_{m,n}\right]$ where $z^k_{m,n}$ is the number of conventional units of the *n*-th order of the *m*-th customer at the *k*-th stage. A production decision is made at the *k*-th stage. For simplicity reasons, we assume that each ordered product is made from the universal charge which is immediately supplemented. Let us define the assignment vector of orders to tools in the form $E_{m,n} = \left[e_{m,n}(j)\right]$ where $e_{m,n}(j)$ is the number of the tool in the *j*-th robot centre able to realize the order $z_{m,n}$. If the order $z_{m,n}$ does not exist in the *j*-th work station, then $e_{m,n}(j) = 0$. Let $b_{j,\alpha}$ be the buffer store between the *j*-th robot centre and the robot centre (*j*+1) in the α-th manufacturing plant. The capacity of each buffer store is calculated in numbers of the ordered semi-products $z_{m,n}$.

2.1 Life of the Tool

The base life of a new brand set of tools used to manufacture orders is given by the vector in the form $G = \left[g(i)\right]$ where *g(i)* is the base number of units which can be manufactured by the *i*-th type of tool before the tool in this robot centre is completely worn out and requires immediate replacement.

Let $\Psi(i) = \left[\psi_{m,n}(i)\right]$ be the matrix of conversion factors determining how many units of the order $z_{m,n}$ can be realized with the use of the *i*-th tool, $\psi_{m,n}(i) > 0$. If the order $z_{m,n}$ is not realized by the *i*-th tool at all, then $\psi_{m,n}(i) = -1$.

Let us now define the life matrix for realizing the order $z_{m,n}$ in the form $G(i) = \left[g_{m,n}(i)\right]$ where the life matrix element $g_{m,n}(i)$ is the number of the order $z_{m,n}$ conventional units which can be realized by means of the *i*-th tool before the tool is completely worn out. This element takes its values according to (1).

$$g_{m,n}(i) = \psi_{m,n}(i) \cdot g(i) \tag{1}$$

2.2 Capacity of the Buffers

Let us introduce the base capacity vector of buffer stores in the complex production system in the form $Gb = \left[gb(j)\right]$ where *gb(j)* is the base capacity of the *j*-th buffer store in the each manufacturing plant.

Let $\Psi b(j) = \left[\psi b_{m,n}(j)\right]$ be the matrix of conversion factors determining how many units of the order $z_{m,n}$ can be stored in the *j*-th buffer, $\psi b_{m,n}(j) > 0$. If the *n*-th order of the *m*-th customer does not exist, then $\psi b_{m,n}(j) = -1$.

Let us now define the capacity matrix for storing units of the order $z_{m,n}$ in the form $Gb(j) = \left[gb_{m,n}(j)\right]$ where the element $gb_{m,n}(j)$ is the maximum capacity of the j-th buffer store for the conventional units of the order $z_{m,n}$. This element takes its values according to (2) where the expression $\lfloor x \rfloor$ represents the floor function.

$$gb_{m,n}(j) = \left\lfloor \psi b_{m,n}(j) \cdot gb(j) \right\rfloor \tag{2}$$

2.3 State of the Tools

The state of the complex production system consisting of parallel manufacturing plants changes after every operation on the order unit. In this case, it is useful to analyse the state of tools as well as the state of buffer stores. It is possible to present the state of the tools by means of the state of tools matrix for each manufacturing plant in the form (3), where $s^k(i,j,\alpha)$ represents the value of relative tool wear ($0 \le s^k(i,j,\alpha) \le 1$) of the i-th tool in the j-th robot centre in the α-th manufacturing plant. This element takes its value according to (4) where $x^k_{m,n}(i,j,\alpha)$ is the number of the order $z_{m,n}$ units realized by the i-th tool in the j-th robot centre in the α-th manufacturing plant at the k-th stage.

$$S^k(\alpha) = \left[s^k(i,j,\alpha)\right] \text{ , } \alpha=1, \dots, A \tag{3}$$

$$s^k(i,j,\alpha) = \begin{cases} s^{k-1}(i,j,\alpha) & \begin{array}{l}\text{if no operation on the order} \\ \text{is realized by the } i-\text{th tool} \\ \text{in the } j-\text{th robot centre} \\ \text{of the } \alpha-\text{th plant at the } k\text{th stage}\end{array} \\ s^{k-1}(i,j,\alpha) + \dfrac{x^k_{m,n}(i,j,\alpha)}{g_{m,n}(i)} & \text{otherwise} \end{cases} \tag{4}$$

If the state of the tool should be exceeded ($s^k(i,j,\alpha) \ge 1$), no unit of any order can be realized in the manufacturing plant and it triggers the need to carry out the replacement process to resume the production in the discussed robot centre. If the i-th tool has to be replaced with a new one, the state of this tool changes to zero after carrying out the replacement procedure.

In addition, it is useful to monitor how many conventional units of any order can be realized with use of the specific tool. Hence, the matrix of the available capacity of the tools in the form $P^k_{m,n}(\alpha) = \left[p^k_{m,n}(i,j,\alpha)\right]$ is assumed, where $p^k_{m,n}(i,j,\alpha)$ is the number of conventional units of the order $z_{m,n}$ which still can be realized with the use of the i-th tool in the j-th robot centre of the α-th manufacturing plant. If the available capacity of the tool does not allow to realize at least one conventional unit of the order $z_{m,n}$ then the replacement process in this robot centre is carried out automatically. The available capacity of the i-th tool in the j-th robot centre of the α-th manufacturing plant for the order $z_{m,n}$ is determined in the form (5), where expression $\lfloor x \rfloor$ represents the floor function.

$$p^k_{m,n}(i, j, \alpha) = \left\lfloor g_{m,n}(i) \cdot \left(1 - s^k(i, j, \alpha)\right)\right\rfloor \tag{5}$$

It is then possible to calculate the total available capacity of the α-th manufacturing plant at the k-th stage according to the formula (6).

$$P^k(\alpha) = \sum_{i=1}^{I} \sum_{j=1}^{J} g(i) \cdot \left(1 - s^k(i, j, \alpha)\right) \tag{6}$$

2.4 State of the Buffer Stores

It is also possible to evaluate the state of the complex production system according to the capacity utilization of the buffer store. The state of the buffer stores is defined through the state matrix of buffer stores in the form (7), where $sb^k(j,\alpha)$ represents the value of relative capacity utilization ($0 \le sb^k(j,\alpha) \le 1$) of the j-th buffer store in the α-th manufacturing plant at the k-stage. In the case of accepting the strategy of realizing orders one after another which results in storing more than one type of the product in the j-th buffer store the state of the buffer store can be calculated according to (8), where $sb^k_{m,n}(j,\alpha)$ is the number of conventional units of the n-th order of the m-th customer already stored in the j-th buffer store in the α-th manufacturing plant.

$$Sb^k = \left[sb^k(j,\alpha)\right] \; ; j=, \ldots, J\text{-}1; \; \alpha=1, \ldots, A \tag{7}$$

$$sb^k(j,\alpha) = \frac{\displaystyle\sum_{m=1}^{M} \sum_{n=1}^{N} \frac{sb^k_{m,n}(j,\alpha)}{\psi_{m,n}}}{gb(j)} \tag{8}$$

If the buffer store is full (no units of any order can be stored) the robot centre before this buffer store becomes blocked.

In addition, it is useful to monitor the allowable capacity of buffer stores. Hence, let us introduce the matrix of allowable capacity of buffer store in the form (9), where $pb^k_{m,n}(j,\alpha)$ is the number of conventional units of the order $z_{m,n}$ which can be stored in the j-th buffer store in the α-th manufacturing plant. This variable takes its value according to (10), where expression $\lfloor x \rfloor$ represents the floor function.

$$Pb^k_{m,n} = \left[pb^k_{m,n}(j,\alpha)\right] \; ; j=, \ldots, J\text{-}1; \; \alpha=1, \ldots, A \tag{9}$$

$$pb^k_{m,n}(j,\alpha) = \left\lfloor gb_{m,n}(j) \cdot \left(1 - sb^k(j,\alpha)\right)\right\rfloor \tag{10}$$

2.5 The General Formula for Calculation of the Total Production Time

The time of realization one conventional unit of the *n*-th order of the *m*-th customer with the use of the *i*-th tool in the *j*-th robot center is expressed by means of variable $\tau^{pr}_{m,n}(i, j)$. Throughout the manufacturing process tools get worn out and require replacement for new ones. The replacement time of the i-th tool is represented by means of the variable $\tau^{repl}(i)$. Consequently, the total manufacturing time of all orders is calculated in accordance with the formula (11), where ΔT is the time during which elements are manufactured simultaneously.

$$T = \sum_{\alpha=1}^{A}\sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{i=1}^{I}\sum_{j=1}^{J} y'_{m,n}(i,j,\alpha)^k \cdot \tau^{pr}_{m,n}(i,j) + \sum_{\alpha=1}^{A}\sum_{k=0}^{K}\sum_{i=1}^{I}\sum_{j=1}^{J} y''_{m,n}(i,j,\alpha)^k \cdot \tau^{repl}(i) - \Delta T \qquad (11)$$

The variable $y'_{m,n}(i, j,\alpha)^k$ takes the value of one if one conventional unit of the order $z_{m,n}$ is realized with the use of the *i*-th tool in the *j*-th robot centre in the *α*-th manufacturing plant at the *k*-th stage, otherwise it equals zero. Similarly, the variable $y''_{m,n}(i, j,\alpha)^k$ takes the value of one if the replacement procedure of the *i*-th tool in the *j*-th robot centre in the *α*-th manufacturing plant at *k*-th stage is carried out, otherwise it equals zero.

3 Control of the Complex Production System

The control of the complex production systems consists in implementing heuristic algorithms which choose a manufacturing plant from the set of plants to place the order to be realized and also choose an order from the matrix of orders for manufacturing. Firstly, the multiple simulation of random choice of orders and manufacturing plants can be used. Consequently, the best solution is searched for.

Secondly, some other heuristic algorithms can be put forward. Combination of the control algorithm of either the maximal or minimal orders and the algorithm of either maximal or minimal available capacity is possible to use. Thus, the following control algorithms are implemented:

- the control algorithm of the maximal order [*ħ*(max)],
- the algorithm of the minimal order [*ħ*(min)],
- the algorithm of the maximal flow capacity of the production plant [ƛ (max)],
- the algorithm of the minimal flow capacity of the production plant [ƛ (max)].

The algorithm of maximal or minimal order chooses the order with maximal or minimal number of units for realization. The algorithm of the maximal or minimal available capacity chooses the manufacturing plant where the available capacity is maximal or minimal. It is also possible to use some production criteria for evaluation of used control algorithms. In this case, the total order realization time, the value of the remaining capacity after realization of all orders and the total tool replacement time are considered.

4 Simulation Study of the Specific Production System

The computer simulator was created in the C# programming language with the use of the .NET Framework 4 platform as well as the programming environment Microsoft Visual C# 2010 Express. It simulates the behaviour of the complex manufacturing system described in the paper by means of the assumptions and its corresponding model while implementing the required objects. Data are introduced in the matrix form which is adequate to the way presented in the detailed specification earlier in the paper. The simulator enables us to carry out calculation experiments. The experiments consist in carrying out the required number of simulations on the random basis as well as four simulation procedures for each possible combination of available heuristic algorithms. It is possible to present all final results and choosing the best solution which meets the specific criterion.

4.1 Definition of the System

The complex production system consists of 3 identical manufacturing plants arranged in a parallel way. Each plant consists of 5 robot centres arranged in a series. There are 5 tools which can perform dedicated operations in each robot centre. The tools cannot be regenerated. The simulation study is realized for the manufacturing plant equipped with either interoperation buffer stores or without buffer stores. Three customers set orders to be realized by the production system. The number of conventional units of the orders for each specific customer is specified in the matrix (12). The *n*-th order of the *m*-th customer is realized in accordance with the assignment vector. These vectors for all orders are presented in the form (13).

$$Z^0 = \begin{bmatrix} 240 & 2900 & 0 & 170 \\ 380 & 0 & 150 & 740 \\ 0 & 810 & 0 & 210 \end{bmatrix} \tag{12}$$

$$E = \begin{bmatrix} \{1,2,4,3,5\} & \{2,1,3,5,4\} & \{0,0,0,0,0\} & \{2,4,1,3,5\} \\ \{3,2,1,5,4\} & \{0,0,0,0,0\} & \{4,2,1,3,5\} & \{2,1,4,5,3\} \\ \{0,0,0,0,0\} & \{5,2,1,3,4\} & \{0,0,0,0,0\} & \{2,4,1,5,3\} \end{bmatrix} \tag{13}$$

The base life vector for a new brand set of tools is given in the form $G=[120,80,50,40,30]$ and the base capacity vector of buffer stores in each manufacturing plant is given in the form $Gb=[3,5,2,4]$. The matrixes of conversion factors for all tools and buffers are specified according to (14) where *i=1,...,5* and *j=1,...,4.*

$$\Psi(i)=\begin{bmatrix} 1 & 2 & -1 & 3 \\ 2 & -1 & 1 & 1 \\ -1 & 2 & -1 & 5 \end{bmatrix},\ \Psi b(j)=\begin{bmatrix} 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ -1 & 1 & -1 & 1 \end{bmatrix} \tag{14}$$

The times of realization of one conventional unit of the order $z_{m,n}$ in each *j*-th robot centre with the use of determined tools (according to the assignment matrix) are defined in the vectors (15). The vector of replacement times of tools is defined in the form $T^{repl}=[15,13,9,5,26]$.

$$\begin{gathered} T_{1,1}^{pr}=[23,54,10,15,21]\ ,T_{1,2}^{pr}=[23,37,42,26,33]\ , \\ T_{1,4}^{pr}=[18,23,31,34,42]\ ,T_{2,1}^{pr}=[39,41,45,16,49]\ , \\ T_{2,3}^{pr}=[16,19,21,11,27]\ ,T_{2,4}^{pr}=[11,12,20,17,13]\ , \\ T_{3,2}^{pr}=[10,16,15,22,30]\ ,T_{3,4}^{pr}=[14,27,20,35,40]\ , \end{gathered} \tag{15}$$

4.2 Results of the Simulation

The simulation study is realized for production system with two or three manufacturing plant. The manufacturing plants equipped with either interoperation buffer stores or without buffers are analysed.

All simulations are to be run for some initial values of tool wear and for the discussed control algorithms. Firstly, a random choice of orders and manufacturing plants was used. The random choice was carried out 1000 times and then the best result was searched for.

Consequently, the orders were realized by means of all combinations of discussed heuristic control algorithms. Some manufacturing criteria are used for evaluation of the implemented control algorithms.

The results of the simulation for these manufacturing criteria are shown in the following tables. Table 1 presents the results of the total order realization time. The values of the remaining capacity after realization of all orders are presented in Table 2. Table 3 shows the values of the total tool replacement time. The minimal or maximal values for each initial value of the state of tools are highlighted.

Table 1 The total order realization time of the discussed production systems with some initial values of tool wear for used control algorithms

Tool wear	Production system	Control algorithms				
		Random choice (1000 runs)	ħ(max) & ƛ (max)	ħ(min) & ƛ (max)	ħ(max) & ƛ (min)	ħ(min) & ƛ (min)
0%	2 plants + buffers	122171	122171	147877	122171	147877
	2 plants without buffers	122570	122570	158120	122570	158120
	3 plants + buffers	122171	122171	142336	122171	142336
	3 plants without buffers	122570	122570	146665	122570	146665
30%	2 plants + buffers	122180	122180	147899	122180	147899
	2 plants without buffers	122780	122780	158451	122780	158451
	3 plants + buffers	122180	122180	142345	122180	142345
	3 plants without buffers	122780	122780	146733	122780	146733
50%	2 plants + buffers	122180	122180	147884	122180	147884
	2 plants without buffers	122690	122690	158344	122690	158344
	3 plants + buffers	122180	122180	142345	122180	142345
	3 plants without buffers	122690	122690	146703	122690	146703

Table 2 The values of the remaining capacity of the discussed production systems with some initial values of tool wear for used control algorithms

Tool wear	Production system	Control Algorithms				
		Random choice (1000 runs)	ħ(max) & ƛ (max)	ħ(min) & ƛ (max)	ħ(max) & ƛ (min)	ħ(min) & ƛ (min)
0%	2 plants + buffers	1461.67	1460.67	1080.33	1460.67	1080.33
	2 plants without buffers	1461.67	1460.67	1050.67	1460.67	1050.67
	3 plants + buffers	2191.67	2110.67	1861.33	2110.67	1861.33
	3 plants without buffers	2181.67	2111.67	1981.33	2111.67	1981.33
30%	2 plants + buffers	1159.67	1158.67	1208.33	1158.67	1208.33
	2 plants without buffers	1159.67	1158.67	1118.67	1158.67	1118.67
	3 plants + buffers	1918.67	1767.67	1798.33	1767.67	1798.33
	3 plants without buffers	1918.67	1918.67	1798.33	1918.67	1798.33
50%	2 plants + buffers	821.67	820.67	930.33	820.67	930.33
	2 plants without buffers	821.67	820.67	930,67	820,67	930,67
	3 plants + buffers	1426.67	1305.67	1416.33	1305.67	1416.33
	3 plants without buffers	1416.67	1386.67	1386.33	1386.67	1386.33

Table 3 The total tool replacement time of the discussed production systems with some initial values of tool wear for used control algorithms

Tool wear	Production system	Control Algorithms				
		Random choice (1000 runs)	ħ(max) & ƛ (max)	ħ(min) & ƛ (max)	ħ(max) & ƛ (min)	ħ(min) & ƛ (min)
0%	2 plants + buffers	4783	4933	4742	4933	4742
	2 plants without buffers	4808	4944	4792	4944	4792
	3 plants + buffers	4699	4658	4642	4658	4642
	3 plants without buffers	4745	4786	4704	4786	4704
30%	2 plants + buffers	4819	5051	4932	5051	4932
	2 plants without buffers	4940	5059	5009	5059	5009
	3 plants + buffers	4756	4891	4802	4891	4802
	3 plants without buffers	4849	4950	4884	4950	4884
50%	2 plants + buffers	4821	5036	4929	5036	4929
	2 plants without buffers	4882	5023	4985	5023	4985
	3 plants + buffers	4953	4797	4774	4797	4774
	3 plants without buffers	4911	4903	4847	4903	4847

The results presented in the tables prove the immense role of buffer in manufacturing systems. They lead to minimizing the total order realizing time as well as to minimizing the total tool replacement time. In our case, practically constant value of total order realization time is achieved for all types of production systems. This is achieved by the presence of one big order element (in our case $z_{1,2}$=2900). This order later influences the total time of realization of all orders. It is possible to demonstrate this situation by means of the order realization time schedule. The order realization time schedule for the production system is presented in Fig. 2. As seen, it consists of 2 manufacturing plants equipped with buffer stores. The results also show that the defined manufacturing system should be controlled by means of the algorithm of the maximal order. If the minimal value of the total tool replacement time is to be prioritized, it is possible to use the algorithm of the minimal order. On the other hand, the use of the control algorithm of either maximal or minimal available capacity does not have any impact on the final results. It can be seen that to find a good solution, it is advisable to use the method of the random choice of orders and manufacturing plant. This method achieved for the most experiments better results than in case of implementing pairs of heuristic algorithms.

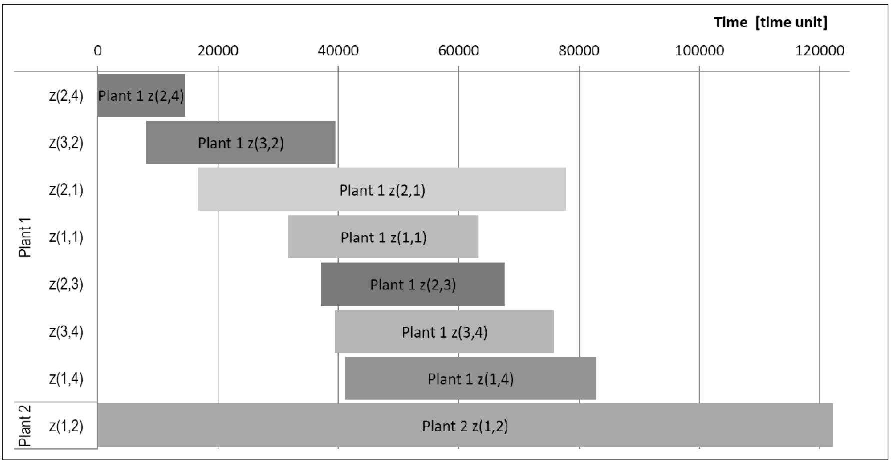


Fig. 2 The order realization time schedule for the production system consists of 2 manufacturing plants

5 Conclusions

The specification and the subsequent model of the complex production system were verified by the goals formulated in the performed simulation study. This simulation study confirms the improvements in productivity of the production system equipped with interoperation buffer stores. In addition, as seen in the tables with results, there is a need to use the simulation approach (the multiple random choice) because in this way it is possible to find better results in accordance with the stated criterion. In our case, better results were achieved for the most of the performed simulation experiments. The total replacement time was minimized slightly only in case of implementing heuristic algorithms. Nevertheless, if the list of orders is modified, the simulation approach should be carried out again in order to seek for the satisfactory solution, better than in case of implementing pairs of heuristic algorithms. These assumptions could be proved if further simulation research were carried out for other varied sets of customers' orders. Certain attention should be paid to the influence of the capacity of buffer stores in the manufacturing process to ensure its independence.

Acknowledgments. This work was supported in part by European Regional Development Fund within the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089.

References

[1] Modrák, V., Pandian, R.S.: Operations Management Research and Cellular Manufacturing Systems: Innovative Methods and Approaches, p. 439. IGI Global (2012) ISBN 978-1-61350-047-7

[2] Katsanos, E., Bitos, A.: Methods of Industrial Production Management: A Critical Review. In: Proceedings of the 1st International Conference on Manufacturing Engineering, Quality and Production Systems, vol. I, pp. 94–99. WSEAS Press, Brasov (2009)

[3] Polya, G.: How to Solve It. Princeton Univ.Pr. (1945)

[4] Pearl, J.: Heuristics: Intelligent Search Strategies for Computer Problem Solving. Add.-Wesley (1984)

[5] Bucki, R., Chramcov, B.: Modelling and simulation of the order realization in the serial production system. International Journal of Mathematical Models and Methods in Applied Sciences 5(7) (2011) ISSN 1998-0140, `http://www.naun.org/journals/m3as/` (cit. April 30, 2012)

[6] Guizzi, G., Murino, T., Romano, E.: A discrete event simulation to model passenger flow in the airport terminal. In: Proceedings of the 11th WSEAS international conference on Mathematical Methods and Computational Techniques in Electrical Engineering (MMACTEE 2009), pp. 427–434. WSEAS Press, Athens (2009)

[7] Chramcov, B., Beran, P., Daníček, L., Jašek, R.: A simulation approach to achieving more efficient production systems. International Journal of Mathematics and Computers in Simulation 5(4) (2011) ISSN 1998-0159 (cit. June 30, 2011)

[8] Bucki, R., Frąckiewicz, Z., Marecki, F.: Modelling and Simulation of Complexes of Operations in Logistic Systems. In: Proceedings of the 29th International Conference on Mathematical Methods in Economics, Janská Dolina, Slovakia, pp. 88–93 (2011)

[9] Modrák, V.: On the conceptual development of virtual corporations and logistics. In: Proceedings of the International Symposium on Logistics and Industrial Informatics, LINDI 2007, pp. 121–125. University of Applied Sciences Wildau (2007)

[10] Bucki, R., Suchánek, P., Vymětal, D.: Information Control of Allocation Tasks in the Synthetic Manufacturing Environment. International Journal of Mathematics and Computers in Simulation 3(6), 324–332 (2012) ISSN 1998-0159

[11] Marusza, S.: The Computer Simulator of the Elastic Manufacturing System with the Parallel Structure of Plants and Serial Production Work Stations. Diploma thesis. Institute of Management and Information Technology, Bielsko-Biala, p. 93 (2013)

Usage of Modern Exponential-Smoothing Models in Network Traffic Modelling

Roman Jašek, Anna Szmit, and Maciej Szmit

Abstract. The article summarized current state of our works regarding usage of exponential smoothing Holt-Winters'-based models for analysis, modelling and forecasting Time Series with data of computer network traffic. Especially we use two models proposed by J. W. Taylor to deal with double and triple seasonal cycles for modelling network traffic in two local area networks and three campus networks. We use three time series with data of TCP, UDP and ICMP traffic (given by number of packets per interval) on each network.

Keywords: Holt-Winters Models, Network Traffic Engineering, Time Series Analysis.

1 Introduction

In many areas the occurrence of atypical value (named anomaly) may indicate occurrence of undesirable phenomenon, like symptom of the disease in medicine, change the characteristics of the process in engineering or security incident in telecommunication networks. The anomaly can be defined as irregularity, something or someone that is strange or unusual, any event or measurement that is out of the ordinary regardless of whether it is exceptional or not. Anomaly Detection (AD), also called Outlier Detection, is area of statistics and computer science

Roman Jašek
Tomas Bata University in Zlin, Faculty of Applied Informatics, Czech Republic
e-mail: jasek@fai.utb.cz

Anna Szmit
Lodz University of Technology , Faculty of Management, Poland
e-mail: agorecka@p.lodz.pl

Maciej Szmit
Orange Labs, Corporate IT Security Agency, Poland
e-mail: maciej.szmit@orange.com

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 435–444.
DOI: 10.1007/978-3-319-00542-3_43

research (see e.g. [17]) related to pattern recognition. AD is an important issue in variety of applications: medicine (see e.g. 5), biosurveillance (see e.g.3), power engineering (see e.g. [15], [39]), medical informatics (see e.g. [11]), computer vision (see e.g. [13]), computer networks security (see e.g. [27]), production management (see e.g. [10]), general-purpose data analysis and mining (see e.g. [1],[12]) and others. In case of telecommunication networks the corresponding approach is named Network Behavior Anomaly Detection (NBAD) and is the part of discipline named Network Traffic Engineering. There observed phenomena in computer networks may be measured in several ways (f.e. there are three levels of measurements in papers [36] and [37]: Packet-Level Measurements, Flow-Level Measurements and Link-Level Measurements)

For AD it is important firstly to recognize a pattern of system behavior and then - secondly – to find observations that do not fit the pattern. The first task may be done by various types of modelling form classic mathematical models up to Artificial Intelligence based ones (the wide review of AI models used in anomaly detection in network traffic can be find in article 38).

The previous works of us ([21-24], [28], [33]) were according to time series modelling and forecasting using econometric methods on application of selected time series prediction models in computer networks security area and to methods of detection anomalies of network traffic (at the packet-level measurements) using confidence band-based algorithms (see e.g.: [2], [16]). The article summarized current state of our works regarding usage of Holt-Winter's-based models for modelling a few kinds of network traffic.

2 Modelling in Network Traffic Engineering

There are four main network traffic engineering tasks according the recommendations of International Telecommunication Union (see Figure 1):

- Traffic demand characterisation,
- Grade of Service (GOS) objectives,
- Traffic controls and dimensioning,
- Performance monitoring.

Main research methods of traffic engineering are modelling and computational simulation (ITU network traffic engineering recommendations review can be found in article [40], see also [28]). In particular recommendation [30] suggests usage a few curve fitting models like: Linear, Parabolic, Exponential, Logistic or Gompertz and a few smoothing models like: simple exponential smoothing, double exponential smoothing, discounted regression, Holt's method and Holt-Winters' seasonal models.

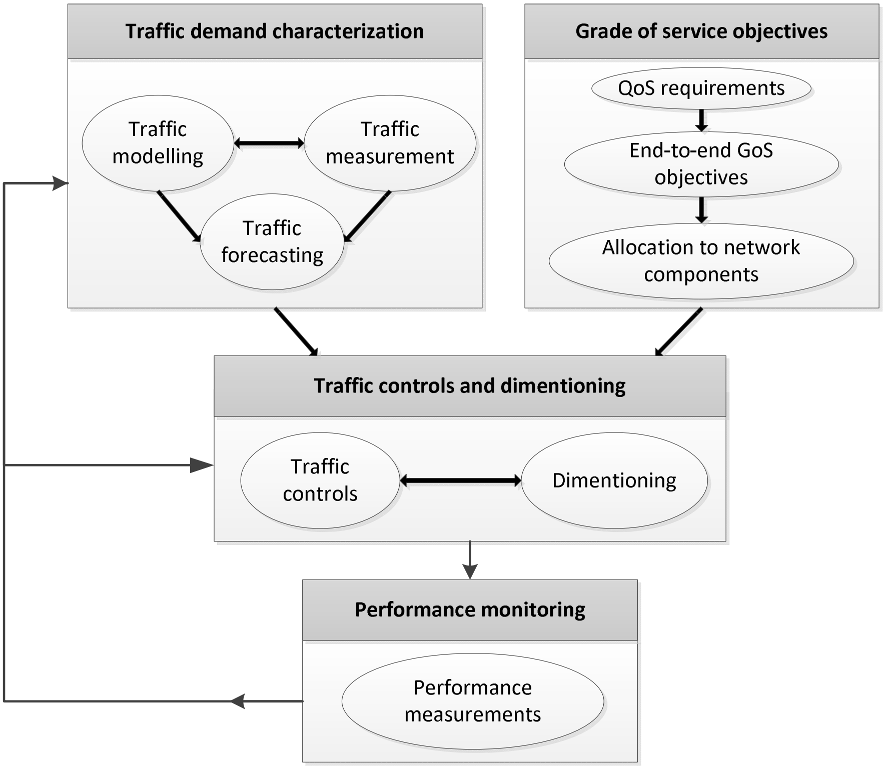

Traffic demand characterization
Traffic modelling
Traffic measurement
Traffic forecasting
Grade of service objectives
QoS requirements
End-to-end GoS objectives
Allocation to network components
Traffic controls and dimentioning
Traffic controls
Dimentioning
Performance monitoring
Performance measurements

Fig. 1 Traffic engineering tasks. Source: [29]

3 Holt-Winters and Two Taylor's Models

The basic structures of the model was provided by C.C. Holt in 1957. Holt's student Peter Winters enhanced the model by adding seasonal constituent in 1960 (see [31]). The Holt-Winters model, called also the triple exponential smoothing model, is a well-known adaptive model used to modelling time series characterized by trend and seasonality.

In the additive version of Holt-Winters model (see [11] p. 248), the smoothed variant of the y_t time series is presented as the sum of three constituents:

$$\hat{y}_t = L_t + T_t + S_{t-r} \tag{1}$$

where:

$\hat{y}_t$ is the value estimated by the model of the variable in moment t,
r is the length of the seasonal periodicity,

$$L_t = \alpha(y_t - S_{t-r}) + (1-\alpha)(L_{t-1} + T_{t-1}) \quad (2)$$

is the constituent smoothing out the level of the time series,

$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1} \quad (3)$$

represents the increase of the time series resulting from the trend,

$$S_t = \gamma(y_t - L_t) + (1-\gamma)S_{t-r} \quad (4)$$

is the seasonal component of the time series,

α, β and γ are smoothing parameters, estimated for the particular time series, while y_t is the real value of the variable in moment t and the parameters α, β and γ belong to $[0;1]$ interval.

Some extension of the Winters model was, proposed by Taylor in 2003 and 2010 (see [25]; Interesting study about stability of the model is presented in [14]), to deal with double and triple seasonal cycles. In additive version double-seasonal Holt-Winters-Taylor model is determined by equations:

$$\hat{y}_t = L_{t-1} + T_{t-1} + D_{t-r1} + W_{t-r2} \quad (5)$$

where:

r_1 is the length of the seasonal 1 periodicity,

r_2 is the length of the seasonal 2 periodicity,

$$L_t = \alpha(y_t - D_{t-r1} - W_{t-r2}) + (1-\alpha)(L_{t-1} + T_{t-1}) \quad (6)$$

is the constituent smoothing out the level of the series,

$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1} \quad (7)$$

corresponds to the increase of the series resulting from the trend,

$$D_t = \gamma(y_t - L_t - W_{t-r2}) + (1-\gamma)D_{t-r1} \quad (8)$$

is a seasonal component of the series for first seasonality period,

$$W_t = \delta(y_t - L_t - D_{t-r1}) + (1-\delta)W_{t-r2} \quad (9)$$

is a seasonal component of the series for second seasonality period.

The other model proposed by Tayor [32] (herein after referred to as second Taylor model) is given following equations (for the forecast horizon $k = 1$):

$$\hat{y}_{t+1} = l_t + d_{t-r1+1} + w_{t-r2+1} + \varphi e_t \quad (10)$$

$$e_t = y_t - (l_{t-1} + d_{t-r1} + w_{t-r2}) \tag{11}$$

$$l_t = l_{t-1} + \alpha e_t \tag{12}$$

$$d_t = d_{t-r1} + \delta e_t \tag{13}$$

$$w_t = w_{t-r2} + \omega e_t \tag{14}$$

where:

l_t is a smoothed level,

d_t and w_t are the seasonal indexes for the intraday and intraweek cycle,

α, δ, ω are smoothing parameters,

φ is an autoregressive adjustment for first-order residua autocorrelation.

Sample time series model based on the first Taylor model and its fit is shown on figure 2 and main constituents of the model (level, trend and the seasonal component of the series for the second seasonality period) is shown on figure 3.

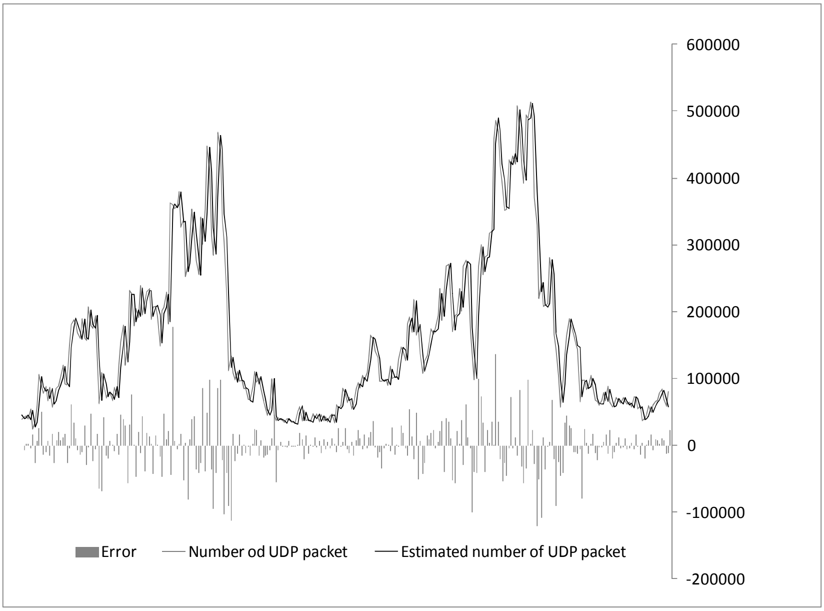


Fig. 2 Fragment of original T3 UDP time series (described below) and the value estimated by the first Taylor model. Source: own research

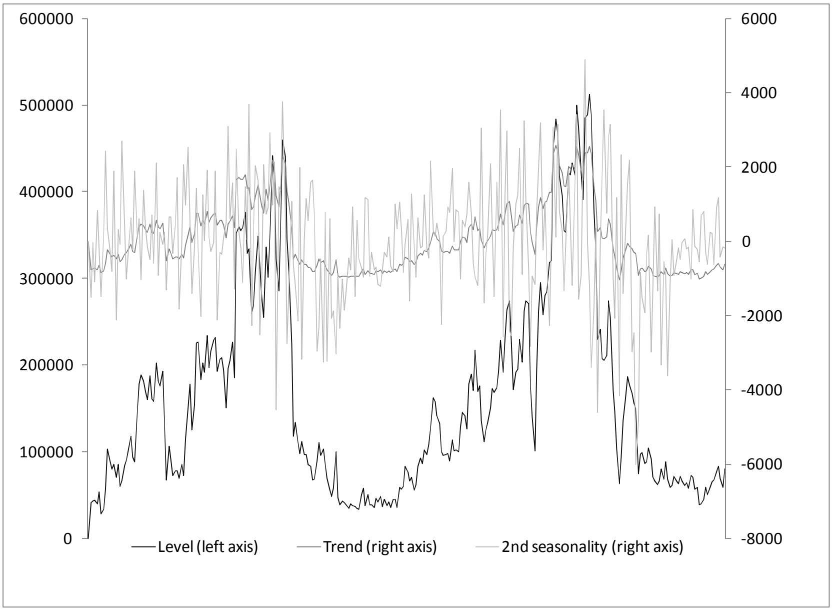


Fig. 3 Main constituents of the model from figure 2. Source: own Research.

4 Experiments

In our works we decided to collect network traffic data from a few small- and middle-sized networks, described in the Table 1. Our goal is to compare a wide spectrum of models to find the most promising for prediction of network traffic.

Model analysis was carried out for which the parameters were estimated through minimization of the expression

$$\frac{Mean\_Absolute\_Error}{Mean}$$

where:

$$Mean = \frac{\sum_{t=1}^{n} y_t}{n} \qquad (15)$$

where n is the length of the data series.

Table 1 Investigated networks description (detailed information about these networks and descriptive statistics of collected time series are described in [33]). Source: own research.

Symbol	description
W1	Amateur campus network consisting of circa 25 workstations. IDS has worked on the router which act also as the gateway to the Internet as well as a few servers (www, ftp etc.).
T2	Campus network provided by middle-size IAP (about 400 clients)
T3	A network in a block of flats; one of networks mentioned in T1, containing about 20 clients
MM	Home network connected to the campus amateur network (with maximum speed of inbound traffic set on the bandwidth manager to 4 Mbps. The home network consist of five computers and two servers protected by firewall.
II	Local Area Network in small company (about 40 computers, two intranet servers).

Like in previous research we decided to use MAE rather than MSE (Mean Squared Error) based measure because there were a lot of so-called outliers are noted in the analysed samples and MSE-based measure can be oversensitive in those cases (see e.g.: [4],[22], [23], [34], [7], [20]).

Table 2 Models fit. Source: [23], [22], own research.

Series	Protocol	Holt-Winters	Taylor model 1	Taylor model 2
W1	TCP	45,92%	45,92%	45,94%
W1	UDP	30,19%	30,19%	25,54%
W1	ICMP	31,27%	31,27%	30,72%
T2	TCP	4,19%	4, 20%	17,76%
T2	UDP	15,87%	15,87%	4,14%
T2	ICMP	8,53%	8,53%	7,47%
T3	TCP	4,11%	4,11%	3,95%
T3	UDP	15,45%	15,45%	15,53%
T3	ICMP	8,69%	8,69%	8,79%
MM	TCP	64,40%	64,47%	48,89%
MM	UDP	28,88%	30,03%	23,88%
MM	ICMP	10,57%	10,57%	10,58%
II	TCP	36,42%	36,42%	34,14%
II	UDP	49,55%	49,55%	44,86%
II	ICMP	110,65%	110,65%	71,88%

The detailed results of the research are presented in Table 2. As one can see the second Taylor model gives better fit then the first Taylor model in 10 cases and worse in 5 cases and the first Taylor model fit and Holt-Winter's model fit are almost the same.

5 Summary: Directions for the Future Research

There are a few contemporary tries to extension of Holt-Winters models as models with general seasonality (see: [8]) or robust version of the method (see e.g.: [8], [34]). As we know from our previous research (see: 24) usage of robust statistics methods can significantly improve the fit of the model so idea of using robust version of exponential smoothing models seems to be interesting.

The other issue concerns algorithms of detection anomalies, especially confidence band based ones (see: [2], [21-24]). Different ways of modelling and prediction needs different algorithms for confidence bands computing.

References

1. Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. In: ACM SIGMOD Conference Proceedings, pp. 93–104 (May 2000)
2. Brutlag, J.D.: Aberrant behavior detection in time series for network monitoring. In: Proceedings of the 14th System Administration Conference, New Orleans, Fla, USA, pp. 139–146 (2000)
3. Burkom, H.S., Murphy, S.P., Shmueli, G.: Automated time series forecasting for biosurveillance. Statist. Med. 26, 4202–4218, doi:10.1002/sim.2835
4. Chakhchoukh, Y., Panciatici, P., Bondon, P.: Robust estimation of SARIMA models: Application to short-term load forecasting. In: IEEE/SP 15th Workshop on Statistical Signal Processing, SSP 2009, August 31-September 3, pp. 77–80 (2009)
5. Chuah, M.C., Fu, F.: ECG Anomaly Detection via Time Series Analysis. In: Thulasiraman, P., He, X., Xu, T.L., Denko, M.K., Thulasiram, R.K., Yang, L.T. (eds.) ISPA 2007 Workshops. LNCS, vol. 4743, pp. 123–135. Springer, Heidelberg (2007)
6. Cordeiro, C., Neves, M.M.: Forecasting with exponential smoothing methods and bootstrap, http://www.caos.inf.uc3m.es/~jperalta/ICTSF/images/DescriptionMethodCordeiroNeves.pdf
7. Guzik, B., Appenzeller, D., Jurek, W.: Prognozowanie i symulacje. Wybrane zagadnienia, Wydawnictwo AE w Poznaniu, Poznań (2004)
8. Hanzák, T.: Holt-Winters method with general seasonality, http://ocipoez.dml.cz/bitstream/handle/10338.dmlcz/142058/Kybernetika_48-2012-1_1.pdf
9. Hanzák, T., Cipra, T.: Exponential smoothing for time series with outliers, http://ww.dml.cz/bitstream/handle/10338.dmlcz/141565/Kybernetika_47-2011-2_1.pdf
10. Hao, M.C., Keim, D.A., Dayal, U., Schneidewind, J.: Business Process Impact Visualization and Anomaly Detection, http://ivi.sagepub.com/content/5/1/15.short

11. Hauskrecht, et. al.: Evidence-based anomaly detection in clinical domains. In: Proceedings of the Annual American Medical Informatics Association (AMIA) Symposium (2007), `http://www.cs.pitt.edu/~milos/research/AMIA-07.pdf`
12. Knorr, E.M., Ng, R.T., Tucakov, V.: Distance-Based Outliers: Algorithms and Applications. VLDB Journal 8(3-4), 237–253 (2000)
13. Kratz, L., Nishino, K.: Anomaly Detection in Extremely Crowded Scenes Using Spatio-Temporal Motion Pattern Models, Department of Computer Science. Drexel University, `https://www.cs.drexel.edu/~kon/publication/LKratz_CVPR09_preprint.pdf`
14. Lawton, R.: On the Stability of the Double Seasonal Holt-Winters Method. University of the West of England, `http://forecasters.org/submissions09/LawtonRichardISF2009.pdf`
15. Wang, L., ,Zhang, R.-Q., Sheng, W., Xu, Z.-G.: Regression Forecast and Abnormal Data Detection Based on Support Vector Regression, `http://en.cnki.com.cn/Article_en/CJFDTOTAL-ZGDC200908020.htm`
16. Liu, W., Lin, S., Piegorsch, W.W.: Construction of Exact Simultaneous Confidence Bands for a Simple Linear Regression Model. International Statistical Review 76(1), 39–57, doi:10.1111/j.1751-5823.2007.00027.x
17. Markou, M., Singh, S.: Novelty detection: a review part 1: statistical approaches. Signal Processing 83, 2481–2497 (2003)
18. Maronna, R., Martin, R.D., Yohai, V.: Robust Statistics - Theory and Methods. Wiley (2006)
19. Rivlin, A.E., Shimshoni, I.: Ror: Rejection of outliers by rotations in stereo matching. In: Conference on Computer Vision and Pattern Recognition (CVPR 2000), pp. 1002–1009 (June 2000)
20. Rousseeuw, P.J., Leroy, A.M.: Robust Regression and Outlier Detection. Wiley (1987) (republished in paperback, 2003)
21. Szmit, M., Adamus, S., Bugała, S., Szmit, A.: Implementation of Brutlag's algorithm in Anomaly Detection 3. In: Federated Conference on Computer Science and Information Systems, Proceedings of the Federated Conference on Computer Science and Information Systems, pp. 685–691. PTI, IEEE, Wrocław (2011)
22. Szmit, M., Szmit, A.: Use of holt-winters method in the analysis of network traffic: Case study. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) CN 2011. CCIS, vol. 160, pp. 224–231. Springer, Heidelberg (2011)
23. Szmit, M., Szmit, A.: Usage of Modified Holt-Winters Method in the Anomaly Detection of Network Traffic: Case Studies. Journal of Computer Networks and Communications 2012, doi:10.1155/2012
24. Szmit, M., Szmit, A.: Usage of Pseudo-estimator LAD and SARIMA Models for Network Traffic Prediction. Case Studies, Communications in Computer and Information Science 291, 229–236 (2012), doi:10.1007/978-3-642-31217-5_25
25. Taylor, J.W.: Short-Term Electricity Demand Forecasting Using Double Seasonal Exponential Smoothing. Journal of Operational Research Society 54, 799–805 (2003), `http://r.789695.n4.nabble.com/file/n888942/ExpSmDoubleSeasonal.pdf`
26. de la Torre, F., Black, M.J.: Robust principal component analysis for computer vision. In: Proceedings of the Eighth International Conference on Computer Vision (ICCV 2001), pp. 362–369 (2001)

27. Vala, R., Malaník, D., Jašek, R.: Usability of software intrusion-detection system in web applications. In: Herrero, Á., Snášel, V., Abraham, A., Zelinka, I., Baruque, B., Quintián, H., Calvo, J.L., Sedano, J., Corchado, E. (eds.) Int. JointConf. CISIS'12-ICEUTE'12-SOCO'12. AISC, vol. 189, pp. 159–166. Springer, Heidelberg (2013)
28. Szmit, A., Szmit, M.: O wykorzystaniu modeli ekonometrycznych do prognozowania ruchu sieciowego, Zarządzanie rozwojem organizacji, Spała (accepted for publication, 2013)
29. ITU-T Recommendation E.490.1: Overview of Recommendations on traffic engineering, ITU (2003), http://www.itu.int/rec/T-REC-E.490.1-200301-I/en
30. ITU-T E.507 Models for Forecasting International Traffic, ITU (1998), http://www.itu.int/rec/T-REC-E.507-198811-I/en
31. Goodwin, P.: The Holt-Winters Approach to Exponential Smoothing: 50 Years Old and Going Strong, FORESIGHT Fall pp. 30–34 (2010), http://www.forecasters.org/pdfs/foresight/free/Issue19_goodwin.pdf
32. Taylor, J.W.: Exponentially Weighted Methods for Forecasting Intraday Time Series with Multiple Seasonal Cycles. International Journal of Forecasting 26, 627–646 (2010), http://users.ox.ac.uk/~mast0315/MultSeasExpWtdMethods.pdf
33. Szmit, M.: Využití nula-jedničkových modelů pro behaviorální analýzu síťového provozu, Internet, competitiveness and organizational security, TBU, Zlín (2011)
34. Gelper, S., Fried, R., Croux, C.: Robust forecasting with exponential and Holt-Winters smoothing, Leuven (2007), https://lirias.kuleuven.be/bitstream/123456789/120456/1/kbi_0718.pdf
35. Szmit, M., Szmit, A., Adamus, S., Bugała, S.: Usage of Holt-Winters Model and Multilayer Perceptron in Network Traffic Modelling and Anomaly Detection. Informatica 36(4), 359–368
36. Münz, G.: Traffic Anomaly Detection and Cause Identification Using Flow-Level Measurements, TUM, Müchen (2010), http://www.net.in.tum.de/fileadmin/TUM/NET/NET-2010-06-1.pdf
37. Wang, Y.: Statistical Techniques for Network Security: Modern Statistically-Based Intrusion Detection and Protection. IGI Global (2009)
38. Palmieri, F., Fiore, U.: Network anomalny detection though nonlinear analysis. Computers and Security 29, 737–755 (2010)
39. Grzenda, M., Macukow, B.: Heat Consumption Prediction with Multiple Hybrid Models. In: Omatu, S., Rocha, M.P., Bravo, J., Fernández, F., Corchado, E., Bustillo, A., Corchado, J.M. (eds.) IWANN 2009, Part II. LNCS, vol. 5518, pp. 1213–1221. Springer, Heidelberg (2009)
40. Villén-Altamirano, M.: Overview of ITU Recommendations on Traffic Engineering. Paper presented in the ITU/ITC workshop within 17th ITC

On Approaches of Assessment of Tribo Data from Medium Lorry Truck

David Valis, Libor Zak, and Agata Walek

Abstract. The paper deals with application of selected analytical methods in order to analyse field data from heavy off-road military vehicles. The information from the engine oil are interpreted in form of polluting particles like particles from wear process (e.g. Fe, Pb, Cu, etc.) and particles from oil deterioration itself (like Mn, Si, Zn, etc.). These particles can give us information both about system state and about oil state. We have reasonable set of oil data from field operation available. Based on the data we assume being able to determine the system condition and propose some changes (e.g. in residual operation life, in maintenance modifications in the intervals, in mission planning, etc.). Selected methods like regression analysis and fuzzy inference system are used for the data assessment.

Keywords: Field data assessment, off-line diagnostics, inputs to maintenance optimization, tribo-diagnostics, regression analysis, fuzzy logic, residual life estimation.

1 Introduction

The growing dependability and operation safety requirements for modern equipment together with the increasing complexity and continuous attempts to reduce operation and maintenance costs might be satisfied among others by the consistent use of modern diagnostic systems. The main task of object technical state diagnostics is not only to find out incurred failures, but also to prevent from the failure occurrence with the help of sensible detection and changes localization in the

David Valis
University of Defence, Brno, Czech Republic
e-mail: david.valis@unob.cz

Libor Zak · Agata Walek
Brno University of Technology, Brno, Czech Republic
e-mail: zak.l@fme.vutbr.cz, waleka@feec.vutbr.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 445–454.
DOI: 10.1007/978-3-319-00542-3_44

object structure and in its behaviour changes. Many various approaches have been published on system diagnostics and CBM (Condition Based Maintenance).

A tribotechnical system (TTS), friction in it, wear and lubrication, and especially the outcomes of it are the subjects of our major concern. We would like to analyse the outcomes from technical diagnostics of TTS. There exists wide range of data from the TTS which are not analysed further. We find this is a pity. Our main objective is to extract maximum information from the diagnostic of TTS in order to gain tools for optimising: maintenance, cost-benefit processes, operation and mission planning. The authors will apply selected mathematical tools to get some inputs into previously mentioned areas. Regarding the tribotechnical system, the basic information about tribological process, operating and loss variables are provided [1-3].

Owing to the TTS we have got a lot of diagnostic oil data. In view of tribodiagnostics this data is considered to be the final outcome. This data can tell us a lot about lubricants / life fluids quality itself as well as about system condition. Such data are very valuable. System operation, taking the oil samples and the outcomes themselves, are very fuzzy therefore we use approaches from the fuzzy logic theory. The procedure and results presented below are based on standard mathematical principles – a regression function and a regression analysis and fuzzy logic. From both presumptions we can expect reasonable costs savings. As from the military point of view we would like to determine remaining residual life to be able to perform a mission. Following the regression analysis it is possible among others to assess the operating history of an observed vehicle.

2 Objects of Diagnostics and Methods

The assumed objects of diagnostics in our case the medium lorry T810 engines have not been ready yet in terms of design to use the ON-LINE system, though in practice similar possibilities for other applications have already existed. It results from the information stated above that we are still supposed to use OFF-LINE engine diagnostics system when sampling lubrication fluid at certain intervals, and using known and optimised special tribodiagnostic methods [4]. In our case we use the results and information from atomic emission spectrometry. Following this analysis we can obtain the information about the presence of the elements of a specific kind and the amount of elements. When evaluating data, the information is transformed many times and provides only estimated reality which might be different from reality itself. If the vagueness in classes distribution is not given by a stochastic character of measured characteristics but by the fact that the exact line among states classes does not exists, it will be later on good to use fuzzy set theory and adequate multi-criteria fuzzy logic. However, we cannot identify their real origin – e.g. as a result of fatigue, cutting or sliding. Therefore in our further research we try to identify where the elements might come from. We base our assumptions on idea to increase the potential for maintenance optimisation inputs and cost benefit analysis inputs.

3 Oil Field Data Assessment and Mathematical Model

Having enough statistically important set field data obtained from the diagnosed objects. It fulfils the basic assumption that we might be capable to solve this problem successfully. Since the data sets are very extensive, we are not going to introduce them here except for a part/example of ferrum particles representing the sample of T810 – it is presented in Table 1. We deal with dozens of samples taken and analysed at different types of observed engines. In certain aspects we consider the engine from an infantry fighting vehicle to be a reference object, because the event of a failure type has occurred in it. All tribodiagnostic processes related to the failure occurrence have been recorded. We assume having potential for system residual life estimation based on the field data course. The both approaches based on regression and FIS (Fuzzy Inference System) are presented below.

Table 1 Input data of Fe particles

Sample/Mh	Fe particles (ppm)	Sample/Mh	Fe particles (ppm)
1/0	17.57	26/233	21.81
2/8	20.88	27/244	18.31
3/11	15.77	28/255	23.27
4/22	19.58	29/259	21.91
5/26	20.53	30/269	24.41
6/35	12.73	31/271	20.98
7/46	15.84	32/283	23.57
8/57	16.41	33/294	18.63
9/64	23.15	34/305	18.09
10/72	23.94	35/316	22.05
11/84	20.86	36/327	22.66
12/95	17.59	37/331	22.89
13/106	19.52	38/341	10.34
14/109	19.15	39/351	19.02
15/119	25.83	40/363	13.96
16/136	24.07	41/374	12.55
17/146	21.82	42/383	11.63
18/153	19.66	43/395	15.47
19/164	18.53	44/402	24.03
20/175	23.31	45/416	23.71
21/179	24.03	46/427	15.93
22/188	27.95	47/438	11.53
23/200	20.12	48/445	13.59
24/211	19.57	49/453	12.45
25/222	22.84	50/464	12.61

3.1 Utilization of Regression Model

In this paper we present the final outcomes of regression functions utilization to try to describe the data precisely. We concentrate on the Fe particles only and regarding the one vehicle engine type (T810). Therefore dependencies like linear, parabolic and base function – square root plus confidence intervals in all instances will be applied. Exploring and analysing variable dependencies, the values of which are obtained when performing an experiment, is considered to be an important statistical task. In view of their random character, a random vector X = ($X_1,\ldots,X_k$) represents independent variables and a dependent variable is represented by a random variable Y.

When describing and examining the dependence of Y on X, we use a regression analysis, and this dependence is expressed by the following regression function:

$$y = \varphi(\mathbf{x}, \boldsymbol{\beta}) = E(Y \mid \mathbf{X} = \mathbf{x})_{,} \tag{1}$$

where $\mathbf{x} = (x_1,\ldots,x_\mathrm{k})$ is vector of numerical variables, y is a dependent variable, $\boldsymbol{\beta} = (\beta_1,\ldots, \beta_\mathrm{m})$ is vector of regression coefficients β_j.

For our data we will look for a regression function in a linear form and we will apply a linear regression model:

$$y = \sum_{j=1}^{m} \beta_j f_j(\mathbf{x}) \quad , \tag{2}$$

where $f_\mathrm{j}(\mathbf{x})$ are well-known functions where $\beta_1,\ldots, \beta_\mathrm{m}$ are not involved.

For the data we will select gradually the following regression functions:

- $m=2, f_1(x)=1, f_2(x)=x$, regression function: $y=\beta_1+\beta_2 x$
- $m=3, f_1(x)=1, f_2(x)=x, f_3(x)=x^2$, regression function: $y=\beta_1+\beta_2 x+\beta_3 x^2$
- $m=2, f_1(x)=1, f_2(x)=x^{1/2}$, regression function: $y=\beta_1+\beta_2 x^{1/2}$

The coefficient of determination (R^2) will show its suitability for approximation / data spacing with a relevant regression function. With the coefficient getting bigger, the regression analysis reflects the assessed data better. The form of the coefficient of determination calculation is as follows:

$$R^2 = 1 - \frac{S^*_{\min}}{\sum_{i=1}^{n} y_i^2 - n(\bar{y})^2}, \text{ where } S^*_{\min} = \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{m} b_j f_{ji} \right)^2 . \tag{3}$$

where $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$, b_j is point estimation of β_j and $f_{ji} = f_j(x_i)$. The outcomes from the regression analysis for group of vehicles of the same type are presented below in figures 1-3.

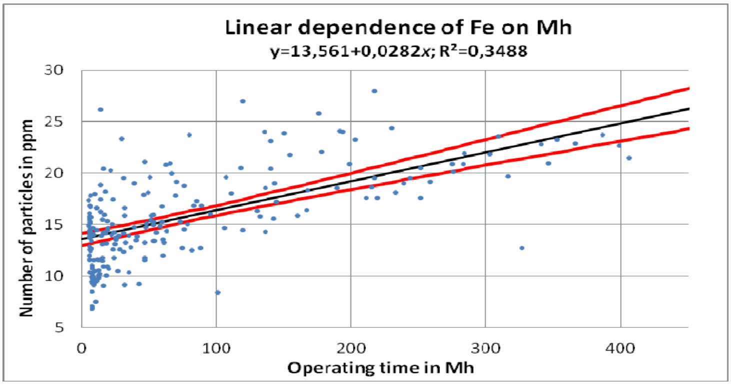


Fig. 1 Linear dependence of Fe particles course (for group of vehicles) on operating time in Mh

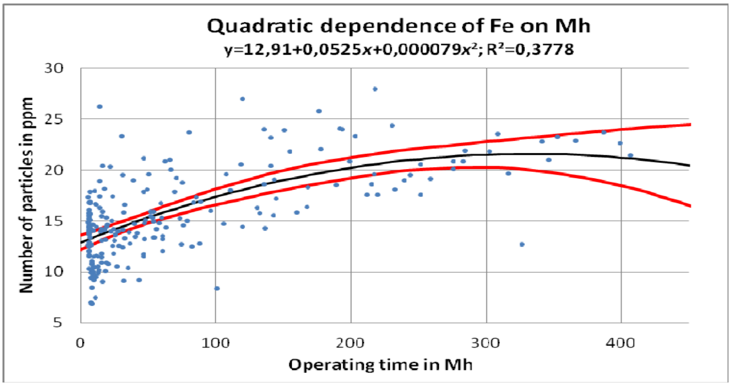


Fig. 2 Quadratic dependence of Fe particles course (for group of vehicles) on operating time in Mh

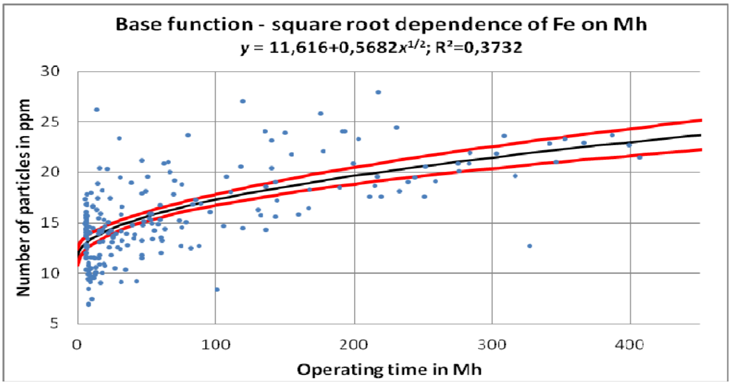


Fig. 3 Base function dependence of Fe particles course (for group of vehicles) on operating time in Mh

3.2 Utilisation of Fuzzy Inference System (FIS) and Comparison with Regression Approach

A Fuzzy Inference System (FIS) is based on the terms fuzzy *set* and fuzzy *relation* which were introduced by Lotfi A. Zadeh in 1965. The fuzzy set is one of the possible generalizations of the term set. The fuzzy set is a pair ($\mathbf{U}$, μ_A) where $\mathbf{U}$ is a universal set and μ_A: $\mathbf{U} \rightarrow \langle 0,1 \rangle$ is a membership function assigning the elements from U to fuzzy set A. The membership is marked with $\mu_{\mathrm{A}}(x)$.

Nowadays one of the most widely used applications is a Fuzzy Inference System – FIS (once used as a term “fuzzy regulator”). Two basic types of the FIS are used, and they are Mamdami and Sugeno [5, 6]. Each FIS consists of input and output variables and the FIS rules. For each FIS we specify:

- the number of input and output variables,
- for each input and output the number of predefined values (linguistics values) in the form of fuzzy sets,
- FIS rules described by predefined values.

We do not often expect a fuzzy set to be the FIS output, but we wish to get a single value $z_0 \in Z$, i.e. we want to defuzzify the FIS output. The centroid method is one of the most frequently used defuzzication methods. The FIS specified this way is called *Mamdani* FIS [5].

If we do not know how the process works (i.e. the FIS rules cannot be set), but the sufficient amount of input and output data is available, we can use the modification of Mamdani-FIS Sugeno (Takagi-Sugeno FIS) [5].

When looking for the FIS correlation between output values and input ones as for an unknown process, the method used a lot more frequently is a Sugeno FIS method which is in fact a Mamdani FIS modification. In order to find a relevant FIS, we use the data that serves as a background for the input and output values of the process. In most cases these values are a subset of real numbers, and therefore the inputs and outputs are in a numerical form. The input variables are similar to Mamdani FIS. The output variables Z_j are in constant or linear forms.

$$Z_j = \alpha_j \text{ or } Z_j = \alpha_j + \beta_{1,j} x_1 + \beta_{2,j} x_2 + \ldots + \beta_{n,j} x_n, \tag{4}$$

where α_j, $\beta_{i,j}$ $i = 1, 2, \ldots, n, j = 1, 2, \ldots, k$ are suitable constants, k is the number of rules in the FIS model, and n-tuple (x_1, x_2,…, x_n) consists of n input variables to the FIS (model). Sugeno FIS output is the values weighted average Z_j where the weight is obtained by comparing the input (x_1, x_2,…, x_n) with predefined input values [6].

To find a suitable Sugeno FIS, which describes the selected data, it is appropriate to divide the data into tuning and checking data. We find the FIS that corresponds to the tuning data best. The tuning part of data is divided into smaller parts, and predefined input (output) values and the rules describing relationship between relevant inputs and outputs are assigned to each part. There are two basic ways of dividing the data:

- dividing the area (which includes turning data) into smaller parts. A fuzzy set is assigned to each part, and their combination is used for creating rules.
- applying clustering methods to find clusters in data. One rule is made for each cluster.

After selecting the number of fuzzy sets (linguistics values) and rules, we search for appropriate parameters (α_j, $\beta_{i,j}$) using output variables Z_j. These parameters were found through a neural network. The tuning itself results in setting parameters for the FIS to describe assigned tuning data as well as possible. The accuracy is verified by calculating the output values from the test data by the FIS, and then they will be compared with the original output of the test data. The design, tuning and selection of the FIS were performed in MATLAB (Version 5.3) – FuzzyToolbox.

We concentrated on quadratic and base function courses only – as presented above – when looking for correlation with the fuzzy models results. The outcomes are presented in figures 4 and 5.

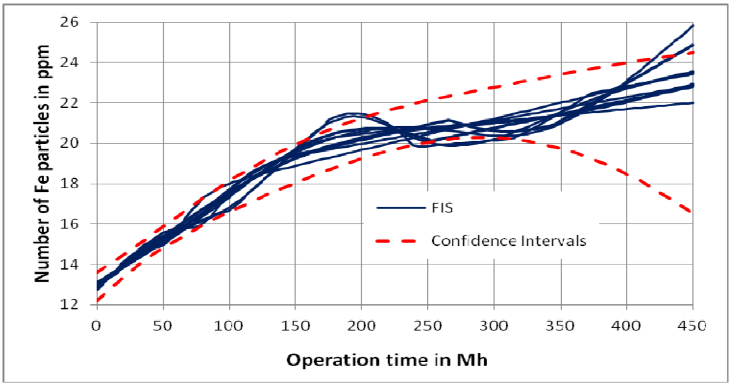


Fig. 4 Comparison of quadratic Fe course and fuzzy model

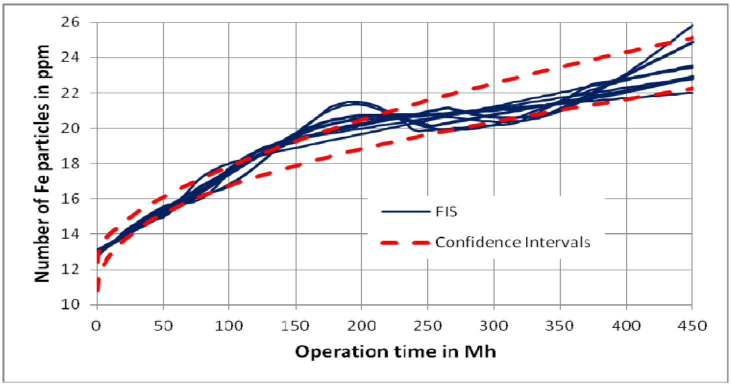


Fig. 5 Comparison of base function Fe course and fuzzy model

4 Summary

It is remarkable that the Fe particles generation based on oil field data might have more likely both quadratic and first base function course. This co-relation outcome is based on the analysis performed above and with using the regression and fuzzy approach.

Moreover, we do have available data from similar engine life test. These data and engine is for us taken as reference item for further analysis. Therefore we decided to try to estimate the residual life based on the reference engine and all of these outcomes. We believe that such capability in “reading the diagnostic data” might help in mission planning, maintenance optimisation or e.g. in cost benefit analysis. Some proposals are mentioned e.g. in [7-17]. The residual life based on Fe particles generating for T810 was analytically estimated for the interval approximately from 1167Mh to 2783 Mh. Therefore based on the analysis real values of the actual oil change in fact are always believed to be less than 45% of available oil capacity. The graphical presentation is shown in figure 6.

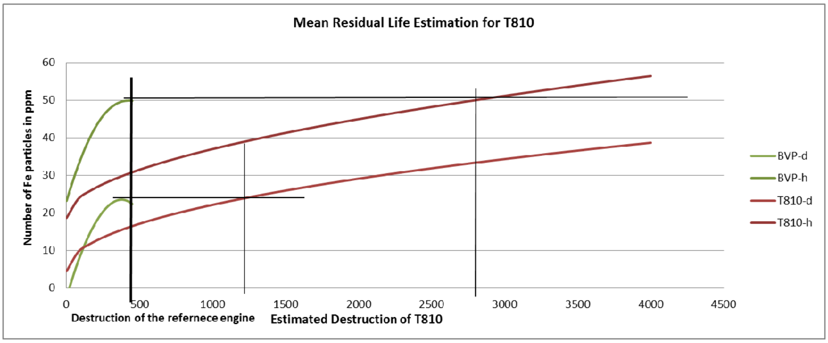


Fig. 6 Graphical estimation of mean residual life of T810 based on Fe particles

The authors have broadened the possibilities of extracting and utilization of pieces of information from TTS diagnostics. Although the regression analysis and FIS are common mathematical tools they have never been applied for analysis of TTS oil data. The potential of the TTS oil data is incredible. The authors describe just small portion of the capabilities. Application of FIS supports our idea of describing some data generation course by selected regression forms. The authors present capabilities of analysis results for the system residual life estimation (RLE). For the RLE the regression correlations of observed systems are applied. Using the TTS oil data from reference engine it is found that RLE might be determined for the other similar units. This approach will be further extended and developed into more precise RLE approaches.

5 Conclusion

In this paper we were looking for dependencies among measured values using statistical and fuzzy methods. While have applied two quite specific approaches therefore the comparison of acquired results is reasonable. At the beginning we did not know the exact theoretical background of the possible dependence of Fe particles occurrence on operating time. We were looking for this dependence using suitable approximate methods. The regression analysis and the Fuzzy Inference System serve as starting methods. When dealing with the regression analysis, it is necessary to choose a regression analysis form in advance. If the regression analysis form cannot be deduced theoretically, it is necessary to select relevant regression functions and then compare them with the measured data. In the case of fuzzy methods we did not have to select the form of an expected function, but determine the form and the amount of language values. They are important to set the Fuzzy Inference System. For different forms and amount of language values we obtain different dependence forms and shapes. It follows from the other results of other sets of field data obtained from different vehicle types that when combining properly both methods, we can find out that the dependence of measured data correspond with a real process. When dealing with the assessed data, it is advisable to use the FIS first, and then, following the form of a found dependence, select a relevant regression function. Despite taking a different analytical approach when applying single methods, the results are very similar to each other (see e.g. figures 4 and 5). It can be then assumed that the processed dependencies can be used for describing expected development under real operating conditions. The obtained results will be used for further research, e.g. the optimizing of maintenance procedures, mission planning, or the estimation of residual operating units, etc.

Acknowledgements. This paper has been prepared with the great support of the Project for Institutional Development of K-202 University of Defence, Brno and the research project no. 3, „Management Support of Small and Middle-Sized Firms Using Mathematical Methods" of Academy Sting, Business College in Brno.

References

1. Gesellschaft für Tribologie e.V. GfT Arbeitsblatt 7: Tribologie - Verschleiß, Reibung, Definitionen, Begriffe, Prüfung (GfT, Moers) (2002) (in German)
2. Czichos, H., Habig, K.-H.: Tribologie-Handbuch; Reibung und Verschleiß, 2nd edn. Vieweg, Weisbaden (2003) (in German)
3. Koucky, M., Valis, D.: Suitable approach for non-traditional determination of system health and prognostics. Zeszyty naukowe. Styczen – Marzec 1(159), 123–134 (2011) ISSN 1731 – 8157
4. Lippay, J.: Tribological diagnostics of heavy of road lorries Tatra 815 engines which operate with OA-M6 ADS II oil. Inauguration Thesis. Military Academy Brno (1991) (in Czech)

5. Mamdani, E.H.: Applications of fuzzy logic to approximate reasoning using linguistic synthesis. IEEE Transactions on Computers 26(12), 1182–1191 (1977)
6. Sugeno, M.: Industrial applications of fuzzy control. Elsevier Science Pub. Co. (1985)
7. Vališ, D., Koucký, M., Žák, L.: On approaches for non-direct determination of system deterioration. Eksploatacja i Niezawodnosc – Maintenance and Reliability 14(1), 33–41 (2012) ISSN 1507-2711
8. Quigley, J., Walls, L.: Trading reliability targets within a supply chain using Shapley´s values. Reliability Engineering and System Safety 92(10), 1448–1457 (2007)
9. Revie, M., Bedford, T., Walls, L.: Supporting Reliability Decisions During Defence Procurement Using a Bayes Linear Methodology. IEEE Transactions on Engineering Management 58(4), 662–673 (2011)
10. Rak, J., Pietrucha, K.: Risk in drinking water quality control. Przemysl Chemiczny 87(5), 554–556 (2008)
11. Stodola, P., Jamrichova, Z., Stodola, J.: Modelling of Erosion Effects on Coating of Military Vehicles Components. Transactions of Famena 36(3), 33–44 (2012)
12. Stodola, J., Stodola, P.: Tribology´s Contribution to Efficient Maintenance of Military Engines. In: ICMT 2009, Oprox, Brno (2009)
13. Stodola, J., Stodola, P.: Mechanical System Wear and Degradation Process Modelling. Transactions of Famena 34(4), 19–32 (2010)
14. Stodola, J., Stodola, P.: Operation Reliability and Diagnostics of Mechanical Systems. Transactions of Famena 33(1), 47–56 (2009)
15. Jodejko-Pietruczuk, A., Mlynczak, M., Zajac, M.: Assessment of economical lifetime of heavy-duty machines, Case study. Reliability, Risk and Safety: Theory and Applications 1-3, 531–534 (2010) ISBN 978-0-415-55509-08
16. Bartlett, L.M., Hurdle, E.E., Kelly, E.M.: Intergrated System Fault Diagnostics Utilising Diagraph and Fault Tree-based Approaches. Reliability Engineering and System Safety 94(9), 1371–1380 (2009)
17. Edleston, O.S.S.T., Bartlett, L.M.: A Tabu search algorithm applied to the staffing roster problem of Leicestershire police force. Journal of the Operational Research Society 63(4), 489–496 (2012)

Energy and Entropy of Fractal Objects: Application to Gravitational Field

Oldrich Zmeskal, Michal Vesely, Petr Dzik, and Martin Vala

Abstract. Various different approaches to the definition of entropy and their connections with fractal dimensions of systems were described in the paper ***Entropy of Fractal Systems*** presented at the conference **Nostaradamus 2012**. In the second part of the paper, the described findings were applied to study the fractal properties of image structures.

Further development is going to be presented in this paper. Conclusions of general fractal theory will be applied to the general fractal systems represented by elements (elementary particles) having fractal structure. An typical example may include the space and time distribution of mass and electric charge, i.e. the general energy. The properties of fractal fields of these quantities (gravitational, electric or other field) can be described by means of fractal geometry generally at E-dimensional space, where $E = 0, 1, 2, 3, \ldots$ The ***density of energy*** and ***entropy*** of these fractal elements will be also determined from the distribution of their quantity, field intensity and potential.

1 Introduction

The basic properties of fractal structures of the world were published recently [1]. Fractal dimension and fractal measure are the crucial parameters for such description [2]. The fractal basis was used for the study of fundamental physical laws. The complex description of conservative fractal fields in the Euclidean space was also presented. The physical quantities used for the description of different properties of physical reality (gravitational, electric, thermal, acoustic, etc.) were defined. The mathematic of fractal-Cantorian geometry was used for the description of properties of the gravitational and electrostatic field of systems with different mass and charge distribution [3], electric properties of semiconductors [4],

Oldrich Zmeskal · Michal Vesely · Petr Dzik · Martin Vala
Faculty of Chemistry, Brno University of Technology, Czech Republic

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 455–465.
DOI: 10.1007/978-3-319-00542-3_45

thermal properties of bodies [5] and properties of high-energy elementary particles following El Naschie's *E*-infinite theory [6]. Fractal theory generally defined for *E*-dimensional Cantorian space was applied to description of fractal structures in pseudo-Cantorian spaces as is space time [7].

The summary of the findings mentioned in the foregoing articles, their completion and extension of other facts is going to be the subject of this paper. Main attention is going to be paid to the study of energy and entropy of fractal systems, and also in terms of their distribution.

We are going to show that simple mathematical methods can be used to describe physical processes; the results are consistent with current knowledge. That approach, however, allows the extension of the knowledge obtained from experience in space that we perceive with our senses (i.e. space-time) into space with a general number of space and time dimensions.

2 Physical Fields of Fractal Objects

The history of fractals geometry was founded in the 17th century. At the beginning, it was a mathematical discipline that dealt with mathematical objects with fractional character. Many mathematicians published exemplary subsets, which had unusual properties and are now recognized as fractals.

One of the first applications of fractal geometry was realized by Lewis Fry Richardson, who studied the relation between the probability of two countries going to war and the length of their common border. As a part of his research, Richardson investigated how the measured length of a border varies as the unit of measurement is changed. He published empirical statistics which led to this mutual relationship [8]. This research was appreciated by mathematician Benoît Mandelbrot who worked on a wide range of mathematical problems, including mathematical physics, but who is best known as the father of fractal geometry [2]. While Richardson had been collecting data, he realized that there was considerable variation in the various published lengths of international borders. He found that the measured length $L(\varepsilon)$ of border or number of measured parts $N(\varepsilon)$ depends on the size of measure

$$L(\varepsilon) = K\varepsilon^{1-D}\,,\ N(\varepsilon) = L(\varepsilon)/\varepsilon = K\varepsilon^{-D} \tag{1}$$

where ε measure size and K is a number of measured parts for $\varepsilon = 1$. Parameters D and K was named by Mandelbrot as the fractal dimension and fractal measure in 1975 [9]. Later, these findings were applied and verified on the model of two-dimensional and three-dimensional structures. The relationships (1) of these structures can be rewritten to the form

$$n(\varepsilon) = K\varepsilon^{E-D}\,,\ N(\varepsilon) = n(\varepsilon)/\varepsilon^{E} = K\varepsilon^{-D}\,, \tag{2}$$

where $n(\varepsilon)$ is the coverage (density of quantity) of E-dimensional Euclidean space E by fractal structure [1]. This coverage depends on the size scale which is used to measure and units depend on the dimension of the space E, where is a fractal structure defined. Coverage mean e.g. length of fractal structure (for $E = 1$), this area (for $E = 2$), or volume (for $E = 3$).

2.1 Potential and Intensity of the Fractal Field

For physical structures (e.g. mass density, density of electric charge or density of heat radiation), the mathematical term in Eq. (2) can be rewritten to the form

$$\rho(r) = c_{\mathrm{N}} K r^{D-E} , \tag{3}$$

where c_{N} is the an elementary quantity (e.g. mass unit, elementary charge or Boltzmann constant), see Table 2 and $r = 1/\varepsilon$ is size of analyzed area.

It is possible to define the intensity of the physical field $\boldsymbol{F}$ by applying the Gauss-equation in E-dimensional space (right equation is for radial field)

$$\operatorname{div} \boldsymbol{F} = c_{\mathrm{V}}\, \rho(r) , \quad \frac{D}{D-E+1} \frac{\mathrm{d} F_r}{\mathrm{d} r} = c_{\mathrm{V}}\, \rho(r) , \tag{4}$$

where c_{V} is material constant (e.g. gravitational constant, permittivity or Planck constant)

The intensity of physical field $\boldsymbol{F}$ and potential V are interrelated by the equation

$$\boldsymbol{F} = -\operatorname{grad} V , \ \operatorname{div} \boldsymbol{F} = -\operatorname{div} \operatorname{grad} V = -\Delta V , \tag{5}$$

where Δ is the Laplace operator, so both quantities are connected with the density of fractal quantity $\rho(r)$ by means of relation (right equation is for radial field)

$$\Delta V = -c_{\mathrm{V}} \rho(r) , \quad \frac{D}{D-E+1} \frac{\mathrm{d}^2 V_r}{\mathrm{d} r^2} = -c_{\mathrm{V}} \rho(r) . \tag{6}$$

From the density of quantity $\rho(r)$ we can determine the radial field intensity E_{r} and corresponding potential V_{r} of a physical field with using (4) and (6)

$$F_{\mathrm{r}} = \frac{c_{\mathrm{N}} c_{\mathrm{V}} K}{D} r^{D-E+1} , \ V_{\mathrm{r}} = -\frac{c_{\mathrm{N}} c_{\mathrm{V}} K}{D(D-E+2)} r^{D-E+2} \tag{7}$$

on the dimension r of elementary cell. The last equations together with Eq. (3) are very important for describing fractal physical fields not only with radial configuration but also in general cases. For a 3-dimensional space ($E = 3$), these equations then describe the spherical field for a point source of physical quantity ($D = 0$), the cylindrical field for line source ($D = 1$), the planar field for surface source ($D = 2$) and finally the volume field for homogeneously distributed sources ($D = 3$).

2.2 Density of Energy and Entropy of the Fractal Objects

The density of energy can be derived from the density of quantity (3) and from the potential (or intensity) of physical field (7)

$$w(r) = \rho V_r = c_{\mathrm{V}} c_{\mathrm{N}}^2 \frac{K^2 r^{2(D-E+1)}}{D(E-D-2)}\,, \tag{8}$$

A special case happens when the effective density of energy is for equipotential field

$$w_0 = -\frac{F_{\mathrm{r}}^2}{c_{\mathrm{V}}} = -c_{\mathrm{V}} c_{\mathrm{N}}^2 \frac{K^2 r^{2(D-E+1)}}{D^2}\,, \tag{9}$$

Then the frequency ν (wave length λ, respectively) and the fractal dimension D are connected by the following equation [5]

$$\alpha_{\mathrm{w}} = \frac{w}{w_0} = \frac{\nu}{\nu_0} = \frac{\lambda_0}{\lambda} = \frac{D}{D-E+2}\,, \tag{10}$$

where α_w is the so called inverse coupling constant of energy density, ν_0 is the frequency for effective density of field and λ_0 is corresponding wave length.

The entropy of fractal systems can be derived from the Rényi entropy of q order

$$S_q(X) = \frac{1}{1-q} \ln \sum_i^m p_i^q\,, \tag{11}$$

where X is a discrete random variable, p_i is the probability of the event $\{X = x_i\}$. If the probabilities are all the same then all the Rényi entropies of the distribution are equal, with $S(X) = \ln m$, where m is the number of repetitions of the reduced fractal pattern [10].

By comparing of this equation with the definition of fractal dimension (2)

$$D = -\frac{\mathrm{d} \ln N(\varepsilon)}{\mathrm{d} \ln \varepsilon} \approx \frac{\ln m}{\ln r} \tag{12}$$

we will get the dependence between the entropy and fractal dimension

$$S(r) = D \ln r\,. \tag{13}$$

3 Application to Real Physical Fields

Many physical and other experiments can be described using of these power dependences. Distributions of mass density, electric charge density and specific heat at space represent a few typical examples (Fig. 1).

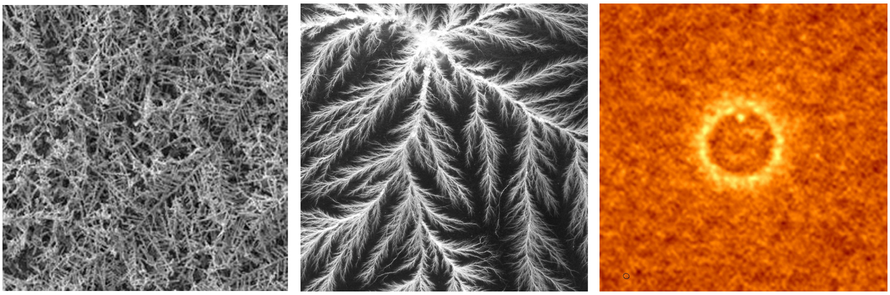


Fig. 1 An example of fractal distribution of mass (silver nitrate fractal trees for solar technology, [11]), electric charge (Lichtenberg figure electron tree in PMMA, [12]), and thermal field (thermal noise - model, [13])

3.1 Model Fractal Structure

The aim of this chapter is to determine the fractal parameters of real structures (fractal distributed mass, fractal distributed charge and fractal distributed temperature). Using the HarFA software, fractal parameter of structures from Fig.1 was determined. The wavelet analysis (modified box counting method) was used [10].

First, the two-dimensional (2D) threshold analysis of partial image crop (256×256 pixels) was performed and the dependence of fractal parameters on the value of the threshold in the range (0–255) was determined.

Then, the one-dimensional (1D) threshold analysis for all rows and columns of the image crop in the horizontal and vertical directions was conducted and fractal parameters were also determined in the same way.

Employing the Eq. 15, provided that the structure has the same fractal character for both previous analyzes (1D, 2D), the scaling parameter r and the number of repetitions $m = r^E - k$ were determined for an E-dimensional space (1D, 2D, 3D, .., k is the number of removed objects).

$$D_1 \approx \frac{\ln(r-k)}{\ln r} \;,\; D_2 \approx \frac{\ln(r^2-k)}{\ln r} \;,\; D_3 \approx \frac{\ln(r^3-k)}{\ln r} \;, \ldots \tag{14}$$

An example of this analysis derived from the Cantor discontinuum is depicted in Figure 2. The left figure shows a very infrequent (rare) structure, the fractal dimension is very small: for $m = r$, the fractal dimension of all structures will be equal to zero. The right figure shows a dense structure, the fractal dimension is very high: for $k = 0$, we will observe that $m = r^E$ and the fractal dimension of all structures will be equal to the Euclidean dimension.

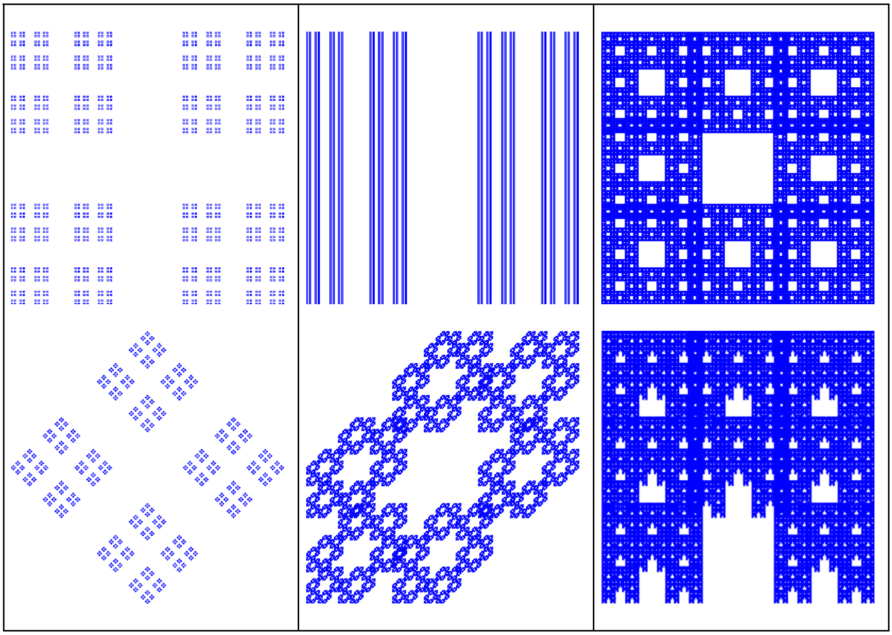


Fig. 2 Different fractal structures in 2D space derived from 1D Cantor discontinuum ($r = 3$, $m = 2$): a) Windows ($r = 3$, $m = 4$), b) Cantor stripes, Fractal stars ($r = 3$, $m = 6$), and c) Sierpinski carpet ($r = 3$, $m = 8$), see Table 1

Table 1 Parameters of different structures derived from Cantor discontinuum in 1D, 2D and 3D space: a) Windows, b) Cantor stripes, c) Sierpinski carpet, together with homogenous (no fractal) structure ($m = r^E$, $k = 0$, $D = E$) and structures with line dimension ($m = r$, $k = r^E - r$, $D = 1$)

		$D = 1$		Windows		Cantor stripes Fractal stars		Sierpinski carpet		$D = E$	
	r	m	k	m	k	m	k	m	k	m	k
1D	1:3	3	0	2	1	2	1	2	1	3	0
2D	1:3	3	6	4	5	6	3	8	1	9	0
3D	1:3	3	24	8	19	18	9	26	1	27	0

The calculated fractal parameters are summarized in Table 1. The scaling parameter r is the same for all structures from Fig. 2. The number of repetitions m is different. All structures are derived from the Cantor discontinuum (first row at the Table 1), but only the Sierpinski carpet is symmetric at both directions (the difference between the number of removed objects is the same, hence $k = r^E - m = \text{const.}$. In this case it is possible extrapolate the results of 1D and 2D analysis to 3D space. It means that the 3D structure will be symmetric at all three directions. This assumption will be applied to real structures, e.g. in Fig. 3

3.2 Gravitational Field

In this case, the physical quantity c_N used in previous equations stands for mass unit ($c_N = m_u$) and the material constant c_V is proportional to the Newton's constant of gravitation ($c_V = 4\pi G_N$). Application examples of mathematical equations (3), (7) and (9) to gravitational field are given in Table 2. It is defined by mass density, gravitational acceleration (field intensity) and gravitational potential.

Table 2 Physical quantities used for gravitational field

density of object	field intensity
$\rho(r) = m_u n(r) = m_u K r^{D-E}$	$F_r = a_r = 4\pi G_N m_u \dfrac{K r^{D-E+1}}{D}$
field potential	density of energy
$V_r = -4\pi G_N m_u \dfrac{K r^{D-E+2}}{D(D-E+2)}$	$w(r) = G_N m_u^2 \dfrac{K^2 r^{2(D-E+1)}}{D(E-D-2)}$

The square crop (256×256 pixels) of structure from Fig. 3 was analyzed by HarFA 5.5, The 1D and 2D wavelet analysis applied to threshold images was used for determining of 1D and 2D fractal parameters.

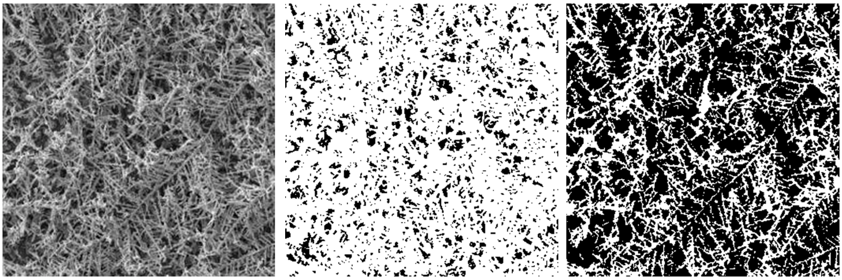


Fig. 3 An example of fractal distribution of mass: "silver nitrate fractal trees for solar technology" [11], gray scale (left image), threshold at level 64 (middle image) and threshold at level 128 (right image)

The results of this analysis are depicted in Fig. 4 and Fig. 5. The dependences of 2D analysis are plotted by solid lines; the dependences of 1D analysis are plotted by dashed lines. Worth special attention is the BW dependence which gives the information about the interface between black and white area (see Fig. 3 middle and right). These images were converted to black-and-white using different threshold values, brightness levels under the threshold level were assigned to white and below threshold level to black.

The graphs of fractal dimensions (Fig. 4) are scaled also as entropy (secondary right axis), see Eq. 14 for scaling value $r = 256$.

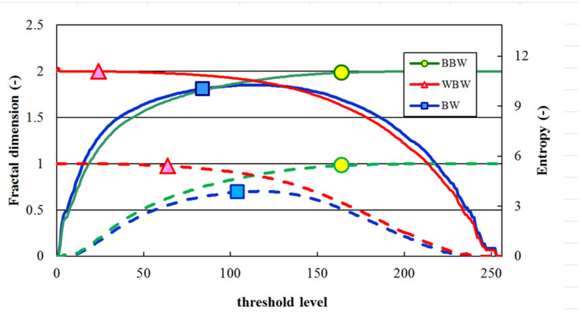


Fig. 4 Dependences of 1D and 2D fractal dimensions of mass structure from Fig. 3 on the threshold level (0-255); BW – fractal dimensions of surrounding lines, BBW – fractal dimensions of black area include surrounding line, WBW – fractal dimension of white area include surrounding lines

Graphs of fractal measures (Fig. 5) are scaled also as relative area (right axis)

$$S_{\mathrm{B}} = \frac{K_{\mathrm{BBW}} - K_{\mathrm{BW}}}{K_{\mathrm{BBW}} + K_{\mathrm{WBW}} - K_{\mathrm{BW}}}\,,\quad S_{\mathrm{W}} = \frac{K_{\mathrm{WBW}} - K_{\mathrm{BW}}}{K_{\mathrm{BBW}} + K_{\mathrm{WBW}} - K_{\mathrm{BW}}}\,,$$

$$S_{\mathrm{BW}} = \frac{K_{\mathrm{BW}}}{K_{\mathrm{BBW}} + K_{\mathrm{WBW}} - K_{\mathrm{BW}}}\,, \tag{15}$$

where S_{B} is size of black area, S_{W} is size of white area and S_{BW} is size of surrounding between black and white area (length of surrounding curve) - $S_{\mathrm{B}} + S_{\mathrm{W}} + S_{\mathrm{BW}} = 1$.

The parameters r (scaling parameters), m (number of repetitions), and k (difference between the number of parts and number of repetitions) were determined from 1D and 2D analysis by analyzing the data from Fig. 4 and using the relations (15)

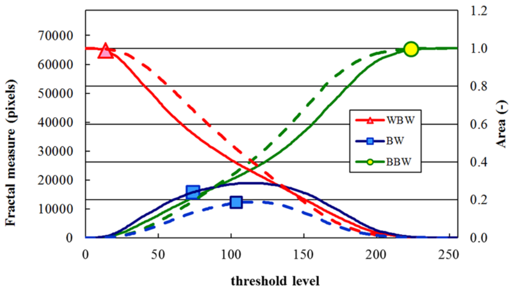


Fig. 5 Dependences of 1D and 2D fractal measures of mass structure from Fig. 3 on the threshold level (0-255); BW – fractal measures of surrounding lines, BBW – fractal measures of black area include surrounding line, WBW – fractal measures of white area include surrounding lines

In Fig. 6, these dependences are imaged as the function of the threshold level. From this figure it is evident that the maximal scaling parameter is **1 : *r* = 1 : 1.93**, and the maximal number of repetition is ***m* = 1.58** and the corresponding fractal dimension is ***D* = 0.700** and the entropy for this square crop (256×256 pixels), Eq. (14) is ***S* = 3.88**. It is in agreement with maximal 1D fractal dimension for BW dependence given at Fig. 4.

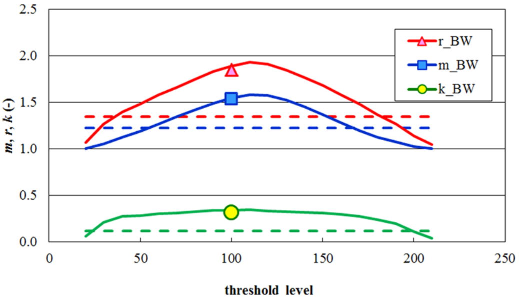


Fig. 6 Dependences of fractal parameters for determining of fractal dimension of mass structure from Fig. 3 on the threshold level (0-255); r_BBW is scaling parameter, m_BBW is a number of repetitions and k_BBW is their difference for surrounding lines calculated from 1D analysis, dashed lines – results of gray scale analysis

For 2D fractal analysis, the maximal number of repetition is ***m* = 3.38.** For the same scaling parameter, the fractal dimension is ***D* = 1.851** and entropy is ***S* = 10.27**. It is in agreement with the maximal 2D fractal dimension for BW dependence given at Fig. 4. And finally, the fractal dimension of 3D fractal structure can be calculated by extrapolation to three dimensional Euclidean space: number of repetition will be ***m* = 6.84,** fractal dimension ***D* = 2.925** and entropy ***S* = 16.22**. The dependence of number of repetitions on the threshold level is for all three cases depicted at Fig. 7

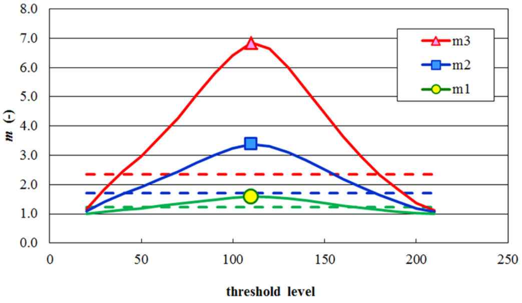


Fig. 7 Dependences of fractal parameter *m* (number of repetition) of mass structure from Fig. 3 on the threshold level (0-255 calculated from 1D (m1) and 2D (m2) analysis, extrapolated to 3D (m3), dashed lines – results of gray scale analysis

Substituting the calculated results (fractal dimension for 3D space, E = 3) to the equations given in Table 2, we get information about fractal character of the analyzed structure. The mass density will be slowly decreasing with power of $D -$ E = –0.075 ($\rho \approx r^{-0.075}$). It is nearly homogeneous structure. The gravitational field intensity will be slowly almost linearly increase $D - E + 1 = 0.925$ ($F_\mathrm{r} \approx r^{0.925}$) and the gravitational potential will be increasing almost quadratically $D - E + 2 = 1.925$ ($V_\mathrm{r} \approx r^{1.925}$). And finally the density of energy will be increasing with the power of $2(D - E + 1) = 1.85$ ($w \approx r^{1.85}$).

4 Conclusion

The fractal theory of physical field is described at this contribution. The fractal quantity such as density, field intensity and potential of physical field was defined for radial coordinates in generally *E*-dimensional Euclidean space. At the end of the theoretical part the density of energy and entropy is defined.

The experimental part demonstrates the introduced theory on the study of the properties of materials using image analysis. The results of 1D and 2D fractal analysis were used for the determination of scaling parameters and number of repetitions. These parameters were then used to determine the prediction of fractal dimension in 3D space assuming a homogeneous 3D structure. The calculated fractal dimension ***D* = 2.925** and entropy ***S* = 16.22** were used for determining of physical parameters (mass density, gravitational field intensity, and gravitational potential) of analyzed structure. We found out that for a nearly homogeneous fractal structure, the mass density is slowly decreasing with scaling, the gravitational field intensity is increasing almost linearly and the gravitational potential is increasing quadratically. The density of energy of gravitational field was increasing with scaling, too.

Acknowledgement. This work was supported by the projects from the Ministry of Industry and Trade of the Czech Republic (Grant FR-TI1/144), project from the Technology Agency of the CR (Grant TA03010548) and from “Centre for Materials Research at FCH BUT" No. CZ.1.05/2.1.00/01.0012 supported by ERDF.

References

[1] Zmeskal, O., Nezadal, M., Buchnicek, M.: Fractal–Cantorian Geometry, Hausdorff Dimension and the Fundamental Laws of Physics. Chaos, Solitons & Fractals 17, 113–119 (2003)

[2] Mandelbrot, B.B.: Fractal Geometry of Nature. W. H. Freeman and Co., New York (1983)

[3] Zmeskal, O., Nezadal, M., Buchnicek, M.: Field and potential of fractal–Cantorian structures and El Naschie’s ∞ theory. Chaos, Solitons & Fractals 19, 1013–1022 (2004)

[4] Zmeskal, O., Buchnicek, M., Vala, M.: Thermal Properties of bodies in fractal and Cantorian physics. Chaos, Solitons & Fractals 25, 941–954 (2005)

[5] Zmeskal, O., Vala, M., Weiter, M., Stefkova, P.: Fractal-cantorian geometry of space-time. Chaos, Solitons & Fractals 42, 1878–1892 (2009)

[6] Zmeskal, O., Buchnicek, M., Bednar, P.: Coupling Constants in Fractal and Cantorian Physics. Chaos, Solitons & Fractals 22, 985–997 (2004)

[7] Zmeskal, O., Nespurek, S., Weiter, M.: Space-charge-limited currents: An E-infinity Cantorian approach. Chaos, Solitons & Fractals 34, 143–156 (2007)

[8] Richardson, L.F., Ashford, O.M., Charnock, H., Drazin, P.G., Hunt, J.C.R., Smoker, P., Sutherland, I.: The Collected Papers of Lewis Fry Richardson, Cambridge (1993)

[9] Mandelbrot, B.B.: A fractal set is one for which the fractal (Hausdorff-Besicovitch) dimension strictly exceeds the topological dimension. Fractals and Chaos. Springer (2004)

[10] Zmeskal, O., Dzik, P., Vesely, M.: Entropy of fractal systems. Computers and Mathematics with Applications (2013), doi:10.1016/j.camwa.2013.01.017

[11] Tiny trees for solar power, UCDavis (2012), http://www.news.ucdavis.edu/search/news_detail.lasso?id=10167

[12] Lichtenberg Figure Oval Shape, Science Enterprices (2012), http://www.scienceenterprises.com/lichtenbergfigureovalshape.aspx

[13] Simulating ALMA observations of a simulated protoplanetary disk (2007), http://www.cv.nrao.edu/~rreid/ppdisksims/

Wavelet Based Feature Extraction for Clustering of Be Stars

Pavla Bromová, Petr Škoda, and Jaroslav Zendulka

Abstract. The goal of our work is to create a feature extraction method for classification of Be stars. Be stars are characterized by prominent emission lines in their spectrum. We focus on the automated classification of Be stars based on typical shapes of their emission lines. We aim to design a reduced, specific set of features characterizing and discriminating the shapes of Be lines. In this paper, we present a feature extraction method based on the wavelet transform and its power spectrum. Both the discrete and continuous wavelet transform are used. Different feature vectors are created and compared on clustering of Be stars spectra from the archive of the Astronomical Institute of the Academy of Sciences of the Czech Republic. The clustering is performed using the kmeans algorithm. The results of our method are promising and encouraging to more detailed analysis.

1 Introduction

Technological progress and growing computing power are causing data avalanche in almost all sciences, including astronomy. The full exploitation of these massive distributed data sets clearly requires automated methods. One of the difficulties is the inherent size and dimensionality of the data. The efficient classification requires that we reduce the dimensionality of the data in a way that preserves as many of the physical correlations as possible.

Pavla Bromová · Jaroslav Zendulka
Faculty of Information Technology, Brno University of Technology, Božetěchova 1/2,
612 66 Brno, Czech Republic
e-mail: {ibromova,zendulka}@fit.vutbr.cz

Petr Škoda
Astronomical Institute of the ASCR, v. v. i., Fričova 298, 251 65 Ondřejov, Czech Republic
e-mail: skoda@sunstel.asu.cas.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 467–474.
DOI: 10.1007/978-3-319-00542-3_46

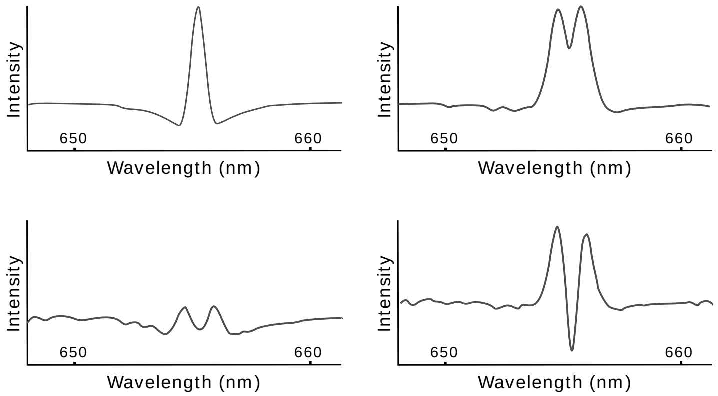


Fig. 1 Typical shapes of emission lines in spectra of Be stars

Be stars are hot, rapidly rotating B-type stars with equatorial gaseous disk producing prominent emission lines H_α in their spectrum [11]. Be stars show a number of different shapes of the emission lines, as we can see in Fig.1. These variations reflect underlying physical properties of a star.

Our work is focused on the automated classification of Be stars based on typical shapes of their emission lines. There has not been much work on classification of Be stars. The only application found [2] is focused on a broader category of variable stars including pulsating Be stars. However, the method is not suitable for our goals, as it is applied on the whole spectrum where the local differences in the shapes of Be lines are lost. We need to zoom at the small part of a spectrum with the Be line and to design a reduced, specific set of features characterizing and discriminating the shapes of Be lines. Due to a large variety of shapes, it is not easy to construct simple criteria (like e.g. Gaussian fits) to identify the Be lines in an automatic manner.

In this paper, we present the feature extraction method based on the wavelet transform and its power spectrum. Both the discrete and continuous wavelet transform are used. Different feature vectors are created and compared on clustering of Be stars spectra from the archive of the Astronomical Institute of the Academy of Sciences of the Czech Republic.

2 Method

The method is based on the wavelet transform and its power spectrum. A resulting feature vector is composed of two parts: 1. wavelet power spectrum, 2. value indicating the orientation of the H_α line (this information is lost in the wavelet power spectrum). The process of creating the feature vector is described here.

2.1 Wavelet Transform

Many different transforms are used in data processing (Fourier transform is perhaps the most widely used) [9]. The goal of these transformations is to obtain a sparse representation of data, and to pack most information into a small number of samples.

The wavelet transform consists in decomposing signals (data) into different frequency components. A wavelet is an elementary oscillatory waveform of a limited duration with an average value of zero. A signal is convolved with a scaled, shifted versions of a wavelet resulting in coefficients, which are a function of scale and position. Their structure leads to a fast computational algorithm. Wavelets are well localized and therefore suitable for revealing local transcient structures in data. Extensive literature exists on wavelets and their applications, e.g. [6, 1, 3, 7, 10].

2.1.1 Continuous Wavelet Transform

The continuous wavelet transform (CWT) is defined as the sum over all time of the signal multiplied by scaled, shifted versions of the wavelet function ψ:

$$C(s,p) = \int_{-\infty}^{\infty} f(t)\psi(s,p,t)dt,$$

where s is a scale of a wavelet and p is a position in the signal. The result of the CWT are wavelet coefficients C, which are a function of scale and position. The wavelet coefficients reflect the correlation between the wavelet and the data array. A larger absolute value of a coefficient implies a higher correlation.

There is a correspondence between wavelet scales and frequency as revealed by wavelet analysis: low scale $\rightarrow$ high frequency, high scale $\rightarrow$ low frequency.

In CWT (unlike DWT), it is possible to use every scale, and the analyzing wavelet is shifted smoothly over the full domain of the analyzed function.

2.1.2 Discrete Wavelet Transform

In the discrete wavelet transform (DWT), scales and positions are based on powers of two (dyadic scales and positions). An efficient way to implement this scheme using filters was developed yielding a fast wavelet transform.

The principle can be described as passing the original signal through two complementary filters – low-pass and high-pass [8]. This results in two signals, referred to as approximation and detail. The approximation is a high-scale, low-frequency component of the signal, the detail is a low-scale, high-frequency component. After each pass through filters, downsampling (removing every alternative coefficient) is performed in order to avoid doubling the amount of data.

The decomposition process can be iterated by splitting the approximation part of a signal as it still contains some details. So a signal is broken down into many lower-resolution components. The decomposition can proceed only until the individual details consist of a single sample. The wavelet transform then consists of approximation coefficients at last level and detail coefficients at all levels. For more details see e.g. [6].

2.2 Wavelet Power Spectrum

The wavelet power spectrum (WPS) is a useful way how to determine the distribution of energy within the signal [12]. By looking for regions of large power within WPS, we can determine which features of the signal are important.

The WPS at a particular decomposition level is calculated by summing up the squares of wavelet coefficients at that level [8]. For a set of wavelet coefficients $c_{j,k}$, where j is the level of decomposition and k is the order of the coefficient, WPS is given by:

$$wps(j) = \sum_{k=0}^{2^j-1} c_{j,k}^2$$

2.2.1 Rectification

It has been shown in [5] that wavelet power spectra are biased in favor of low frequencies. For example, for a signal comprising two sine waves of the same amplitude but distinct frequencies, a wavelet analysis will result in two spectral peaks of a different magnitude, the one on the low frequency being larger. This counters our expectation and is also in contrast to the result of any classical global analysis (such as Fourier transform), making comparison of the peaks across the scales impossible.

In [5], they established theoretically that the bias actually results from the traditional definition of energy for the wavelet power spectra which is not physically consistent. They present a physically consistent definition of energy: the transform coefficients squared divided by the scale they associate. The traditional biased power spectra are therefore easily rectified.

2.3 Normalization

WPS is normalized so that its total energy equals to 1, so it consists of percentages of energy corresponding to individual levels.

2.4 Orientation of Spectral Line

The information about the orientation of a spectral line is lost in the wavelet power spectrum, so we need to add it somehow into the feature vector. We want to distinguish whether a spectral line is oriented up (emission line) or down (absorption line), so we use one positive and one negative value. The question is which absolute value to choose. So far we have tried three values: 1, 0.1, and the amplitude of a spectral line, measured from the continuum of value 1.

3 Experiments

The experimental verification of the feature extraction method is performed using clustering. So far, the whole process has been implemented in Matlab, using its embedded algorithms. The stages are described in following sections.

3.1 Data Selection

We use spectra of Be stars from the archive of the Astronomical Institute of the Academy of Sciences of the Czech Republic. The spectra intended for spectral data mining are divided into 11 categories based on the shape of the H_α line. From them, 3 categories contains spectra of unstable stars, 1 category is composed of uncategorized (unknown) samples, 3 categories contains not enough samples for data mining, so only 4 categories are suitable for our experiments. The number of samples in these 4 categories are: 66, 150, 164, 276. We select at most 200 samples from each category, resulting in 656 samples in total.

From each spectrum, we select only a small part containing the emission line, so that the sample has 256 values and the emission line is approximately in the center. The number 256 was chosen according to the average width of the H_α line and according to DWT requirement on the length of the input data being a power of 2.

The spectra are normalized – lying on a continuum of a value 1.

3.2 Feature Extraction

The feature extraction method is described in the previous chapter. We use both continuous and discrete wavelet transform. As we do not have any reference method for Be stars for comparison, we compare our results with a common way of feature extraction from time series using wavelets – keeping *N* largest coefficients of wavelet transform [4].

From resulting coefficients of the wavelet transform we create different kinds of feature vectors which are used for comparison in experiments:

1. **Spectrum:** original spectrum values, normalized to range [0,1]. (In this case the DWT coefficients are not used.)
2. **Approximation:** DWT approximation coefficients, normalized to range [0,1].
3. **Approximation + detail:** DWT approximation and detail coefficients of the last level, normalized to range [0,1].
4. **10 largest coefs:** 10 largest absolute values of coefficients, normalized to range [-1,1].
5. **20 largest coefs:** 20 largest absolute values of coefficients, normalized to range [-1,1].
6. **DWPS + orientation 1:** one part of a feature vector is a discrete wavelet power spectrum, normalized so that its total energy equals to 1. Second part of a feature vector is a value indicating the orientation of a spectral line – lines oriented up have the value 1, lines oriented down have the value -1.
7. **DWPS + orientation 0.1:** the same as the previous one, except the absolute value of orientation 0.1.
8. **DWPS + amplitude:** one part of a feature vector is normalized wavelet power spectrum as in the previous case. The second part is the amplitude of the spectral line measured from the continuum of value 1.

9. **CWPS 16 + orientation 1:** continuous wavelet power spectrum (normalized), CWT performed with 16 scales. Same orientation as in the previous cases with DWPS.
10. **CWPS 8 + orientation 1:** continuous wavelet power spectrum (normalized), CWT performed with 8 scales. Same orientation as in the previous case.

In experiments up to now, we have used ”symlet 4” wavelet. In DWT, the maximum possible level of decomposition = 5 has been used.

3.3 Clustering

So far, the k-means algorithm in Matlab has been used for clustering. Squared Euclidean distance is used as a distance measure. Clustering is repeated 30 times, each with a new set of initial cluster centroid positions. Kmeans returns the solution with the lowest within-cluster sums of point-to-centroid distances.

3.4 Evaluation

We proposed an evaluation method utilizing our knowledge of ideal classification of spectra based on a manual categorizing.

The principle is simply to count the number of correctly classified samples. We have 4 target classes and 4 output classes, but the problem is we do not know which output class corresponds to which target class. So first we need to map the output classes to the target classes, i.e. to assign each output class a target class. This is achieved by creating the correspondence matrix, which is a square matrix of a size of a number of classes, and where the element on a position (i, j) corresponds to the number of samples with an output class i and a target class j. In a case of a perfect clustering, all values besides the main diagonal would be egual to zero.

Now we find the mapping by searching for the maximum value in the matrix. The row and the column of the maximum element will constitute the corresponding pair of output and target class. We set this row and column to zero and again find the maximum element. By repeating this process we find all corresponding pairs of classes. The maximum values correspond to correctly classified samples. So now we simply count the number of correctly classified samples by summing all maximum values we used for mapping the classes. By dividing by the total number of samples we get the percentual match of clustering which is used as a final evaluation.

4 Results

Fig. 2 shows the percentual match of the clustering for different kinds of feature vectors. The numbers of feature vectors in the figure correspond to the numbers in the numbered list in 3.2.

The best results are given by the last feature vector consisting of the continuous wavelet power spectrum calculated from 8 scales of CWT coefficients, and the value

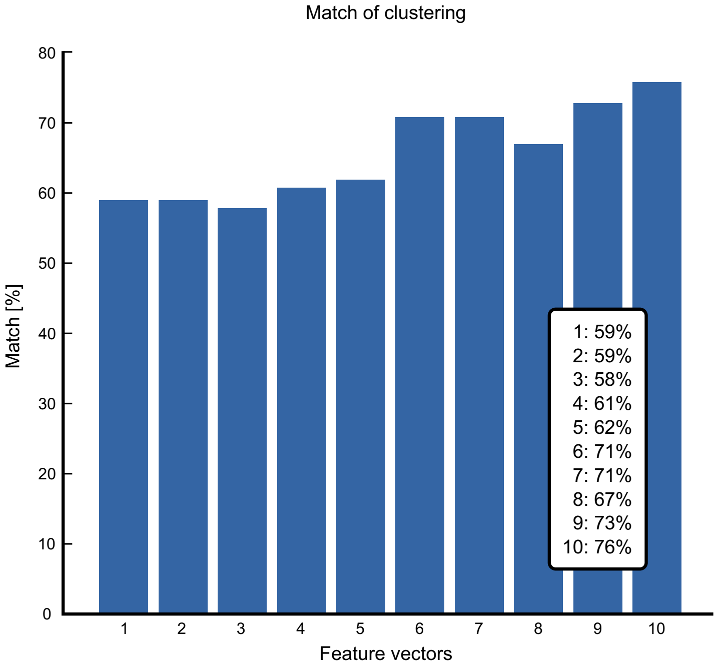


Fig. 2 The match of the clustering using different feature vectors

representing the orientation of the H_α line with absolute value of 1. The match is 14% higher than the best result of a feature vector without WPS. Also the results of all other feature vectors containing WPS are better than the feature vectors without WPS.

These results are promising and indicate the continuous wavelet transform being more suitable for our purpose. This encourages to focus on the CWT and perform more experiments, as we have focused more on the DWT up to now.

5 Conclusion

The goal of our work is to create a feature extraction method for classification of Be stars based on typical shapes of their emission lines. We aim to design a reduced, specific set of features characterizing and discriminating the shapes of the H_α emission lines.

In this paper, we have analysed the capabilities of using wavelet power spectrum for classification of spectra of Be stars. We have presented a feature extraction method based on the wavelet transform and its power spectrum. Different feature vectors have been created and compared on clustering of Be stars spectra. The evaluation was performed utilizing our knowledge of ideal classification of spectra based on a manual categorizing.

The best results are given by the feature extraction method based on the CWT. Results are promising and indicate the CWT is more suitable for our purpose than DWT, which encourages to focus on the CWT and perform more detailed analysis.

Acknowledgements. This work was supported by the specific research grant FIT-S-11-2.

References

1. Daubechies, I.: Ten lectures on wavelets. CBMS-NSF regional conference series in applied mathematics. Society for Industrial and Applied Mathematics (1994)
2. Debosscher, J.: Automated Classification of variable stars: Application to the OGLE and CoRoT databases. PhD thesis, Institute of Astronomy, Faculty of Sciences, Catholic University of Leuven (2009)
3. Kaiser, G.: A friendly guide to wavelets. Birkhäuser (1994)
4. Li, T., Ma, S., Ogihara, M.: Wavelet methods in data mining. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, pp. 553–571. Springer (2010)
5. Liu, Y., San Liang, X., Weisberg, R.H.: Rectification of the bias in the wavelet power spectrum. Journal of Atmospheric and Oceanic Technology 24(12), 2093–2102 (2007)
6. Mallat, S.: A Wavelet Tour of Signal Processing: The Sparse Way, 3rd edn. Academic Press (2008)
7. Meyer, Y., Salinger, D.H.: Wavelets and Operators. Cambridge Studies in Advanced Mathematics, vol. 1. Cambridge University Press (1995)
8. Prabakaran, S., Sahu, R., Verma, S.: Feature selection using haar wavelet power spectrum. BMC Bioinformatics 7, 432 (2006)
9. Starck, J.L., Murtagh, F.: Astronomical image and data analysis. Astronomy and Astrophysics Library. Springer (2006)
10. Strang, G., Nguyen, T.: Wavelets and filter banks. Wellesley-Cambridge Press (1996)
11. Thizy, O.: Classical Be Stars High Resolution Spectroscopy. Society for Astronomical Sciences Annual Symposium 27, 49 (2008)
12. Torrence, C., Compo, G.P.: A practical guide to wavelet analysis. Bulletin of the American Meteorological Society 79, 61–78 (1998)

Upcoming Features of SPLAT-VO in Astroinformatics

Petr Šaloun, David Andrešič, Petr Škoda, and Ivan Zelinka

Abstract. During last decade was developed fully automatized (robotic) class of telescopes, that produce huge amount of data per each night. Amount of recorded data is usually in the scale of petabytes. To process properly all data and select an important events it is needed to use sophisticated software methods and algorithms. It caused an appearance of a new field of science - astroinformatics. In this paper we introduce a small part of our contribution to the astroinformatics field - a specialized software SPLAT-VO. It is used for processing and visualization of astrophysical data generated by nonlinear, complex or even chaotic processes in the space. Overview of new features so far prepared for new version of SPLAT-VO. The overview is focused on enhancements of user experience, work with SAMP protocol and other interoperability that improves work with global list of spectra, plot window and analysis menu.

1 Introduction

Virtual observatory (VO) is a collection of interoperating data archives and software tools which utilize the internet to form a scientific research environment in which astronomical research programs can be conducted. As written in [7] the contemporary astronomy is flooded with an exponentially growing petabyte-scaled data volumes produced by powerful ground and space-based instrumentation as well as a product of extensive computer simulations and computations of complex numerical models.

Petr Šaloun · David Andrešič · Ivan Zelinka
Department of Computer Science, Faculty of Electrical Engineering and Computer Science,
VŠB - Technical University of Ostrava, 17. listopadu 15,
708 33 Ostrava-Poruba, Czech Republic
e-mail: {petr.saloun,ivan.zelinka,david.andresic}@vsb.cz

Petr Škoda
Astronomical Institute of the Academy of Sciences, Fricova 298, 251 65 Ondrejov
e-mail: skoda@sunstel.asu.cas.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 475–486.
DOI: 10.1007/978-3-319-00542-3_47

The efficient organization and seamless handling of this information avalanche stored in a world-wide spread heterogeneous databases and the facilitation of extraction of new physical knowledge about the Universe is a primary goal of the rapidly evolving astronomical Virtual Observatory (VO). We give an overview of current spectroscopic capabilities of VO and identify the future requirements indispensable for detailed multi-wavelength analysis of huge amounts of spectra in a semi-automatic manner. VO offers huge data-sets with thousand fuzzy interconnected attributes for researcher, but it is necessary to select the proper one and show them...

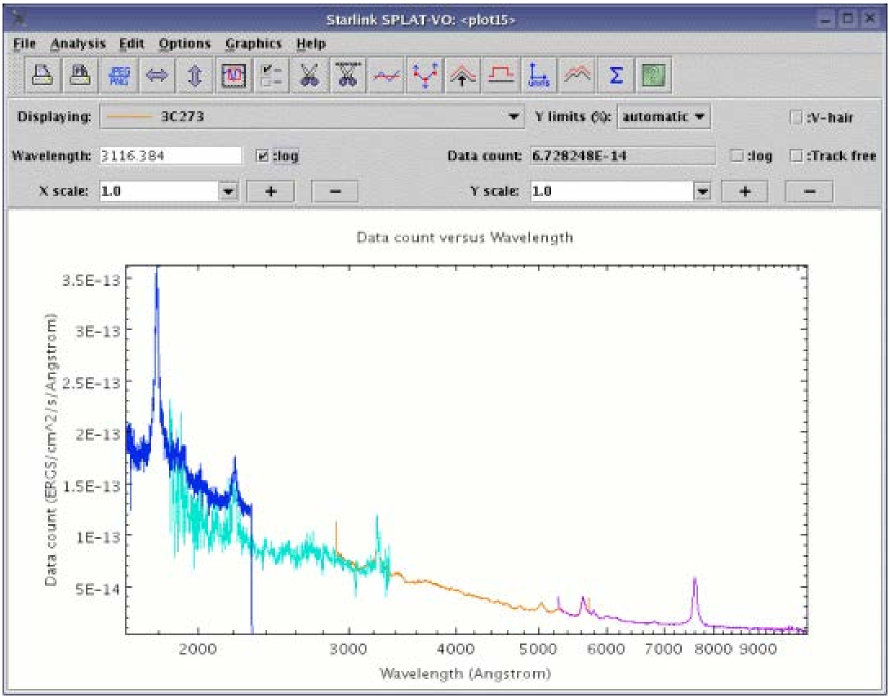

Displaying: 3C273
Y limits (X): automatic
□ :V-hair
Wavelength: 3116.384
☑ :log
Data count: 6.728248E-14
□ :log
□ :Track free
X scale: 1.0
Y scale: 1.0

Fig. 1 SPLAT-VO at work

SPLAT is a shortcut for Spectral Analysis Tool [6] and it is a tool for displaying, modifying and analysing astronomical spectra, see Fig. 1 and Fig. 2 respectively, for example. It was developed in 2003 as a part of Starlink (and its STARJAVA package) project [6]. In 2005, the Starlink was closed down and in 2006 took over by Joint Astronomy Centre[1] that relicensed some of its parts under GNU/GPL licence [3][2].

During its development, the SPLAT was extended to include facilities that allowed an interoperability with the Virtual Observatory [6][3] (VO). The VO is a

[1] `www.jach.hawaii.edu`

[2] `www.gnu.org/licenses/gpl-3.0.txt`

[3] Standardized, coordinated and developed by International Virtual Observatory Alliance (IVOA) - `www.ivoa.net`

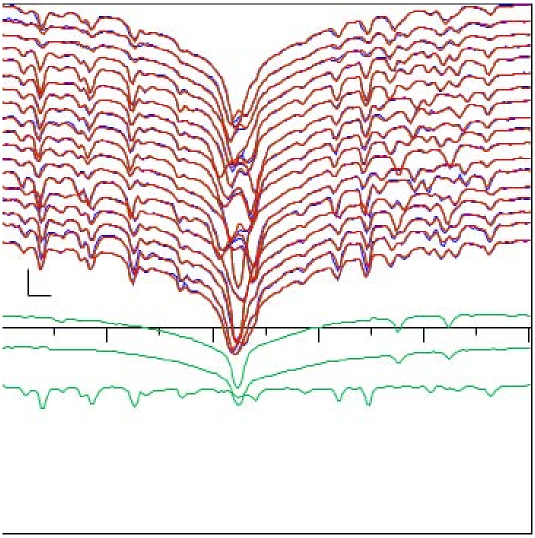


Fig. 2 Double star Her 96 and its H_α spectral line

collection of astronomical archives and software tools that utilizies the internet to allow international collaboration and scientific research [2]. SPLAT itself uses the SSAP[4] (Simple Spectra Access Protocol) protocol for obtaining the spectra from SSAP servers and SAMP[5] (Simple Application Messaging Protocol) protocol to interoperate with other SAMP-compatible tools like TOPCAT[6].

Since 2012, SPLAT-VO is being developed by the GAVO[7] (German Astrophysical Virtual Observatory) in cooperation with the Astronomical Institute of the Academy of Sciences of the Czech Republic [6][8]. Its development is currently focused on implementing new VO standards. The need to extend the scientific features that doesn't immediately relate to the VO and fix some of the bugs or deficiencies resulted in a bachelor thesis at VŠB-TUO[9] supervised by Petr Škoda of Astronomical Institute of the Academy of Sciences of the Czech Republic that scoped some (so far for time reasons just minor) changes to SPLAT-VO as described below.

In the near future, after dealing with coordination and unification of SPLAT's development, we intend to continue in adding new features and implementing new VO standards so the SPLAT can be a feature-rich, user-friendly and even more portable scientific tool and reference implementation of new VO standards.

[4] `www.ivoa.net/Documents/SSA/`

[5] `www.ivoa.net/Documents/latest/SAMP.html`

[6] `www.star.bristol.ac.uk/~mbt/topcat/`

[7] `www.g-vo.org`

[8] `www.asu.cas.cz`

[9] VŠB - Technical University of Ostrava - `www.vsb.cz`

2 New Features

Beside the strictly Virtual Observatory related changes, which will allow the user even more detailed searching for spectra, the upcoming version of SPLAT-VO focuses on the user experience, interoperability with other tools and work with the spectra itself. On the next few pages, we will try to get you familiar with these new features that (as we believe) will make your work with SPLAT more enjoyable.

2.1 Plot Window

The purpose of plot window is quite simple: plotting every spectrum selected in the global list of spectra and allowing the user to perform analysis on it. Every plot window can contain one or more spectra and when working with SPLAT, the user can have opened many plot windows with many spectra plotted in it.

2.1.1 Added Visual Spectrum Selection Capability

So far, when the user worked with multiple spectra in one common plot window, he has had a limited possibility of selection just one of them for further actions. Selection of eg. noisy spectrum and its deletion from plot window could be done only by trial-and-error procedure and therefore quite problematic and time-wasting.

In the upcoming version, the user can click by mouse inside the plot window and SPLAT will automatically select the closest spectrum. The spectrum gets selected in the local spectra drop-down menu above the plot itself, and in the global list of spectra in the main window. From there, it can be easily deleted or modified for plotting and analysis.

The actuall selection is done by getting the $[x,y]$ coordinates of mouse click, that are then transformed to coordinates with the plot's $[0,0]$ as beginning. Then the algorithm searches for all spectra with x coordinates containing the x coordinate of the mouse click. For this, the native (platform dependent) Ast library from the original Starlink project is used. Each spectrum is represented by series of $[x,y]$ coordinates, each coordinate represents one value. The Ast library transforms this descrete list of values to a continuous line that we see in the plot window and it's this set of transformed coordinates, that the selection algorithm uses. So when any spectrum detected on the x coordinate, the algorithm will again use the modified binary search algorithm[10] to select the closest spectrum on the y axis. This is quite similar to the procedure on x axis, the Ast-transformed coordinates are used to locate

[10] The usuall binary search algorithm searches for the exact match of searched item in the sorted list by getting closer to it by dividing the current interval (where the first interval is $0...n$) by two. Then, by comparising the searched value to the subinterval's borders, it divides the first or second subinterval in the next iteration. The modified algorithm doesn't look for the exact match, but gets as close as possible to the searched item and then compares the first item on left side with first item on the right side and selects the closest to the searched one.

the closest match of each x-colliding spectrum's $[x,y]$ coordinate to the mouse click coordinates.

The selection of the spectrum is quite reliable and for ensuring this, the selected spectrum gets immediately highlighted by few-seconds blinking in inverted color (as more closely described in 2.2.2).

The usage of modified binary search algorithm is crucial when considering the user's computer performance so it won't cause any slow-down of SPLAT's interface.

2.2 Global List of Spectra

The global list of spectra is a part of the SPLAT's main window. It is a container (`JList`) for every spectrum opened no matter from which source. Since it is a source of each spectrum in SPLAT (works sort of as a proxy pattern, no matter what the source is, or in what format the spectrum is, each one can be used transparently and in the same way since it is transformed to an internal `SpecData` object), every major component of SPLAT is tighted to it.

2.2.1 Saving in FITS Format

Previous versions of SPLAT were able to save its global list of spectra only to their native binary format (more technically, it's just a serialized object representing the `JList` with individual spectra's instances). But sometimes, there may be a need to save that list to a more universal, standardized format. For example, SPLAT allows to perform many operations on spectra, which results in a new spectrum object that is added to a global list of spectra (eg. cutting a part of spectrum). Saving the list of such spectra to a standardized format would then allow its opening in another tool and continuing to work with it in a way the SPLAT wasn't designed to.

Therefore, the upcoming version of SPLAT-VO will allow the user to save the global list of spectra to a universal FITS (Flexible Image Transport System) format. The FITS format is multipurpose format for storing the scientific data, primary designed and used for astronomy [4]. The FITS allows storing multiple data of multiple type in it, for example, it can contain the image of a star and its spectrum (or spectra) with corresponding metadata. This is done by a concept of extensions. For a programmer, the extension can be described as an instance of a class, where class itself is a type of extension. We recognize these basic types of extensions [1]:

IMAGE Provides a means of storing a multidimensional array similar to that of the FITS primary header and data unit.

TABLE ASCII table extension type contains rows and columns of data entries expressed as ASCII characters.

BINTABLE Provides a more flexible and efficient means of storing data structures than is provided by the TABLE extension type.

Every FITS file consists of one primary Header Data Unit (HDU) and of an unlimited number of extension HDUs. Every HDU contains (beside its actual data) the ASCII header with corresponding metadata as key/value pairs, where the most important keys are standardized and understood by FITS readers (eg. EXTENSION key identifying the type of the current extension).

In the upcoming version of SPLAT, every spectrum will be represented by its own IMAGE extension. So when having, let's say five spectra in the global list, it will result in a FITS output with five IMAGE extensions (one for each spectrum). The only thing that the user will have to do for this, is selecting the newly added FITS/IMAGE format in a dialog window for saving the global list of spectra, instead of the previous proprietary format.

It should be said, that in the current state the SPLAT cannot open the global list of spectra saved in FITS/IMAGE format, but since the cruciality of saving to more universal formats, it will be added very soon. We are also planning to add the support for another types of FITS extensions (eg. BINTABLE), as well as other formats, like VO-Table.

2.2.2 Highlighting the Spectrum after Selection

When user clicked on a spectrum in global spectra list and this spectrum was plotted to a window(s) with multiple spectra, he had no way of knowing, which one of them it is.

In the upcoming version, when user clicks on a spectrum in global spectra list, the spectrum gets immediately highlighted in all plot windows that it is opened in. The highlighting has a form of a few-seconds blinking in an inverted color and is performed sequentially (highlighting in one plot window comes after the previous one is finished).

The algorithm used to do this is the same as in the case of visual spectra selection as briefly described in 2.1.1 and designed to not cause any slow-down of SPLAT's interface. This is done by listening the mouse-click event and performing all the iterations of inverting the spectrum's plot color, repainting the components and sleeping itself in special threads, so SPLAT's user interface doesn't get freezed during the sleep phase and is not touched otherwise. The highlighting threads are also synchronized, so one starts highlighting after the previous is finished and the user gets a perfect sequential overview of the spectrum's plots.

As for the inversion, the spectrum's color is treated as `java.awt.Color`[11] in its integer expression. Basic colors - red (R), green (G) and blue (B) - are encoded this way:

- Red: bits 16-23;
- Green: bits 8-15;
- Blue: bits 0-7.

For inverting the color, we need to decode each basic color (this can be done by class's getters), invert it by subtracting this decoded value from 255 (range for each color) and putting it back to its decade by multiplying it by the corresponding offset (1 for blue, 256 for green and 65536 for red). The final inverted color is then a sum of these inverted components.

[11] `docs.oracle.com/javase/1.4.2/docs/api/java/awt/Color.html`

2.3 Analysis Menu

The plot window's Analysis menu is a doorway to SPLAT's arsenal of analytic features like cutting regions from spectra, fitting, interpolation, filtering and many more. Many of these functions are still based on native, platform dependent libraries of the original Starlink project.

2.3.1 Cut Window: Saving the Ranges

In the 'Cut regions from spectrum' window, user could simply read ranges of regions from a local file (feature available under 'File' menu). But what if he made a visual selection of ranges from a currently plotted spectrum and wanted to save it to a local file?

Since the upcoming version, SPLAT can save the currently defined ranges to a local file (feature is available under 'File' menu) in the same format as the reading feature expects, that is one range (with a whitespace between its beginning and ending value) per one line:

```
# Generated by SPLAT-VO

# Range 1
8601.177 8621.531

# Range 2
8729.026 8748.744
```

Adding this feature was the simplest of all, since SPLAT already possessed this feature, just was using it to a different purpose. Thanks to this, it consisted only from adding the menu item and binding the existing function to it.

2.3.2 Cut Window: Performing on Multiple Spectra

And cutting regions from spectrum again: until now, when user defined all the wanted regions and performed the cut action, SPLAT cutted the regions only from the currently active spectrum.

In the upcoming version, this window will possess a table of all currently plotted spectra in the corresponding plot window. User then can select multiple of it (buttons for selecting and deselecting all the spectra are matter of course) and perform a cut (or any other action in this window) on all the selected spectra.

Adding this feature was quite straightforward. It consisted from creating a new class extending the standard `JTable`, its appropriate cell-render class and passing it a `JList` with all currently plotted spectra. The `JList`'s items then get rendered to this table. One of the new methods returns a list of all selected spectra, which is only a step from modifying the action functions (cut etc.) to iterate over it.

This new spectra selection table is written very generally so adding a similar feature anytime in the future will then be quite easy.

2.4 SAMP-Compatible Tools Interoperability

Simple Application Messaging Protocol (SAMP) is an universal, XML-based, event-driven messaging protocol for exchanging control and data information and thus allows the (not just astronomical) software to interoperate [5].

Usage of SAMP allows building an event-driven publish/subscribe messaging system with both synchronous and asynchronous communication. The whole communication is centralized, all elements communicates with a SAMP hub that takes care of routing the messages [5]. In the typical scenario, an application interested in using the SAMP looks for a SAMP hub using the appropriate discovery mechanism (the hub itself - using the same mechanism - guards that there is only one SAMP hub running), registers with it and sends to it metadata containing its name, type of messages that it's interested in and asks the hub for information about other clients. Then, when some hub client (publisher) has a message to publish, it will notify the hub that will decide (based on type of the message) which registered clients (subscribers) will be notified as well. When a subscriber is notified, it can ask the hub for the message, the hub will then require the message from the publisher and sends it to a subscriber.

For having a better idea about SAMP's purpose and capabilities, imagine an (e-)conference (or any other form of real-time cooperation) where one user performs an action in his client (eg. clicking on a spectrum in TOPCAT) and other users in their completely different clients see the same information but in a form native for their clients (eg. spectrum sent by TOPCAT plotted in SPLAT) in the same time.

2.4.1 All SAMP Spectra to the Same Plot Window

In the previous versions of SPLAT, all spectra received via the SAMP protocol got opened in their own plot windows. Very often, it would be useful or even necessary for analysis to have them plotted in the same window, but for this, the user had to do it manually.

As stays in the title, the upcoming version will be able to do this for the user automatically. We added new checkbox under the 'Interoperability' menu that can be used to switch between the current state (each spectrum to its own plot window) and plotting all the spectra received from SAMP to their common plot window.

Technically it is done by adding the new 'Source type' attribute into the spectrum's I/O process and mapping every plot window to its initial spectrum's source type (in case of SAMP, this reference is removed from every possibly opened 'SAMP-containing' plot window when user disables this feature, so it allows to plot one bulk of spectra received from SAMP to one window and another bulk to another window). Then, when the new spectrum is received (in this case from SAMP, but this may vary in the future, read more) and the feature is switched on, SPLAT will first look for an existing plot window that is mapped to SAMP source type and if found, it will plot the spectrum into it (and properly resizes it as usual).

As said, this feature was added using the new attribute for spectrum's I/O process defining the spectrum's source. So far this attribute is bind just to SAMP protocol, but it allows simple modifications of I/O behaviour based on the spectrum's source in the future by simple writing the extra logic for non-default source type.

3 Bug Fixes

Since the SPLAT is based on Starlink's libraries, which has been in use for many years, discovery of new bugs is limited to less critical parts of SPLAT. Yet, for some users, they make the SPLAT unusable. We are now in the process of removing those we know about, but we of course welcome new bug reports.

3.1 SSAP Query Window

In some cases, when user tried to open the SSAP Query window to search for spectra via SSAP protocol, the only thing, that he saw was a window with a list of queried SSAP servers. He couldn't enter any values or parameters.

This was caused by a `NullPointerException` in SSAP Query window initialization. After hotfixing it and consulting with Margarida Castro Neves of Heidelberg University, the problem was located in a `PropertyChangeSupport` property initialization. Since the `JPanel` already has it, it was fixed by removing it and refactoring the rest of the code to use the `JPanel`'s one.

4 Features for Developers and Experimentators

It will still take some time before the new version will be released, but if anyone would like to try it now or participate on testing, here is the essential build[12] 'HOWTO' and short description of its innovations.

4.1 Java 1.7 Compatibility

By Java 1.7 compatibility is ment its building compatibility (runtime compatibility is generally guaranteed by Java itself). Until now, Java 1.6 has been used to build SPLAT. Building with 1.7 was causing build failures for several reasons described below.

Still, building with Java 1.7 keeps SPLAT compatible with Java 1.6 runtime environment.

4.1.1 JSAMP's Jar Sign Algorithm

When built with 1.7, SPLAT was throwing `java.lang.SecurityException: invalid SHA1 signature file digest` exception during its start. The

[12] As said before, our current primary goal is to unify the development since the SPLAT's source codes are located in several repositories. Version described in this article can be found at Github: `https://github.com/and146/SPLAT-ARI/`

reason is the changed sign algorithm for jars in Java 1.7. This problem can be fixed by adding the `digestalg="SHA1"` parameter to `jarsigner`, which will force using the previous sign algorithm.

Since STARJAVA/SPLAT is being built by the Ant utility, the `digestalg` parameter should come to the JSAMP's build.xml file. It also requires modification of STARJAVA/SPLAT's customized Ant (added some STARJAVA-special Ant targets) in the sense of adding the support for these new parameters to its source, since they are available from higher versions of Ant that STARJAVA contains.

4.1.2 Java 1.6 Backward Compatibility

By default, SPLAT's binaries are runnable under a version of Java that has been used to compile it. Keeping the backward compatibility therefore required adding the `target` parameter for `javac` in build.xml file of SPLAT and all of its required tools.

4.1.3 Other Modifications

This section closely relates to 4.2 and usage of a build script. By default, this script builds the entire STARJAVA package so when javac 1.7 is used, it throws errors in other tools as well (yet some of them are not required to successfully build the SPLAT). This is a short list of modifications required to successfully use the javac 1.7, yet none of them is so far a part of its original distribution.

As first, Frog was using a Sun's proprietary (not Java-standard) library for exporting the image (`JPEGCodec` and `JPEGImageEncoder`), which was removed in Java 1.7. This needed to be rewritten to use its standard equivalent (`ImageIO`).

Then, TOPCAT is using in `SyntheticColumnQueryWindow` class getters and setters for a parameter called `type`. Unfortunately, this class extends the Java's `java.awt.Windows` which since 1.7 contains an internal enum called `Type`. This was causing the `incompatible return data type of uk.ac.starlink.topcat.SyntheticColumnQueryWindow.getType()` exception during the build with javac 1.7.

And finally, on some systems the build could fail on invalid encoding of TAMFITS's source. This was fixed by explicit adding the `encoding="iso-8859-1"` parameter to its build.xml file.

4.2 Building Using the Buildscript

Basically, until now the only way how to build SPLAT was (at least in some way) following the instructions in the original STARJAVA's README file. To automatize this (and to add some more 'shortcuts'), the build script was created.

The build script (`_builder.sh`) can be used on every Linux system and beside the automation of STARJAVA's build process with some system-checks included it also allows quite user-friendly way to build only a subpart of STARJAVA or to use a specific version of Java (for full list of its capabilities, run it with the `--help` parameter).

4.2.1 Source

As said before, our current primary goal is to unify the development since the SPLAT's source codes are located in several repositories. Version described in this article can be found at Github: `https://github.com/and146/SPLAT-ARI/`.

4.2.2 Prerequisites

Checks for most of them are included in the build script, so just shortly here:

- Java Advanced Imaging (JAI) (`http://download.java.net/media/jai/builds/release/1_1_3/INSTALL.html`)
- Java Development Kit (JDK) >= 1.6 (`http://www.oracle.com/technetwork/java/javase/downloads/index.html`)

4.2.3 Build

The actual build is quite straightforward, so again, very shortly here (demonstrated on GNU/Linux):

```
# first, download the source
(starjava_parent) $ git clone
git://github.com/and146/SPLAT-ARI/

# enter the newly created directory and run the build
 script (starjava_parent) $ cd SPLAT-ARI
(SPLAT-ARI) $ ./_builder.sh

# the binaries should be located in
(starjava_parent)/bin:
(SPLAT-ARI) $ ../bin/bin/splat/splat
```

5 Conclusion

Being more focused on user experience should allow to current users to work with SPLAT more effectively while for more reserved astronomers that so far had a reason to avoid SPLAT, we hope we can provide a reason or reasons to give it a try.

In the near future, we are planning to work on (beside strictly VO-related parts, that are currently being developed at Heidelberg University) user-defined line lists, working with spectrum's header information in plot window or supporting more file formats and at least some of these changes should be a part of the next release as well.

Acknowledgements. The following two grants are acknowledged for the financial support provided for this research: Grant Agency of the Czech Republic - GACR P103/13/08195S, by the Development of human resources in research and development of latest soft computing methods and their application in practice project, reg. no. CZ.1.07/2.3.00/20.0072 funded by Operational Programme Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic and grant SV 4603351.

References

1. Pence, W.D.: FITS Extension Names (2012), http://fits.gsfc.nasa.gov/xtension.html (cited March 19, 2013)
2. International Virtual Observatory Alliance, International Virtual Observatory Alliance (2013), http://ivoa.net/ (cited March 22, 2013)
3. Joint Astronomy Centre, Starlink (2012), http://starlink.jach.hawaii.edu/starlink (cited March 18, 2013)
4. Donahue, M., Kimball, T.: FITS File Format. In: HST Data Handbook (1997), http://www.stsci.edu/documents/dhb/web/c02_datafiles.fm2.html (cited March 19, 2013)
5. Taylor, M., Boch, T., Fitzpatrick, M., Allan, A., Fay, J., Paioro, L., Taylor, J., Tody, D.: SAMP - Simple Application Messaging Protocol (2012), http://www.ivoa.net/Documents/SAMP/20120411/REC-SAMP-1.3-20120411.html (cited March 22, 2013)
6. Draper, P.W.: Starlink SPLAT-VO (2013), http://star-www.dur.ac.uk/~pdraper/splat/splat-vo/ (cited March 18, 2013)
7. Skoda, P.: Optical Spectroscopy with the Technology of Virtual Observatory (2011), http://arxiv.org/abs/1112.2779 (accessed April 01, 2013)

Mobile Sensor Data Classification Using GM-SOM*

Petr Gajdoš, Pavel Moravec, Pavel Dohnálek, and Tomáš Peterek

Abstract. The paper uses a previously-introduced modification of standard Kohonen network (SOM), called GM-SOM. This approach uses partitioning the problem in case of insufficient resources (memory, disc space, etc.) and parallel processing of input data set to process all input vectors at once, with the use of modern multi-core GPUs to achieve massive parallelism. The algorithm pre-selects potential centroids of data clusters in the first step and uses them as weight vectors in the final calculation. In this paper, the algorithm has been demonstrated on a new UCI HAR dataset, representing activities recorded by smartphone sensors, which are prone to random noise due to the sensor behavior. Moreover the separation of classes is not linear, which introduces additional complexity and makes it hard to process the data by linear algebra methods.

Keywords: accelerometer, activity recognition, gyroscope, Kohonen Network, parallel calculation.

1 Introduction

Many algorithms for pattern recognition were proposed for the human activity recognition: the subspace clustering method [10], hierarchical recognition model

Petr Gajdoš · Pavel Moravec · Pavel Dohnálek
Department of Computer Science, FEECS, VŠB – Technical University of Ostrava,
17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic
e-mail: {petr.gajdos,pavel.moravec}@vsb.cz

Petr Gajdoš · Pavel Moravec · Tomáš Peterek
IT4 Innovations, Centre of Excellence, VŠB – Technical University of Ostrava,
17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic

* This work was supported by the Bio-Inspired Methods: research, development and knowledge transfer project, reg. no. CZ.1.07/2.3.00/20.0073 funded by Operational Programme Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic and IT4Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 supported by Operational Programme 'Research and Development for Innovations' funded by Structural Funds of the European Union and state budget of the Czech Republic.

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 487–496.
DOI: 10.1007/978-3-319-00542-3_48

[4], he Naive Bayes classifier [9], the Dynamic Bayesian Network [7], the Binary Decision Tree [8], Kernel Discrimination Analysis [5] and many other.

With the massive boom of GPU-based calculations, massive parallelism, memory considerations, simplicity of algorithms and CPU-GPU interaction have yet again to play an important role. Recently, we have introduced the GPU-based modification of classic Kohonen's self-organizing maps (*SOM*), which allows us to dynamically scale the computation to fully utilize the GPU-based approach and tested its performance on intruder data set [3].

In this paper, we will use our modification of Kohonen's algorithm for the classification of data obtained from UCI HAR Dataset, introduced in [1]. The dataset contains real-live data collected from smartphone's orientation sensors, which are by themselves prone to random noise. The separation of classes is not linear (especially between sitting and standing, walking upstairs/downstairs), which introduces additional complexity and makes it hard to process the data by linear algebra methods, but feasible by some of the bioinspired methods, such as the used self-organizing map.

The paper is organized as follows: in second chapter we mention classic SOM networks and describe the basic variant we have used. In third chapter we describe our approach and provide the calculation algorithm. The next chapter introduces experimental data we have used and the final chapter before conclusion presents the comparison of results provided by our method with classic SOM calculation.

2 Self-organizing Neural Network

In following paragraphs, we will describe the used Kohonen self-organizing neural networks (self-organizing maps – *SOM*) variant. First self-organizing networks have been proposed in the early 1970s by Malsburg who was succeeded by Willshaw. SOM was proposed by Teuvo Kohonen in in the early 1980s and has been improved on by his team. The summary of the basic SOM method may be found in [6].

The self-organizing map is one of the common ways to represent and visualize data, and how to map the original dimension and structure of the input space into another – usually lower-dimensional – output space structure.

The basic idea of SOM is derived from the human brain, which uses internal 2D or 3D representation of information. We can imagine the input data as vectors, which are recorded in neural network. Only the adjacent neurons are interconnected.

Besides of the input layer the SOM contains only the (competitive) output layer. The number of inputs equals to the dimension of the input space. Every input is connected with each neuron in the grid, which also serves as an output (each neuron in the grid is a component of the output vector). With growing number of output neurons, the quality of the coverage of input space increases, but unfortunately so does the computation time.

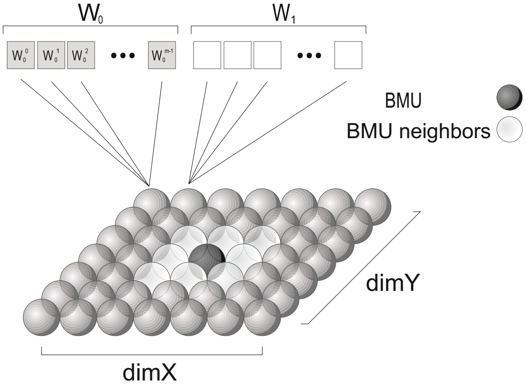


Fig. 1 Kohonen network structure

SOM may also be used as a classification or clustering tool that can find clusters of input data which are intrinsically closer to each other than to the rest of the clusters.

All experiments in this paper are based on the following definition of the SOM:

- The SOM is initialized as a fixed-topology network. The dimensions are represented by $dimX$ and $dimY$ variables of a 2-dimensional topology.
- V^m is a m-dimensional input vector.
- W^m is a m-dimensional vector of weights.
- $N = dimX * dimY$ is the number of neurons and every neuron $n \in < 0, N-1 >$ has its weight vector W_n^m
- The value r is the neighborhood radius, is initialized to $min(dimX, dimY)/2$ and systematically reduced up to the unit distance.
- All weight vectors are updated after processing of each of the input vectors.
- We know the number of epochs e from the beginning.

The Kohonen algorithm consists of following steps:

1. **Initialization of the network**
   All weight vector coordinates are set to a randomly-generated or pre-calculated values, based on the used initiation algorithm. The learning factor η, $0 < \eta < 1$, which determines the weight adaptation speed is set to a value slightly below 1 and monotonically decreases to 0 during the learning process. Because of this, the weight adaptation is fastest in the beginning, slowing down in the end.
2. **Introducing the input vector**
   Introduce k training input vectors $V_1, V_2, \ldots, V_k$, which are learned in random order.
3. **Calculation of the distance**
   The distance of each of the neurons to the input vector is calculated.

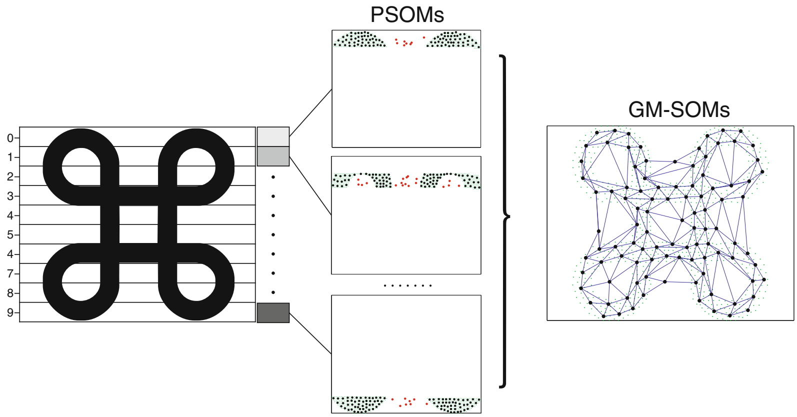


Fig. 2 An illustrative scheme of GM-SOM method. The input vectors are divided into ten parts in this case.

4. **Choice of closest neuron**
   The closest neuron for introduced input is selected. If the neuron is selected in a competition, we calculate the neighborhood around it and for each neuron in this neighborhood the weights are going to change. The size, shape and the degree of influence of the neighborhood are the network parameters, with the later two decreasing during the iterations of the learning algorithm.
5. **Adjustment of weights**
   The weights of closest neuron and neurons in its neighborhood are adapted in following way:

$$W_{ij}(t+1) = W_{ij}(t) + \eta(t)h(v,t)(V_i - W_{ij}(t)) \tag{1}$$

   where $i = 1, 2, \ldots, dimX$ a $j = 1, 2, \ldots, dimY$ and the radius r of neuron's local neighborhood is determined by adaptation function $h(v)$.
6. **The steps from step 2 on are repeated until the number of epochs e is reached.**

To generate the best organization of neurons in clusters, a large neighborhood and a high influence of introduced input are used in the beginning. Later, the primary clusters arise and the neighborhood and learning factor are reduced. Also the $\eta \to 0$, and with each iteration the changes become less significant.

3 Our GM-SOM Method

The basics of SOM learning process have already been described in previous section. The following text contains a short description of our method, which was firstly introduced in [2], that leads to results close to those of the classic SOM. For better

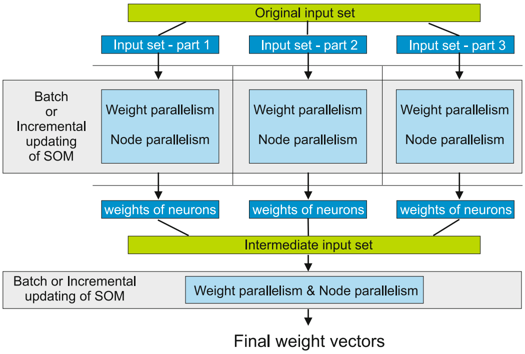


Fig. 3 GM-SOM: The SOM parallelization in the GM-SOM method

understanding, please refer also to the Figure 2 for clarification of our method. The method has been named Global-Merged SOM, which suggests, that the computation is divided into parts and their results are merged to obtain the final solution. The whole process of GM-SOM can be described in following steps:

1. **Input set split.** The input vectors set is divided into a pre-defined number of parts. With a higher number of pars, the precision of the GM-SOM increases, on the other hand, a high number of parts means a larger set of vectors in the final phase of computation process. As a result, the number of parts should be usually determined from the number of instances, typically $p * N \ll k$, where p is the number of parts, N is the number of neurons and k is the number of input vectors. The split of input vectors into separate parts should not influence the final result.
2. **Computation in individual parts.** Classic SOM method is used for every part. We will use the acronym *PSOM* from now on to indicate SOM, which is computed on a given part for simplicity sake. All PSOMs are executed with same settings (the initial weight vectors distribution, the number of neurons, etc.) The division speeds up the parallel PSOM computation on the GPU. Further, the number of epochs in this case may be lower than that used for the processing of input set by a single SOM. This is represented by a f factor (set to $\frac{1}{3}$) in our experiments.
3. **Merging of the calculated partial results.** The Weight vectors computed in each PSOM – corresponding to the neurons with at least a single hit from the set of input vectors for given part – are used as input vectos in the final merging in the final phase of GM-SOM. The neurons without hits and their weight vectors have light gray color in Figure 2. In fact, a higher threshold value could be used when necessary. A merged SOM is then computed and output weights vectors are the final result of the GM-SOM method.

The main difference between the well known batch SOM algorithms and our algorithm and is the complete independence of separate parts, each updating different

PSOMs. We can apply different settings of PSOMs and the final merging phase, e.g. a more dense neuron network can be used in case of a larger input set. Also, we can incrementally update the GM-SOM – any additional set of input vectors may be processed by a new PSOM as a separate part and the final SOM will be recalculated after that.

4 Data Collection

The new UCI HAR Dataset [1] represents data indicating several basic types of user actions whilst carrying a mobile device. Significant information from device sensors have been recorded for different people and then aggregate data were generated, resulting in 561 features.

Volunteers (30) aged between 19 and 48 years were employed in this measurements. Each volunteer alternated six different states (walking, walking upstairs, walking downstairs, sitting, standing, laying). Data have been recorded by a single type of smartphone (Samsung Galaxy II), affixed on the volunteer's waist. This category of smartphones is typically equipped with a 3-axial linear accelerometer and with 3-axial gyroscope. The volunteers have been also video-recorded to label the data manually. The signals from the accelerometer and the gyroscope have been noise-filtered and then sampled in fixed width sliding windows of 2.56 seconds and 50% overlapped.

The accelerometer signals contain gravitational and body movement components, therefore a Butterworth low-pass filter with 0.3Hz cut of frequency was applied to separate these components. From each window a feature vector has been obtained by calculating the time-frequency spectrum.

The training set contains 7352 instances (21 subjects: 1, 3, 5, 6, 7, 8, 11, 14, 15, 16, 17, 19, 21, 22, 23, 25, 26, 27, 28, 29 and 30) and the testing set 2947 instances (9 subjects: 2, 4, 9, 10, 12, 13, 18, 20 and 24). The number of vectors is individual for each of these subjects and ranges from 281 to 409 vectors (for all 6 classes).

5 Experiments

The experiments have been performed on 8×8 and 16×16 hexagonal SOM grid. 100 iterations have been used for both the original SOM network and the final calculation. The resulting classes for the latter case are shown in Figures 4 and 5. Cells with a dot in the center indicate neurons with hits from a single class of the training set and black cells neurons without any hits.

In both cases here is a problem with separation of classes 4 and 5 (sitting and standing), which are represented by light and dark blue colors in the bottom left area of both networks, but this had to be expected, since the smartphone was affixed to waist and the data provided by its internal sensors would be almost the same in most cases.

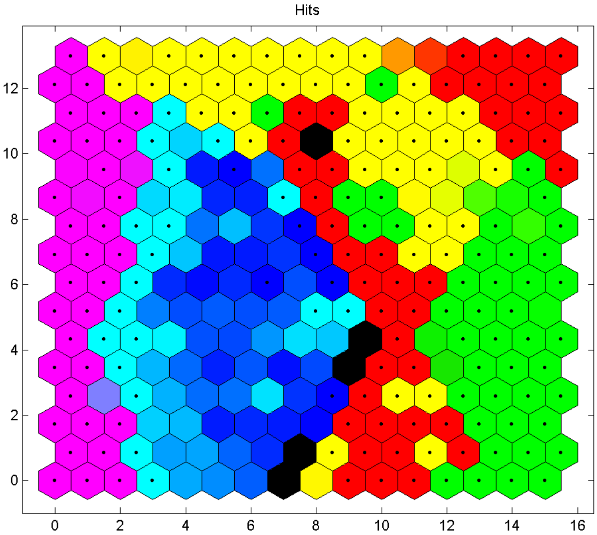


Fig. 4 Resulting Kohonen network for classic SOM

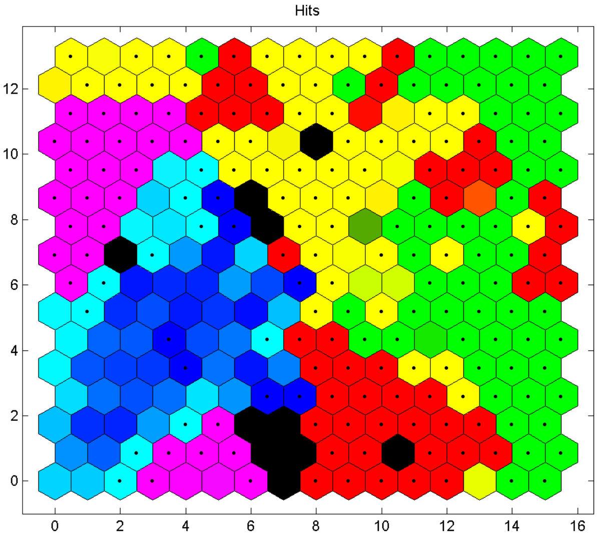


Fig. 5 Resulting Kohonen network for GM-SOM

Table 1 Overlap of individual classes based on testing set data for classic SOM

	1	2	3	4	5	6
1	100%	2.92%	0.18%	0%	0%	0%
2	2.92%	100%	1.55%	0%	0%	0%
3	0.18%	1.55%	100%	0%	0%	0%
4	0%	0%	0%	100%	48.79%	0.45%
5	0%	0%	0%	48.79%	100%	0%
6	0%	0%	0%	0.45%	0%	100%

Table 2 Overlap of individual classes based on testing set data for GM-SOM

	1	2	3	4	5	6
1	100%	0.29%	0.14%	0%	0%	0%
2	0.29%	100%	0.93%	0.09%	0%	0%
3	0.14%	0.93%	100%	0%	0%	0%
4	0%	0.09%	0%	100%	50.63%	1.04%
5	0%	0%	0%	50.63%	100%	0%
6	0%	0%	0%	1.04%	0%	100%

The Tables 1 and 2 show the overlap of individual classes.

In the case of GM-SOM, the whole training data collection was divided into several parts for individual experiments. Since the data set was tagged by the individual people, we made the division for individual P-SOMs based on this information. In each experiment, every part represented an input data set for SOM. Then all partial results (weights of neurons) were merged together and became a new input set for final computation of SOM.

Table 3 Number of classes classes assigned to each neuron (out of 256) based on testing set data

Classes per neuron	classic SOM	GM-SOM
0	5	10
1	181	185
2	70	60
3	0	1
4	0	0
5	0	0
6	0	0

Table 3 shows the number of classes per each of the neurons of the 16×16 hexagonal SOM grid.

6 Conclusion

The need of parallel computation of SOM lead us previously to develop a new method, which was tested on several types of data collections for both performance and quality. Although it has some common features with some hierarchical and batch SOM algorithms, it has its unique properties, which make it easier to implement on the GPU.

In this paper, we have used GM-SOM to process the accelerometer data. The results presented in previous section show that both the classic SOM and our modification performs well in most cases, however unlike the original paper for this dataset it fails to clearly separate the sitting and standing positions. Since the measurement setup provides similar data and we analyzed the individual vectors and not the time series, this is quite understandable.

References

1. Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes-Ortiz, J.L.: Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In: Bravo, J., Hervás, R., Rodríguez, M. (eds.) IWAAL 2012. LNCS, vol. 7657, pp. 216–223. Springer, Heidelberg (2012)
2. Gajdos, P., Moravec, P.: Two-step modified SOM for parallel calculation. In: Pokorný, J., Snásel, V., Richta, K. (eds.) DATESO, CEUR Workshop Proceedings, vol. 567, pp. 13–21 (2010), `CEUR-WS.org`
3. Gajdoš, P., Moravec, P.: Intruder data classification using GM-SOM. In: Cortesi, A., Chaki, N., Saeed, K., Wierzchoń, S. (eds.) CISIM 2012. LNCS, vol. 7564, pp. 92–100. Springer, Heidelberg (2012)
4. Khan, A., Lee, Y., Lee, S.: Accelerometer's position free human activity recognition using a hierarchical recognition model. In: 2010 12th IEEE International Conference on e-Health Networking Applications and Services (Healthcom), pp. 296–301 (2010), doi:10.1109/HEALTH.2010.5556553
5. Khan, A., Lee, Y.K., Lee, S., Kim, T.S.: Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis. In: 5th International Conference on Future Information Technology (FutureTech), pp. 1–6 (2010), doi:10.1109/FUTURETECH.2010.5482729
6. Kohonen, T.: Self-Organizing Maps, 2nd edn. Springer, Berlin (1997)
7. Mirarmandehi, N., Rabiee, H.: An asynchronous dynamic bayesian network for activity recognition in an ambient intelligent environment. In: 5th International Conference on Pervasive Computing and Applications (ICPCA), pp. 20–25 (2010), doi:10.1109/ICPCA.2010.5704069
8. Parkka, J., Cluitmans, L., Ermes, M.: Personalization algorithm for real-time activity recognition using pda, wireless motion bands, and binary decision tree. IEEE Transactions on Information Technology in Biomedicine 14(5), 1211–1215 (2010), doi:10.1109/TITB.2010.2055060

9. Yang, X., Dinh, A., Chen, L.: Implementation of a wearerable real-time system for physical activity recognition based on naive bayes classifier. In: 2010 International Conference on Bioinformatics and Biomedical Technology (ICBBT), pp. 101–105 (2010), doi:10.1109/ICBBT.2010.5479000
10. Zhang, H., Yoshie, O.: Improving human activity recognition using subspace clustering. In: 2012 International Conference on Machine Learning and Cybernetics (ICMLC), vol. 3, pp. 1058–1063 (2012), doi:10.1109/ICMLC.2012.6359501

Tensor Modification of Orthogonal Matching Pursuit Based Classifier in Human Activity Recognition*

Pavel Dohnálek, Petr Gajdoš, and Tomáš Peterek

Abstract. Human physical activity monitoring is a relatively new problem drawing much attention over the last years due to its wide application in medicine, homecare systems, prisoner monitoring etc. This paper presents Orthogonal Matching Pursuit as a method for activity recognition and proposes a new modification to the method that significantly increases the recognition accuracy. Both methods show promising results in both total recognition and differentiation between certain activities even without the necessity of prior data preprocessing. The methods were tested on raw sensor data.

Keywords: orthogonal matching pursuit, activity monitoring, pattern recognition, raw data.

1 Introduction

Processing large datasets generated by processes of the real world represents one of the most important challenges in data mining and information

Pavel Dohnálek · Petr Gajdoš
Department of Computer Science, FEECS,VŠB – Technical University of Ostrava, 17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic
e-mail: {pavel.dohnalek,petr.gajdos}@vsb.cz

Pavel Dohnálek · Petr Gajdoš · Tomáš Peterek
IT4 Innovations, Centre of Excellence, VŠB – Technical University of Ostrava, 17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic

* This article has been elaborated in the framework of the IT4Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 supported by Operational Programme 'Research and Development for Innovations' funded by Structural Funds of the European Union and state budget of the Czech Republic and supported by the SGS in VŠB Technical University of Ostrava, Czech Republic, under the grant No. SP2013/70.

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 497–505.
DOI: 10.1007/978-3-319-00542-3_49

retrieval. Ever-growing datasets present the opportunity of discovering previously unobserved patterns. Extracting interesting knowledge from experimental raw datasets, measurements or observations and understanding complex data has their use in various research areas, such as medicine, bio-chemistry, physics, industry, etc. In the context of processing bio-signals and pattern matching in general, several classes of algorithms have been developed. Although data analysis is a major issue in data mining, general processing of large data can be extremely time consuming. All known methods and algorithms solve the problem with respect to the precision and recall.

One of the applications of these methods is human activity recognition. In this paper, Orthogonal Matching Pursuit (OMP) is shown to be capable of recognizing 12 different physical activities such as lying, sitting, standing, walking, running, cycling and others. Also, its modification significantly improving classification accuracy is proposed. The methods have been tested on the Physical Activity Monitoring for Aging People 2 (PAMAP2) dataset.

1.1 Current Approaches in Activity Recognition

Generally, in activity recognition authors attempt to recognize static states (lying, sitting, standing, etc.), dynamic states (walking, running, etc.) and/or transition states (i.e. standing to walking). The data is usually preprocessed to achieve higher recognition accuracy while keeping the computation times at near real-time values [7]. Classification methods are based on the C4.5 algorithm [8], Naive Bayes classifiers [14], Classification And Regression Tree (CART) [9], Adaptive Neuro Fuzzy Interference System (ANFIS) [4], Naive Bayes (NB) [6], k-Nearest Neighborhood (kNN)[5], Iterative Dichotomiser 3 (ID3) [2], Nearest Neighborhood (NN) [10], sparse approximation [15, 16] and others. In this paper, sparse approximation is the idea behind the used classifier.

2 Sparse Approximation for Activity Recognition

In order to use the sparse approximation of a signal for classification, the sparse coefficients must first be computed. To do this, a modified version of the OMP algorithm is used and compared to the standard OMP defined in [1]. Originally, OMP is a greedy algorithm that selects vectors from the entire training set that are the closest match to the signal being classified (the query signal). The training set can be represented as an $m \times n$ real-valued matrix A, where m is the length of both training and query signal and n is the number of training samples. Also, $n = t \times c$, where t is the number of training samples for a given class and c is the number of classes. The OMP algorithm requires the training set to contain the same number of training samples for each class. Therefore, the training matrix has the form of $A = [a_{11}, a_{21}, ..., a_{t1}, a_{12}, ..., a_{tc}]$, where $a_{ij}, i = 1..t, j = 1..c$ is the

ith training sample of class j and length m. Each column of every training matrix is normalized to unit length.

The proposed modification divides the training set into c submatrices, each containing only the training vectors of the kth class, with $k = 1..c$. The matrices are arranged as a tensor $T \in \mathbb{R}^{m \times t \times c}$. The sparse representation is then computed for each class, yielding c sparse representations of the query signal. The sparse coefficients of the kth sparse representation are obtained by finding the sparse solution to the equation

$$y = T_k s_k, \tag{1}$$

where $y \in \mathbb{R}^m$ is the query vector, $T_k \in \mathbb{R}^{m \times t}$ is the kth class training matrix and $s \in \mathbb{R}^t$ is the sparse coefficient vector. The stopping criterion in the implementation is reaching the sparse coefficient vector of the desired dimensionality. Due to the modification to the OMP algorithm, the process computes the sparse coefficients only from the training matrix of a given class. Therefore, the increased accuracy of the approach depends on a reasonable distribution of the best-fitting training vectors to the c classes.

2.1 Classification

To classify the query signal vector, a strategy of computing the residual value from the difference between the query vector and its sparse representation converted into the vector space of the training matrix vectors is employed. This is performed for each class resulting in c residuals. The classification is then based on the minimum residual. Formally, the classification problem can be stated as follows:

$$arg\, min\, r_k(y) = \|y - T_k s_k\|_2. \tag{2}$$

The same approach, although solved by a slightly different algorithm, is also used in [16] and [15].

Computing the residuals is generally not computationally expensive and can be performed in real time, depending on the size of the training matrices. Only very large training matrices can slow the process down significantly.

3 The PAMAP2 Dataset

The PAMAP2 dataset contains data from measuring nine healthy human subjects, each subject wearing three inertial measurement units (IMUs) by Trivisio, Germany and a heart rate monitor. Each of the three IMUs measures temperature and 3D data from accelerometer, gyroscope and magnetometer. The data is sampled at 100 Hz and transmitted to PC via 2.4 GHz wireless network. Subjects wore one IMU on the dominant wrist, one on the dominant ankle and one on the chest. Detailed information on the dataset can be found in [12] and [13].

The methods have been tested on a single test subject, labeled in the dataset as *subject*101. Data of this subject consists of 376417 measurements, each containing 54 values. The meanings of the values are described in the dataset documentation. Some values can be missing, indicated with the NaN (Not a Number) value. Every NaN value was replaced with a zero.

The activities performed were lying, sitting, standing, walking, running, cycling, Nordic walking, ascending stairs, descending stairs, vacuum cleaning, ironing and rope jumping. Transition activities were discarded. Since some measurements contain only NaN values, these measurements were discarded as well. In total, 249957 activity measurements were used. From each measurement, irrelevant values, that is the orientation of each IMU and timestamp, were removed. Since most heart-rate values were NaNs, they were also removed. As a result, each measurement contains 39 values.

4 Experiments

The following section summarizes the experiments performed and the resulting recognition accuracy, pointing where exactly is the proposed method more accurate then the original OMP and vice versa.

4.1 Tasks Being Solved

The experiments compare the accuracy of the original OMP and the proposed modification. To achieve the goal, two subtasks of the original PAMAP2 dataset tasks have been undertaken. The first subtask is to distinguish and recognize every activity from any other (referred to as *All activity recognition task*). Therefore, the problem is one of classification into 12 classes. The other subtask groups activities together.

It is called the *Intensity estimation task* and its purpose is to group the activities for recognition the effort required to perform them. This task reduces the problem to three classes (effort levels): light, moderate and vigorous. Activities considered to be light are *lying*, *sitting*, *standing*, and *ironing*. Moderate activities are *walking*, *cycling*, *Nordic walking*, *descending stairs* and *vacuum cleaning*. Activities requiring vigorous effort are *running*, *ascending stairs* and *rope jumping*.

4.2 Training Dataset

The training dataset has been formed in the following manner: since the OMP algorithm requires the training set to contain the same number of training vectors for each class, it is first found which class contains the fewest measurements and the measurements are counted. The count is then multiplied by training dataset size percentage, yielding the number of training vectors for each class t (always rounded up if the result is not an integer). Each training matrix is then chosen to be first t vectors of a corresponding class in the whole dataset. The remaining vectors can then be used as the testing set.

In the experiments there are two variables greatly affecting the resulting classification accuracy - the size of training dataset and the number of sparse coefficients used for classification. In total, 36 experiments have been performed, 18 for the original OMP, 18 for the proposed modification. For each implementation, 3 training dataset sizes were chosen, given in percentage of the entire dataset: 10%, 25% and 50%. For each size there are 6 experiments, each with different number of sparse coefficients to be computed (1, 5, 10, 15, 20, and 25). A set of 3600 measurement vectors was used for classification, 300 vectors for each class. The vectors are the first vectors in the testing set for the corresponding class. The data has not been preprocessed in any way, the experiments were performed on the raw dataset data.

4.3 Results

Tables 1, 2 and 3 show the comparison between the modified (left value) and original (right value) OMP for all classes and sparse coefficient numbers. In all tables the numbers are percentual values. It can be clearly seen that results for both methods are the same if sparse coefficient number is set to 1. That is the expected behavior since both algorithms only pick a single training vector in their greed-driven search. Bold font marks the better value. All three tables demonstrate a high increase in accuracy for *Nordic walking*, *ascending stairs* and *descending stairs* activities. These activities proved to be the most difficult ones to recognize using OMP. On the other hand, *standing* and *walking* are activities better differentiated by the standard OMP. In general the tables show results that are favorable to the modified OMP. The algorithm brings considerably better accuracy for activities that are the hardest to recognize for standard OMP while introducing less significant errors to other activities, generally increasing the total recognition accuracy.

Table 4 presents the total accuracy across all classes for given training dataset sizes and numbers of sparse coefficients. The modified OMP performs better in all cases except for the 10% training dataset case with 5, 10 and 15 sparse coefficients. 50% training dataset then shows a significant increase in accuracy (up to 12.87%).

Values in tables 5 through 7 represent the accuracy of the *Intensity estimation* task for each of its classes. While the modified algorithm is not as accurate in recognizing light effort activities, generally losing to the standard OMP by several percentage points, it performs significantly better for both moderate and vigorous effort activities if the training dataset is at least 25% of the whole dataset. The best improvement can be seen for vigorous effort activities in 25% training set with 25 sparse coefficients (28.78%), while the biggest decrease happens for the light effort with the same settings (19.92%).

For 10% training dataset, the results for the modified OMP can be seen as comparable to slightly unfavorable.

While it is a common practice in activity recognition to merge *sitting* and *standing* activities into a single class (as was done, for instance, in [12], [3] and [11]) due to difficulties of distinguishing the two from each other, it can be seen that both versions of OMP are capable of recognizing the activities with great confidence and reliability, providing almost perfect results for 50% training dataset.

Table 1 Accuracy comparison between the two OMP algorithms for the training dataset set to 50%

Activities	Sparse coefficients (training dataset 50%)											
	1		5		10		15		20		25	
lying	100	100	100	100	100	100	100	100	100	100	100	100
sitting	100	100	100	100	100	100	100	100	100	100	100	100
standing	100	100	100	100	100	100	100	100	**100**	99,33	**99,00**	93,00
walking	100	100	100	100	100	100	100	100	99,33	**100**	98,67	**99,33**
running	100	100	98,00	**99,67**	97,33	**98,33**	**94,67**	91,67	**89,33**	83,00	**82,00**	76,00
cycling	100	100	100	100	100	100	**100**	97,67	**100**	86,00	**99,00**	69,33
Nordic walking	69,00	69,00	**84,00**	69,00	**87,33**	67,67	**89,33**	61,33	**81,00**	60,00	**67,00**	54,67
ascending stairs	65,67	65,67	**65,67**	65,00	**83,00**	59,67	**85,67**	46,67	**82,00**	41,33	**79,00**	36,67
descending stairs	35,67	35,67	**52,33**	35,67	**66,00**	32,00	**62,00**	22,33	**53,33**	17,00	**51,67**	14,67
vacuum cleaning	94,67	94,67	87,00	**94,67**	80,00	**89,00**	**74,33**	65,00	**74,00**	52,67	**73,67**	43,67
ironing	100	100	100	100	97,00	**100**	87,33	**97,67**	84,67	**90,33**	83,67	**87,67**
rope jumping	100	100	100	100	100	100	99,00	**99,67**	95,33	**99,00**	93,00	**97,33**

Table 2 Accuracy comparison between the two OMP algorithms for the training dataset set to 25%

Activities	Sparse coefficients (training dataset 25%)											
	1		5		10		15		20		25	
lying	100	100	100	100	100	100	100	100	100	100	100	100
sitting	100	100	100	100	100	100	100	100	100	100	100	100
standing	97,67	97,67	92,33	**97,67**	77,00	**97,67**	44,67	**97,67**	14,67	**97,67**	4,67	**96,00**
walking	99,33	99,33	**99,67**	99,33	99,67	99,67	99,33	99,33	98,00	98,00	93,67	**96,67**
running	99,33	99,33	**100**	99,33	**100**	98,67	**98,33**	92,00	**97,67**	83,00	**93,67**	72,67
cycling	99,67	99,67	94,00	**99,67**	81,00	**99,67**	66,33	**92,00**	52,00	**74,67**	41,33	**57,67**
Nordic walking	31,33	31,33	**62,67**	31,67	**53,67**	27,67	**46,33**	19,00	**39,33**	13,33	**31,00**	15,00
ascending stairs	22,33	22,33	**72,67**	23,00	**81,00**	29,67	**85,67**	31,00	**90,33**	28,00	**90,33**	25,33
descending stairs	78,33	78,33	73,00	**79,00**	64,00	**79,33**	43,33	**57,67**	34,00	**38,00**	27,33	**34,00**
vacuum cleaning	85,33	85,33	**100**	85,33	**81,33**	75,00	**74,00**	57,67	**69,00**	40,00	**65,67**	40,67
ironing	100	100	100	100	100	100	**100**	98,67	**98,00**	92,00	**96,00**	84,33
rope jumping	100	100	100	100	**100**	99,67	99,67	99,67	99,33	**99,67**	**99,33**	99,00

Table 3 Accuracy comparison between the two OMP algorithms for the training dataset set to 10%

Activities	Sparse coefficients (training dataset 10%)											
	1		5		10		15		20		25	
lying	100	100	100	100	100	100	100	100	100	100	100	100
sitting	80,00	80,00	**93,67**	80,67	**100**	85,33	**99,67**	90,33	**94,67**	87,67	**93,33**	87,33
standing	100	100	100	100	100	100	**100**	98,00	**100**	89,67	**100**	82,33
walking	53,33	53,33	43,00	**52,00**	37,00	**50,00**	29,33	**47,67**	31,00	**40,00**	27,33	27,33
running	89,67	89,67	**95,00**	92,67	93,33	**96,33**	85,67	**92,00**	79,00	**82,33**	68,33	**72,33**
cycling	100	100	100	100	**100**	99,00	**100**	81,33	**100**	70,67	**100**	63,33
Nordic walking	31,00	31,00	20,00	**32,00**	20,00	**27,33**	18,00	**27,00**	19,00	**22,67**	16,33	**22,00**
ascending stairs	3,00	3,00	**4,00**	3,67	4,00	4,00	6,30	**11,67**	11,00	**12,00**	**18,00**	8,67
descending stairs	7,00	7,00	**21,67**	16,33	**28,67**	21,33	29,67	**38,33**	34,00	**50,67**	33,67	**51,33**
vacuum cleaning	100	100	100	100	**100**	99,67	**100**	87,00	**100**	79,00	**91,67**	70,67
ironing	99,67	99,67	99,67	99,67	93,00	**100**	86,00	**97,67**	79,67	**90,33**	78,67	**83,00**
rope jumping	25,33	25,33	21,67	**28,67**	21,33	**35,00**	21,33	**34,00**	21,00	**36,33**	21,67	**28,33**

Table 4 Total accuracy comparison for the *All activity recognition* task

Sparse coefficients	Accuracy					
	Training dataset 50%		Training dataset 25%		Training dataset 10%	
1	88,75	88,75	84,44	84,44	65,75	65,75
5	**90,58**	88,67	**91,19**	84,58	66,56	**67,14**
10	**92,56**	87,22	**86,47**	83,92	66,44	**68,17**
15	**91,03**	81,83	**79,81**	78,72	64,66	**67,08**
20	**88,25**	77,39	**74,36**	72,03	**64,11**	63,44
25	**85,56**	72,69	**70,25**	68,44	**62,42**	58,06

Table 5 Intensity estimation task results for 50% training dataset

Activities	Intensity estimation (training dataset 50%)											
	1		5		10		15		20		25	
light	100	100	100	100	99,25	**100**	96,83	**99,42**	96,17	**97,42**	**95,67**	95,17
moderate	79,87	79,87	**84,67**	79,87	**86,67**	77,73	**85,13**	69,27	**81,53**	63,13	**78,00**	56,33
vigorous	88,56	88,56	87,89	**88,22**	**93,44**	86,00	**93,11**	79,33	**88,89**	74,44	**84,67**	70,00

Table 6 Intensity estimation task results for 25% training dataset

Activities	Intensity estimation (training dataset 25%)											
	1		5		10		15		20		25	
light	99,42	99,42	98,08	**99,42**	94,25	**99,42**	86,17	**99,08**	78,17	**97,42**	75,17	**95,08**
moderate	78,80	78,80	**85,87**	79,00	75,93	**76,27**	**65,87**	65,13	**58,47**	52,80	**51,80**	48,80
vigorous	73,89	73,89	**90,89**	74,11	**93,67**	76,00	**94,56**	74,22	**95,78**	70,22	**94,44**	65,67

Table 7 Intensity estimation task results for 10% training dataset

Activities	Intensity estimation (training dataset 10%)											
	1		5		10		15		20		25	
light	94,92	94,92	**98,33**	95,08	**98,25**	96,33	96,42	**96,50**	**93,58**	91,92	**93,00**	88,17
moderate	58,27	58,27	56,93	**60,07**	57,13	**59,47**	55,40	**56,27**	**56,80**	52,60	**53,80**	46,93
vigorous	39,33	39,33	40,22	**41,67**	39,56	**45,11**	37,77	**45,89**	37,00	**43,56**	36,00	**36,44**

5 Conclusions and Future Research

While the standard OMP based classifier brings many advantages on the field of activity recognition, it can be seen that there is still plenty of space for improvements. One such improvement has been presented in this paper and proven to generally increase the accuracy in recognition problems of the PAMAP2 dataset. Future research will be conducted to further improve the accuracy, researching the influence of the number of sparse coefficients on the accuracy, restrictions on the training dataset, optimization for lower computation times etc. We certainly feel sparse approximation based classification to be worth exploring into greater depths, strongly motivated by the success of the method in differentiating standing from sitting as well as its overall performance.

References

1. Blumensath, T., Davies, M.E.: On the difference between orthogonal matching pursuit and orthogonal least squares (2007)
2. Lavanya, D., Rani, D.K.: Performance evaluation of decision tree classifiers on medical datasets. International Journal of Computer Applications 26(4), 1–4 (2011)
3. Ermes, M., Parkka, J., Mantyjarvi, J., Korhonen, I.: Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. Trans. Info. Tech. Biomed. 12(1), 20–26 (2008) doi:10.1109/TITB.2007.899496, http://dx.doi.org/10.1109/TITB.2007.899496
4. Jang, J.S.R., Sun, C.T.: Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence. Prentice-Hall, Inc., Upper Saddle River (1997)
5. Jatoba, L.C., Grossmann, U., Kunze, C., Ottenbacher, J., Stork, W.: Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity. In: 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS 2008, pp. 5250–5253 (2008), doi:10.1109/IEMBS.2008.4650398
6. Kelly, D., Caulfield, B.: An investigation into non-invasive physical activity recognition using smartphones. In: 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 3340–3343 (2012), doi:10.1109/EMBC.2012.6346680
7. Khan, A., Lee, Y.K., Lee, S., Kim, T.S.: A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer. IEEE Transactions on Information Technology in Biomedicine 14(5), 1166–1172 (2010), doi:10.1109/TITB.2010.2051955
8. Lin, S.W., Chen, S.C., Chen, S.C.: Parameter determination and feature selection for c4.5 algorithm using scatter search approach, pp. 63–75 (2012)
9. Loh, W.Y.: Classification and regression trees, pp. 14–23 (2011)
10. Naranjo-Hernandez, D., Roa, L., Reina-Tosina, J., Estudillo-Valderrama, M.: Som: A smart sensor for human activity monitoring and assisted healthy ageing. IEEE Transactions on Biomedical Engineering 59(11), 3177–3184 (2012), doi:10.1109/TBME.2012.2206384
11. Pärkkä, J., Cluitmans, L., Ermes, M.: Personalization algorithm for real-time activity recognition using pda, wireless motion bands, and binary decision tree. Trans. Info. Tech. Biomed. 14(5), 1211–1215 (2010) doi:10.1109/TITB.2010.2055060, http://dx.doi.org/10.1109/TITB.2010.2055060
12. Reiss, A., Stricker, D.: Creating and benchmarking a new dataset for physical activity monitoring. In: The 5th Workshop on Affect and Behaviour Related Assistance (ABRA) (2012)
13. Reiss, A., Stricker, D.: Introducing a new benchmarked dataset for activity monitoring. In: The 16th IEEE International Symposium on Wearable Computers, ISWC (2012)

14. Tapia, E., Intille, S., Haskell, W., Larson, K., Wright, J., King, A., Friedman, R.: Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. In: 2007 11th IEEE International Symposium on Wearable Computers, pp. 37–40 (2007), doi:10.1109/ISWC.2007.4373774
15. Wei, B., Yang, M., Rana, R.K., Chou, C.T., Hu, W.: Distributed sparse approximation for frog sound classification. In: Proceedings of the 11th international conference on Information Processing in Sensor Networks, IPSN 2012, pp. 105–106. ACM, New York (2012) doi:10.1145/2185677.2185699, http://doi.acm.org/10.1145/2185677.2185699,
16. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 210–227 (2009), http://doi.ieeecomputersociety.org/10.1109/TPAMI.2008.79

Global Motion Estimation Using a New Motion Vector Outlier Rejection Algorithm

Burak Yıldırım and Hakkı Alparslan Ilgın

Abstract. Global Motion Estimation (GME) is mainly performed in either pixel or compressed domain. Compressed domain approaches usually utilize existing block matching motion data. On the other hand, in compressed domain based GME, there are many unwanted existing outliers because of noise and foreground objects which are obstacle for GME. In this paper, a new motion vector dissimilarity measure is proposed to remove motion vector (MV)-outliers to provide fast and accurate GME. In experimental results, it is shown that proposed method is fairly successive in terms of both accuracy and complexity compared to the state of the art methods.

Keywords: Dissimilarity Measure, Motion Vectors, Global Motion Estimation, Outlier Rejection.

1 Introduction

Video sequences generally comprise two types of motion: the motion of objects visible in the scene and the global motion that is caused by the moving camera. Object motion is usually difficult to describe, since in general, objects can show articulated or deformable motion. In general, the motion of objects can be rather complex and very difficult to model. Examples are water waves, explosions, traffic lights, or other sudden changes. On the other hand, camera motion is restricted to only a few degrees of freedom like the camera rotation angles or its focal length (zoom) [5].

Burak Yıldırım
Undersecretariat for Defense Industries İnönü Bulvarı Kirazlıdere Mevkii Süleyman Emin Caddesi 6/7 06100 Ankara, Turkey
e-mail: byildirim@ssm.gov.tr

Hakkı Alparslan Ilgın
Ankara University, Electrical and Electronics Eng. Dept. 06100 Tandoğan, Ankara, Turkey
e-mail: ilgin@eng.ankara.edu.tr

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 507 – 515.
DOI: 10.1007/978-3-319-00542-3_50

Global Motion Estimation (GME) is a common process of the modeling of the camera motion with two dimensional (2D) transform parameters and estimation of these parameters in video sequences [1,2]. GME has been widely used in many applications such as image registration, background modeling, moving object segmentation, scene analysis, and MPEG-7 video descriptors.

Approaches to global motion estimation have been naturally classified into one of two main classes: feature-based methods and or featureless methods [8],[11]. Feature-based methods use feature selection and tracking, but they have high computational complexity which is unsuitable for video communication applications [1],[10].

Featureless methods can be categorized into direct and indirect methods. Direct GME methods are pixel-based and try to minimize the prediction errors in the pixel domain. Indirect GME methods contain two stages, and GME is performed at the second stage based on the motion vectors resulted from the first motion estimation stage [1]. This procedure is what we adopt in our proposed approach. The first motion estimation stage uses a block matching algorithm (BMA) which is way of locating matching blocks between frames to obtain motion vectors. The second estimation stage uses GME algorithms to extract global motion parameters using these motion vectors. The motion vectors to be used for GME are superimposed with their original frame is shown as green arrows in Fig. 1 [10].

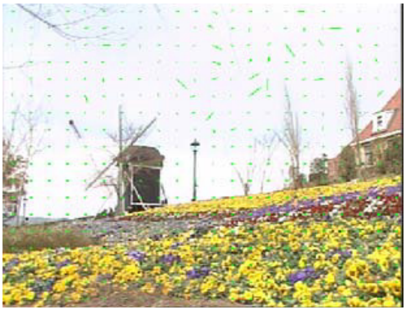


Fig. 1 The motion vectors with their original frame [10]

The outlier MVs representing noise and foreground impair the success of GME. For this reason, these MVs should be removed before GME. In the literature, for both outlier rejection and GME, different methods are used in general. In [1], to estimate the global motion from motion vectors, Newton-Raphson gradient descent method (GD) is used with an iterative approach with outlier rejection. In [6], tensor voting is used for outlier removal whereas it uses GD for GME.

MV based outlier rejection methods have been generally accomplished with spatial magnitude and phase correlation of MVs. Nguyen and Lee have proposed tensor voting for outlier removal before GME [6]. According to their approach (TV_GD), MVs are first encoded by second order tensors. A 2D voting process is then used to smooth MV field. The input and smoothed MV field is compared by a similarity, which uses a phase difference to remove outliers. In [7] (CAS_GD), a cascade composed of three-filters for outlier rejection has been used. The input MV is compared with its different oriented neighbors by both magnitude and phase difference to detect outliers in each filter. In [9] (LSS_ME), least square solution with M-estimator is used in order to reduce influence of outlier MVs and estimate global motion.

In this paper, we propose a new dissimilarity measure to remove MV outliers before GME. This paper is organized as follows: In Section 2, we introduce the overview of the global motion estimation with different motion models. Section 3 describes our new MV outlier rejection approach prior to GME. Section 4 gives the experimental results of the proposed approach comparing with other state of the art methods. Finally, conclusions are summarized in Section 5.

2 Overview of the Proposed Approach

Our proposed approach is mainly divided into two algorithms as MV-based outlier rejection and performing GME respectively. In this process, we first use input motion vectors. These motion vectors are synthesized or obtained from a motion estimation process which will further be explained in our study. The all MVs obtained in a frame generate the input MV field. The input MV field is then utilized for outlier rejection. The inliers retained from the outlier rejection process are used for GME. The overall process is summarized as a block diagram in Fig. 2.

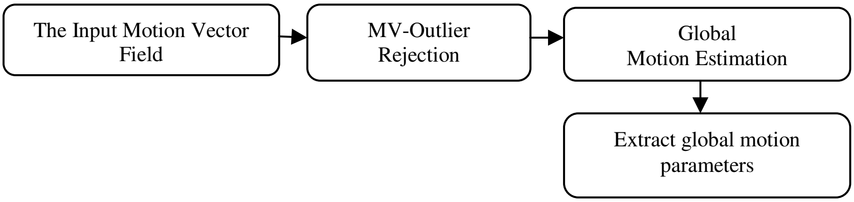

The Input Motion Vector Field
MV-Outlier Rejection
Global Motion Estimation
Extract global motion parameters

Fig. 2 Overall process of Outlier Rejection and Global Motion Estimation

The global motion estimation we have used in our approach [1] is constitutively performed with an iterative process: The input motion vector field (inliers) is used to estimate global motion parameters. First of all, these parameters are first initialized by assigned values using the input MVs. Iterative process is then implemented to estimate the parameters through minimizing the estimation errors between the input and the estimated motion vector field. If the specified condition about the change of the parameters is satisfied, the iteration stops and the estimated parameters are regarded as the global motion parameters.

2.1 Modeling of Camera Motion Using Motion Vector Field

Camera motion is modeled with common global motion models in [1] as translational, geometric, affine and perspective (projective). The perspective model has eight parameters being the most general one among other models as described in [6]. The motion field of a rotational camera[1] can be modeled with the perspective transform. Often, this model is written as the coordinate transformation [5]. In our approach, we have used the perspective model similarly with [6,7]. For the perspective model, the model parameters can be defined as $a = [a_0, a_1, ... a_7]$, and we need to estimate these parameters to obtain global motion. If we denote (x, y) and (x', y') as the coordinates of the current and the reference frame, respectively, perspective model transformation then can be defined as follows:

$$x' = f_x(x, y \mid a) = \left(\frac{a_0 x + a_1 y + a_2}{a_6 x + a_7 y + 1} \right)$$

$$y' = f_y(x, y \mid a) = \left(\frac{a_3 x + a_4 y + a_5}{a_6 x + a_7 y + 1} \right) \tag{1}$$

In MV based GME, the input frame is divided into blocks. Each block is represented with a motion vector. All MVs in the frame generate the input motion vector field. Let us denote *i* as the block index, (x_i, y_i) as the center of the i^{th} block, and (MVx_i, MVy_i) as the motion vector of the i^{th} block, the displacement vector for a motion model can be defined as:

$$\begin{aligned}(\Delta x_i, \Delta y_i) &= (x_i' - x_i, y_i' - y_i) \\ &= (f_x(x, y \mid a) - x_i, f_y(x, y \mid a) - y_i)\end{aligned} \tag{2}$$

We need to minimize the estimation error between the input and the estimated MV field. Total squared error *E* for *N* motion vectors is calculated as:

$$E = \sum_{i=1}^{N} ((MVx_i - (x_i' - x_i))^2 + (MVy_i - (y_i' - y_i))^2) \tag{3}$$

[1] The camera is allowed to rotate around its optical center in any direction, and it may also change the focal length (zoom). However, translational camera motion is not allowed because in such a case, the parallax effect would make it impossible to synthesize a static background image. [5]

Before GME, the outlier MVs, which are opposing to the global motion, should be eliminated. The outlier MVs can be one of the followings as defined in [1]:

- local MVs in local motion regions, where motion vectors do not represent global motion;
- zero or near-zero MVs in homogeneous regions, where motion vectors generated by minimizing prediction error usually fails to capture any motion;
- MVs with large matching errors due to the property of the block-matching process.

These outlier MVs do not fit the real camera motion and, the better they are detected and discarded, the better GME is ensured with a little iteration which means a faster convergence for the model parameters.

3 Motion Vector Dissimilarity Measure for MV-Based Outlier Rejection

Motion vector dissimilarity measure is very useful for detecting outlier MVs in our proposed approach. This measure is commonly used as a “Similarity Measure” to determine motion similarity between two frames [3,4]. In this “Similarity Measure”, each horizontal and vertical component of the input MV is half wave rectified into nonnegative channels. By this way, the angle between two MVs has been mapped to [0°,90°]. Motion similarity is then calculated and summed between two frames using these MVs. In [12], coarse segmentation of moving objects from H.264 compressed video by the difference between MV and global motion vector (GMV) field is performed by exploiting this measure with a little improvement. In [12], similarity value between two motion vectors are calculated and they are regarded as similar if the calculated measure is not less than the predefined threshold, otherwise, no similar.

This similarity measure also takes both advantage of magnitude and direction measure of the corresponding motion vectors. In our approach, we adapt this measure to “Dissimilarity Measure” in order to detect MV outliers in the motion vector field by benefiting the spatial correlation of the input MVs. We derive this measure by both phase and magnitude dissimilarity as shown in equation (4) with no rectification of MVs with a significant change. The input MV $MV(i, j)$, which is located at the center (i, j) of a block, is compared to its surrounding MVs defined as $MV(k,l)$ in our measure. Our dissimilarity measure can be formulated as follows:

$$d(i, j \mid n) = \begin{cases} (5) \ , \ if & \left\| MV(i, j) \right\| > 0 \ and \ \left\| MV(k,l) \right\| > 0 \\ 0 & ,else \end{cases} \tag{4}$$

$$d(i,j\,|\,n)=\frac{\min\left(\left\|MV(i,j)\right\|^{2},\left\|MV(k,l)\right\|^{2}\right)}{\max\left(\left\|MV(i,j)\right\|^{2},\left\|MV(k,l)\right\|^{2}\right)}\cdot(\varphi)_{+}\qquad (u)_{+}=|u| \tag{5}$$

$$=\frac{\min\left(\left\|MV(i,j)\right\|^{2},\left\|MV(k,l)\right\|^{2}\right).\cos^{-1}\left(\frac{\left\langle MV(i,j),MV(k,l)\right\rangle}{\left\|MV(i,j)\right\|.\left\|MV(k,l)\right\|}\right)_{+}}{\max\left(\left\|MV(i,j)\right\|^{2},\left\|MV(k,l)\right\|^{2}\right)} \tag{6}$$

$$dS(i,j)=\sum_{n}d(i,j\,|\,n\leq N) \tag{7}$$

In our formula (5), The input MV is compared with randomly selected N surrounding MVs. (In our approach, N is 8) Each measure is represented as $d(i,j\,|\,n,n=1,....,N)$. The first term at the left side of equation (5) is used to measure magnitude dissimilarity between two MVs. In equation (5), the minimum square norm value for two MVs is divided by the maximum one. This magnitude dissimilarity equals the maximum value (≈1) when the MVs are same or close to each other which can increase to detect MVs representing foreground in real sequences. In the second term at the right side of equation (5), we calculate the phase difference directly and get over the negative phase effect between two MVs with $(\varphi)_{+}$. The phase calculation is rewritten to avoid the computational cost by the dot product in equation (6). After the calculation of the dissimilarity measure for each input MV with its N neighbors, we then sum these measures. We get total dissimilarity measure value $dS(i, j)$ for each input MV from (7). Finally, we sort all dS values in descending order and 30 % of the motion vectors that have maximum dS values are identified as outliers. The most important benefit in our approach is that it works with soft decision as other approaches in [6,7]. In our approach, 30% of the MVs in motion vector field are regarded as outliers for a fair comparison to works in [6,7].

4 Experimental Results

We use the GD algorithm for GME due to its high performance in the proposed approach. Our method (MS_GD) is tested using both synthetic motion vector fields and real sequences. The parameters of four synthetic motion vector fields are used as shown in Table 1 [1]. We generate synthetic motion vector fields using these parameters assuming that images have 16x16 blocks with CIF resolution. To test GME performance, we corrupt motion vector fields with an independent zero mean Gaussiannoise in horizontal and vertical directions with $\sigma = \{0.7, 1.5, 2.2, 3.0\}$. We then remove outliers and perform GME from corrupted motion vector fields. The performance criterion is Signal to Noise ratio (SNR) as described in [1] between MV field generated by parameters a (Table 1) and the MV field generated by the estimated parameters $\hat{a}$. The SNR calculation is formulated in equation (8).

Table 1 Test Global Motion Parameters

Model	Motion Parameters
M1	a = [0.95, 0, 10.4238, 0, 0.95, 5.7927, 0, 0]
M2	a=[0.9964, -0.0249, 1.0981, 0.0856, 0.9457, -7.2, 0, 0]
M3	a=[0.9964, -0.0249, 6.0981, 0.0249, 0.9964, 2.5109, -2.7e-5, 1.9e-5];
M4	a = [1, 0, 4.4154, 0, 1, 0, −1.13e − 4, 0]

$$SNR(a,\hat{a}) = 10 * \log_{10}\left(\frac{\sum_{i=1}^{N}((MVx_i(a))^2 + (MVy_i(a))^2)}{\sum_{i=1}^{N}((MVx_i(a) - MVx_i(\hat{a}))^2 + (MVy_i(a) - MVy_i(\hat{a}))^2)} \right) \tag{8}$$

In our experiment, we compare MS_GD with four powerful approaches: Tensor Voting (TV_GD) in [6], Outlier Rejection Cascade (CAS_GD) in [7], iterative Gradient Descent (GD) in [1] and least square solution using M-estimator (LSS_ME) in [9].

Table 2 shows the converged average SNR values of 50 experiments in MS_GD and other approaches. For a fair comparison with [6,7] before GME, we regard 30% of all MVs as outliers. In Table 2, our method (MS_GD) achieves high SNR values despite noise increments. In SNR comparisons, we see that MS_GD performance is close to the best of the state of the art methods TV_GD and CAS_GD. Moreover, SNR decrease in MS_GD is less than CAS_GD and GD when noise variance increases. We also observe that MS_GD converges faster

Table 2 SNR(dB) Comparison for MV Fields Corrupted by Gaussian Noise

Model	GME Algorithm	Standard Deviation of Gaussian Noise			
		$\sigma = 0.7$	$\sigma = 1.5$	$\sigma = 2.2$	$\sigma = 3.0$
M1	**MS_GD**	**34.78**	**28.47**	**25.49**	**21.83**
	TV_GD	34.49	28.35	25.10	22.66
	CAS_GD	34.42	27.91	24.02	21.07
	GD	31.49	25.15	21.92	19.45
	LSS_ME	33.98	27.45	24.08	20.18
M2	**MS_GD**	**37.60**	**31.17**	**28.28**	**25.75**
	TV_GD	38.01	32.66	28.58	25.71
	CAS_GD	37.30	30.99	28.08	25.06
	GD	35.38	29.11	26.34	23.73
	LSS_ME	37.63	30.98	27.50	24.49
M3	**MS_GD**	**34.42**	**26.97**	**23.64**	**21.09**
	TV_GD	34.97	28.98	25.94	22.20
	CAS_GD	34.32	26.66	23.02	20.66
	GD	33.33	26.00	23.20	20.72
	LSS_ME	35.64	28.42	24.82	21.50
M4	**MS_GD**	**36.64**	**29.98**	**26.23**	**23.23**
	TV_GD	37.73	31.23	28.44	25.84
	CAS_GD	37.02	29.97	25.98	22.94
	GD	35.70	29.87	25.77	23.17
	LSS_ME	37.92	31.69	27.81	24.53

than GD and LSS_ME. It means that the converged parameters are satisfied in just 2 iterations. To achieve SNR values in Table 2, number of iterations in our method is set to 2 which is very fast. In TV_GD, voting range is taken as 50. In the experiments in Table 3, while number of iterations for MS_GD, CAS_GD, TV_GD are one, it is three, and six for LSS-ME and GD, respectively.

Our experiment is secondly evaluated on the real sequences that mostly contain camera motion. In this experiment, we first extract motion vectors from the real sequences to perform GME. Exhaustive search with 8x8 blocks is used to obtain MVs and generate MV field from the sequences. We then remove outlier MVs with our approach and perform GME. After the estimation of global motion parameters, we generate the global motion field using these parameters. We then perform Global Motion Compensation by warping current frame onto the reference frame according to the model using bilinear interpolation as described in [7]. We perform global motion compensation with these estimated MVs and measure the conventional PSNR between the compensated frame and the original (reference) frame. Peak Signal to Noise Ratio (PSNR) is measured on a logarithmic scale and depends on the mean squared error (MSE) ofbetween an original and an impaired image or video frame, relative to $(2^n - 1)^2$ (the square of the highest-possible signal value in the image, where n is the number of bits per image sample) [13]. The PSNR calculation is written as follows:

$$PSNR_{dB} = 10 * \log_{10} \frac{(2^n - 1)^2}{MSE} \tag{9}$$

In Table 3, MS-GD achieves higher PSNR performance than GD. Our PSNR performance is a bit less (about 0.15 dB) than TV_GD and CAS_GD. In Table 3, we compare PSNR values of our method with others for six test sequences. In this experiment, we calculate averaged PSNR over number of frames for each sequence. We then obtain final averaged PSNR value for each method at the bottom of the Table 3. As a result, our method shows very powerful performance between the accuracy and the complexity.

Table 3 PSNR (dB) Performance Comparisons

Video Sequence	**MS_GD**	TV_GD	CAS_GD	LSS_ME	GD
Flower Garden	**23.73**	24.45	24.03	24.35	24.86
Stefan	**24.16**	24.30	24.60	24.60	24.52
City	**29.61**	29.46	29.48	29.90	28.70
Tempete	**27.49**	27.91	27.83	27.66	26.51
Waterfall	**34.94**	34.85	34.86	34.69	34.71
Coastguard	**26.88**	26.86	26.78	26.83	26.54
Average	**27.80**	**27.97**	**27.93**	**28.01**	**27.64**

5 Conclusion

We proposed a new motion vector dissimilarity measure to remove outlier motion vectors before Global Motion Estimation. Our study is mainly performed with outlier rejection and GME respectively. In dissimilarity measure, we benefit from similarity measure of motion vectors in other studies and derive this measure in a different way. In GME section, we used the Newton-Raphson gradient descent method which is very powerful approach for global motion estimation. Our method was tested on both synthetic motion vector fields and real sequences. In synthetic motion vector fields, we obtain very good SNR values. Our method is so powerful that, to achieve the SNR and PSNR performances, we have just used one or two iterations which prove our algorithm is very fast. The results indicate that high performance is obtained compared to the state of the art methods for both accuracy and complexity.

References

[1] Su, Y., Sun, M.-T., Hsu, V.: Global motion estimation from coarsely sampled motion vector field and the applications. IEEE Transactions on Circuits and Systems for Video Technology 15(2), 232–242 (2005)

[2] Stiller, C., Konrad, J.: Estimating motion in image sequences, a tutorial on modeling and computation of 2D motion. IEEE Signal Processing Magazine 16(7), 70–91 (1999)

[3] Yeo, C., Ahammad, P., Ramchandran, K., Sastry, S.: High-speed action recognition and localization in compressed domain videos. IEEE Transactions on Circuits and Systems for Video Technology 18(8), 1006–1015 (2008)

[4] Efros, A., Berg, A., Mori, G., Malik, J.: Recognizing action at a distance. In: Procurement ICCV 2003, Nice, France, pp. 726–733 (2003)

[5] Farin, D.: Automatic Video Segmentation Employing Object/Camera Modeling Techniques. Ph.D. Thesis, Technische Univ. Eindhoven, Eindhoven, Netherlands (2005)

[6] Nguyen, T.D., Lee, G.: Tensor voting based outlier removal for global motion estimation. International Journal of Innovative Computing, Information and Control 9(1), 179–190 (2013)

[7] Chen, Y.-M., Bajic, I.V.: Motion vector outlier rejection cascade for global motion estimation. IEEE Signal Processing Letters 17(2), 197–200 (2010)

[8] Farin, D., de With, P.H.N.: Evaluation of a feature-based global motion estimation system. In: SPIE Visual Communications and Image Processing, pp. 1331–1342 (2005)

[9] Smolic, A., Hoeynck, M., Ohm, J.R.: Low-complexity global motion estimation from P-frame motion vectors for MPEG-7 applications. In: Procurement ICIP 2000, pp. 271–274 (2000)

[10] Soldatov, S., Strelnikov, K., Vatolin: Low Complexity Global Motion Estimation from Block Motion Vectors (2013), http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.63.5502

[11] Guerreiro, R.F.C., Aguiar, P.M.Q.: Global Motion Estimation: Feature-Based, Featureless, or Both?! In: Campilho, A., Kamel, M.S. (eds.) ICIAR 2006. LNCS, vol. 4141, pp. 721–730. Springer, Heidelberg (2006)

[12] Niu, C., Liu, Y.: Moving Object Segmentation Based on Video Coding Information in H.264 Compressed Domain. In: Procurement CSIP 2009, pp. 1–5 (2009)

[13] Richardson Iain, E.G.: H.264 and MPEG-4 Video Compression-Video Coding for Next-generation Multimedia, p. 23. John Wiley & Sons Ltd., The Atrium (2003) ISBN 0-470-84837-5

Searching for Dependences within the System of Measuring Stations by Using Symbolic Regression*

Petr Gajdoš, Michal Radecký, and Miroslav Vozňák

Abstract. This article deals with searching for dependences within the System of Measuring Stations. Weather measuring stations represent one of the most important data sources. The same could be said about stations that measure the composition of air and the level of pollutants. Knowledge of the current state of air quality resulting from the measured values is essential for citizens, especially in areas affected by heavy industry or dense traffic. Computation of such air quality indicators depends on values obtained from measuring stations which are more or less reliable. They can have failures or they can measure just a part of the required values. In general, searching for dependences represents a complex and non-linear problem that can be effectively solved by some class of evolutionary algorithms. This article describes a method that helps us to predict the levels of air quality in the case of station failure or data loss. The model is constructed by the symbolic regression with usage of the principles of genetic algorithms. The level of air quality of a given station is predicted with respect to a set of surrounding stations. All experiments were focused on real data obtained from the system of stations located in the Czech republic.

Keywords: Air pollution, Symbolic Regression, Quality Index, Knowledge extraction.

Petr Gajdoš · Michal Radecký
Department of Computer Science, FEECS,VŠB – Technical University of Ostrava, 17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic
e-mail: {petr.gajdos,michal.radecky}@vsb.cz

Miroslav Vozňák
Department of Telecommunicatios, FEECS, VŠB – Technical University of Ostrava, 17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic
e-mail: miroslav.voznak@vsb.cz

* This work was supported by the Bio-Inspired Methods: research, development and knowledge transfer project, reg. no. CZ.1.07/2.3.00/20.0073 funded by Operational Programme Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic and IT4Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 supported by Operational Programme 'Research and Development for Innovations' funded by Structural Funds of the European Union and state budget of the Czech Republic.

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 517–528.
DOI: 10.1007/978-3-319-00542-3_51

1 Air Pollution as a Global Problem

Every day, the average person inhales about 20,000 liters of air. Every breath may constitute a risk of inhaling dangerous chemicals that have found their way into the air. These dangerous substances can be either in the form of gases or particles. Because it is located in the atmosphere, air pollution is able to travel easily.[7]

The sources of air pollution are natural and human-based, local and global, and it has been a problem throughout history. Unfortunately, today's technical and consumerist age has very considerable influence on air quality. As a result, air pollution is a global problem and has been the subject of global cooperation and conflict as well as the interest in the environment at the local level.

1.1 Air Pollution in the Czech Republic

The Czech Republic is a landlocked country in Central Europe. It is bordered by Poland to the north, Germany to the west, Austria to the south and Slovakia to the east. The significant part of the borders is formed by mountains, which define the geographical and climatical conditions affecting the state of air pollution. The highly industrial character of Czech economy, as well as the transportation and local heating methods, are also very important factors for the quality of air.

1.2 Problem Definition

It is necessary to have information on the air components to specify its quality and pollution. There are several ways to obtain this type of data. The local measurement is one of them. The network of local measurement stations is often the basis for collecting this type of data. Thanks to the data and extracted information, models of air pollution state and distribution, prediction of the following states, etc., can be calculated.[17][15] Unfortunately, the measurement stations are technical devices and can have some technical problems, failures or downtimes. In these cases, the stations are not a data source and they are temporarily removed from the network. This is a space for the application of sophisticated methods that allow the virtual replacement of such a station, based on the knowledge of a given station and its surroundings.

2 Air Pollution and Its Components

Air pollution is defined based on the content of certain gases and particles in the air [13]. Limits for its specific composition and concentration of all substances are given by the law and rules defined for different countries in different ways. In the case of the Czech Republic the following substances are observed.

SO_2 - sulfur dioxide High concentrations of sulfur dioxide (SO_2) affect breathing and may aggravate existing respiratory and cardiovascular disease. Sensitive populations include asthmatics, individuals with bronchitis or emphysema, children and the elderly. SO_2 is also a primary contributor to acid rain, which causes acidification of lakes and streams and can damage trees, crops, historic buildings and statues.

NO_2 - nitrogen dioxide Nitrogen dioxide (NO_2) is a brownish, highly reactive gas that is present in all urban atmospheres. NO_2 can irritate the lungs, cause bronchitis and pneumonia, and lower resistance to respiratory infections.

CO - carbon monoxide Carbon monoxide (CO) is a colorless, odorless and poisonous gas produced by the incomplete burning of carbon in fuels. When CO enters the bloodstream, it reduces the delivery of oxygen to the body's organs and tissues. Health threats are most serious for those who suffer from cardiovascular disease. 77% of the nationwide CO emissions are from transportation sources. Thus, the focus of CO monitoring has been at traffic-oriented sites in urban areas where the main source of CO is motor vehicle exhaust. Other major CO sources are wood-burning stoves, incinerators and industrial sources.

O_3 - ozone Ozone (O_3) is the major component of smog. Although O_3 in the upper atmosphere is beneficial because it shields the earth from the sun's harmful ultraviolet radiation, high concentrations of O_3 at ground level are a major health and environmental concern. The reactivity of O_3 causes health problems because it damages lung tissue, reduces lung function and sensitizes the lungs to other irritants. Scientific evidence indicates that ambient levels of O_3 not only affect people with impaired respiratory systems, such as asthmatics, but healthy adults and children as well.

PM - particulate matter Particulate matter (PM) is a mixture of particles that can adversely effect human health, damage materials and form atmospheric haze that degrades visibility. PM is usually divided up into different classes based on size, ranging total suspended matter (TSP) from PM-2.5 (particles less than 2.5 microns) to PM-10 (particles less than 10 microns in aerodynamic diameter). Particulate matter (PM) includes dust, dirt, soot, smoke and liquid droplets directly emitted into the air by sources such as factories, power plants, cars, construction activity, fires and natural windblown dust. In general, the smallest particles pose the highest human health risks. PM exposure can affect breathing, aggravate existing respiratory and cardiovascular disease, alter the body's defense systems against foreign materials, and damage lung tissue, contributing to cancer and premature death.

2.1 Air Quality Evaluation

It is relatively easy to determine the real concentration of each component in the atmosphere due to measuring methods and equipment. However, only the knowledge of these separated values is not sufficiently informative. Based on Czech specifications, all substances have a defined range that allows to discretize the values to the quality index. Each substance can be described by six quality indexes, where 1 means very good air quality and 6 means very poor air quality, see Table 1.[4]

Table 1 Rules for discretization of all relevant substances

Quality index	**SO_2**	**NO_2**	**CO**	**O_3**	**PM_{10}**
	$\mu g/m3(1h)$	$\mu g/m3(1h)$	$\mu g/m3(8h)$	$\mu g/m3(1h)$	$\mu g/m3(1h)$
1 (very good)	0 - 25	0 - 25	0 - 1000	0 - 33	0 - 20
2 (good)	26 - 50	26 - 50	1001 - 2000	34 - 65	21 - 40
3 (fair)	51 - 120	51 - 100	2001 - 4000	66 - 120	41 - 70
4 (suitable)	121 - 350	101 - 200	4001 - 10000	121 - 180	71 - 90
5 (poor)	351 - 500	201 - 400	10001 - 30000	181 - 240	91 - 180
6 (very poor)	>500	>400	>30000	>240	>180

The air pollution monitoring is formed on a network of measuring stations spread accross the Czech republic, see Fig 1. Nowadays, this network contains 110 stations of different classification (traffic, urban, suburban, rural, industrial). Each of these stations is able to measure the above mentioned substances (all or some of them) at its location. Measured values of substances, resp. its quality index, are aggregated into the overall station quality index. This overall number (the meaning is the same as for individual elements) is the main way of presentation of air quality towards the general public. The calculation of air quality index is based on the assessment of 1-hour concentration of sulphur dioxide (SO_2), nitrogen dioxide (NO_2), ozone (O_3), suspended particles (PM_{10}) and 8-hour running concentrations of carbon monoxide (CO). The calculation considers the possible impact of air pollution on the health of the population. Air quality index is determined for each quantity in the given locality separately - the highest of them is presented. Air quality index is presented only in the case if the 1-hour data are available for at least NO_2 and PM_{10}, if they are the subject of standard measurements in the locality. If there is not possibility to calculate the quality index of a given station, the station index is determined as "not measured" or "inclomplete data". The term "not measured" indicates that all necessary substances are not measured on the station during a given hour. The term "incomplete data" may indicate that the instruments at the station were under calibration during a given hour.

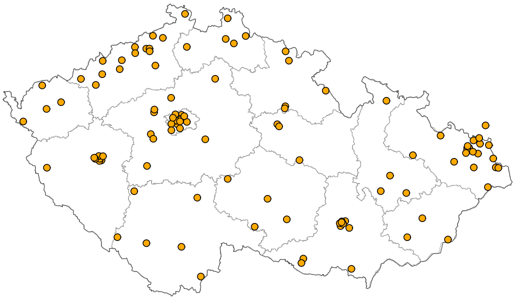


Fig. 1 The Czech Republic and station locations

Examples of the overall quality index calculation are presented below. The real data from 2012-04-09 13:22 CEST are used.

- The station Prague-Legerova (ALEGA) has measured the following values: SO_2 = not measured; NO_2 = 18.9 $\mu g/m3$; CO = 912 $\mu g/m3$; O_3 = not measured; PM_{10} = 14 $\mu g/m3$. Based on discretization rules, this station has defined by relevant indexes: NO_2 = 1; CO = 1; PM_{10} = 1. So, the overall station index is 1 – very good.
- The station Mikulov-Sedlec (BMISA) has measured the following values: SO_2 = incomplete data; NO_2 = 1.0 $\mu g/m3$; CO = not measured; O_3 = 88.6 $\mu g/m3$; PM_{10} = 6 $\mu g/m3$. Based on discretization rules, this station has defined by relevant indexes: NO_2 = 1; O_3 = 3; PM_{10} = 1. So, the overall station index is 3 – fair.
- The station Bily Kriz (TBKRA) has measured the following values: SO_2 = 7.5 $\mu g/m3$; NO_2 = 5.7 $\mu g/m3$; CO = not measured; O_3 = 83.6 $\mu g/m3$; PM_{10} = not measured. This station has defined no overall quality index bacause PM_{10} is not measured.

3 Symbolic Regression via Genetic Programming

Symbolic regression via genetic programming is a branch of empirical modeling that evolves summary expressions for available data. Although intrinsically difficult (the search space is infinite), recent algorithmic advances coupled with faster computers have enabled application of symbolic regression to a wide variety of industrial data sets. Unique benefits of symbolic regression include human insight and interpretability of model results, identification of key variables and variable combinations, and the generation of computationally simple models for deployment into operational models.

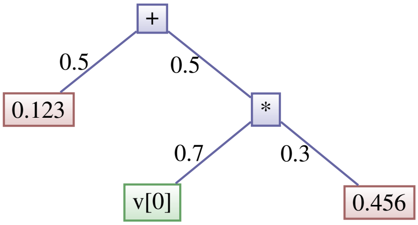
The challenging task of symbolic regression is to identify and express a real or simulated system or a process, based on a limited number of observations of the system's behavior. The system under study is characterized by some important parameters which need to be available for an observer, but are usually difficult to monitor, e.g. they need to be measured in a lab, simulated or observed in real time only, or at high time and computational expenses. Empirical modeling attempts to express these critical variables via other controllable variables that are easier to monitor, can be measured more accurately or timely, are cheaper to simulate, etc. Symbolic regression provides such expressions of crucial process characteristics, or, response variables, defined (symbolically) as mathematical functions of some of the easy-to-measure input variables, and calls these expressions empirical input-output models (or input-response models).

Industrial modeling problems, for which symbolic regression is used, have two main characteristics:

- No or little information is known about the underlying system producing the data, and therefore no assumptions on model structure can be made.

- The available data is high-dimensional, and often imbalanced, with either an abundant or insufficient number of samples.

A particular solution of symbolic regression in the form of mathematical formula can be illustrated by a tree structure as well, see Figure 2. Some nodes have different colors, which will be discussed in more detail later. The edge or edges can have weights to make the whole model more fuzzy.



Illustrative example:
Lets say we have a set of N input vectors of dimension k. Then $v[0]$ represents the first element of input vectors. The formula

$$r_i = (0.5*0.123)+(0.5*((0.7*v[0])*(0.3*0.456)))$$

returns a result r_i for every i-th input vector, where $i \in N$.

Fig. 2 Illustrative result of Symbolic Regression

The set of symbols (functions) depends on the application area and represents one of the advantages of symbolic regression. In statistics, attribute dependences can be simulated and tested with a set of symbols of logical functions $S = \{\wedge, \vee, \supset, \equiv, \neg\}$, in the area of signal processing with a set of goniometric functions $S = \{sin, cos\}$, etc.

To discover acceptable models with realistic time and computational effort, symbolic regression exploits a stochastic iterative search technique, based on the artificial evolution of model expressions. This method called genetic programming looks for appropriate expressions of the response variable in the space of all valid formulas containing a minimal set of input variables and a proposed set of basic operators and constants.

Evolutionary algorithms are stochastic search methods that mimic the metaphor of natural biological evolution, which applies the principles of evolution found in nature to the problem of finding an optimal solution to a solver problem. An evolutionary algorithm is a generic term used to indicate any population-based optimization algorithm that uses mechanisms inspired by biological evolution, such as reproduction, mutation and recombination. Candidate solutions to the optimization problem play the role of individuals in a population, and the cost function determines the environment within the solutions "live". Evolution of the population then takes place after the repeated application of the above operators. Genetic algorithm is the most popular type of evolutionary algorithms.

3.1 Genetic Algorithms (GA)

Genetic algorithms (GA) were described by John Holland in 1960s and further developed by Holland and his students and colleagues at the University of Michigan in the 1960s and 1970s. GA used Darwinian Evolution to extract nature optimization strategies that use them successfully and transform them for application in mathematical optimization theory to find the global optimum in a defined phase space [2][8][9]. We refer to [14][1][11][6][5][12][10][3][16] for more detailed information on the usage of GA.

3.2 Proposed Genetic Algorithm for Symbolic Regression

Typically, any genetic algorithm used for purpose of optimization consists of the following features:

1. Chromosome or individual representation.
2. Objective function “fitness function”.
3. Genetic operators (selection, crossover and mutation).

Applying GA on a population of individuals or chromosomes shows that several operators are utilized.

Chromosome Encoding. The chromosome in the form of a tree structure represents a suitable encoding for our purposes. Just for simplicity, we pass away the physical arrangement of such data structure in the computer memory. The following rules were applied in our solution:

- There are only three types of tree nodes: Constant, Variable, Function. See the red, green and blue nodes in Figure 2.
- The edges’ weights are set randomly.
- All leaf nodes are Constant or Variable nodes.

Objective Function. The goal of GA is to find a solution to a complex optimization problem, which is optimal or near-optimal. GA searches for better performing candidates, where the performance can be measured in terms of objective “fitness function”. Because of knowledge of the results of all input vectors in the training set, the fitness function minimizes the measured error between known and achieved results. All input vectors go through all trees (chromosomes) and finally a fitness value is computed for every chromosome.

Selection Operator. Selection determines which solution candidates are allowed to participate in crossover and undergo possible mutation. The chromosomes are sorted with respect to their fitness function results. A given percentage of worst chromosomes is deleted. This is a matter of application settings.

Crossover Operator. Promising candidates, as represented by relatively better performing solutions, are combined through a process of recombination referred to as crossover. This ensures that the search process is not random but rather that it is consciously directed into promising regions of the solution space. Crossover exchanges subparts of the selected chromosomes, where the position of the subparts is selected randomly to produce offspring.

Mutation Operator. New genetic material can be introduced into the population through mutation. This increases the diversity in the population. Mutation occurs by randomly selecting particular elements in a particular offspring. In the case of symbolic regression, following mutation are defined:

- Function/Symbol node (FN) can be substituted by FN only to prevent tree structure breaks. Both functions should have the same arity.
- Constant node (CN) and Variable node (VN) can mutate randomly
- All weights can be changed randomly.

4 Experiments and Results

4.1 Data Collection and Analysis

The collection of data represents the first step in creating a database suitable for further processing and knowledge mining. Thanks to the open access of the Czech Hydrometeorological Institute (www.chmi.cz), it is possible to download live generated data under a creative commons license. The data is collected from the stations across the Czech Republic and it is provided for download in a form of XML document format. This XML document is updated one or two times per hour and contains information on each station (name, code, location, classification, etc.) as well as all measured values of defined substances of air pollution. Finally, the data collection consists of approximately 4400 records per a day.

For the experiments, three pollution stations were selected; **ALEGA**, **BBMVA**, and **TOZRA**. They are located close to three important cities (Prague, Brno, and Ostrava) in the Czech Republic. Table 2 shows the program settings of selected

Table 2 Parameter settings for mentioned experiments

Description	ALEGA	BBMVA	TOZRA
AND, OR, NOT	yes	no	no
$<, >, min, max$	yes	yes	yes
$+, -, *, /$	no	yes	yes
How many times the experiment will be repeated	5	5	5
Data dimension = number of variables in the input vector	109	109	109
Probability of occurrence of Function Node	0.7	0.7	0.7
Probability of occurrence of Variable Node	0.2	0.2	0.2
Probability of occurrence of Constant Node	0.1	0.1	0.1
The number of trees in a single population	200	200	200
Max number of Function Nodes in a Tree during tree generation	10	10	10
The percentage of trees that will be deleted before new GP loop	0.5	0.5	0.5
The percentage of deleted trees that will be substituted by clones of best trees	0.5	0.5	0.5
Total number of epochs of a genetic algorithm	1000	1000	1000

performed experiments. Different symbol sets (operators) were used just in experiments as can be seen in a few first rows of Table 2. Many other settings were tested during experiments and the table shows such settings, where the best adjustments of simulated and real data have been achieved.

4.2 Results of Experiments

The main goal of the experiments was to achieve the nearest adjustment of real and simulated data. For the sake of simplicity, all graphs in figures show two splines just to illustrate the real (red) and simulated (green) values. In fact, the graphs is too dense and all values are integers from the interval $< 0; 6 >$.

The final precision of simulation was computed in following steps:

1. Resulting model consists of mathematical formula that expresses the data relationship between measured stations.
2. Such model can be evaluated in a time for every input vector. Resulting value is given by a float number that is transformed into an integer value from the interval $< 0; 6 >$ (quality index).
3. The model is evaluated as a percentage of correspondence between real and simulated quality indexes.

The first selected station (ALEGA - see the Figure 3) is a station in Prague, the capital of the Czech Republic. The precision of simulation was 93.3% in terms of the final evaluation of the quality index. It is worth noting that there is a coal mining industry to the north-west of Prague which respects the direction of dependent stations (ALEGA → AVELA).

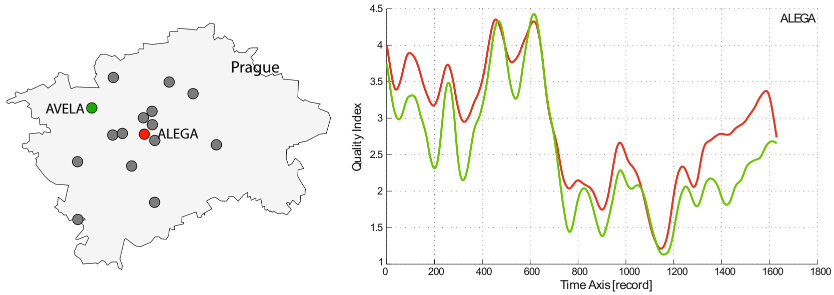


Fig. 3 Locations of dependent stations (ALEGA - AVELA), related dependency function and an illustrative graph of achieved results.The relationship between ALEGA and AVELA stations can be expressed by $ALEGA_i = 0.99806 * AVELA_i$, where i is the index of measurement in time.

The second selected station (BBMVA - see Figure 4) is a station near Brno, the biggest city of South Moravia. The precision of simulation was 91.6% in terms of the final evaluation of the quality index. The dependent station (BBMLA) is located close to the by-pass highway of Brno city.

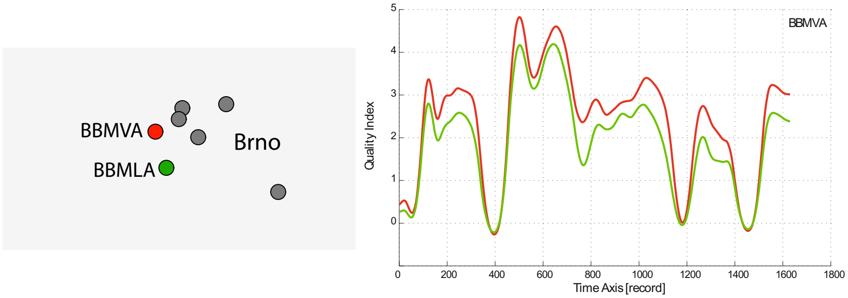


Fig. 4 Locations of dependent stations (BBMVA - BBMLA), related dependency function and an illustrative graph of achieved results. The relationship between BBMVA and BBMLA stations can be expressed by: $BBMVA_i = 0.82243 * BBMLA_i$, where i is the index of measurement in time.

The last selected station (TOZRA - see Figure 5) is a station near Ostrava, the industrial city in North-East Moravia. Currently, there are several air polluters and the most discussed question is the air pollution coming from Poland. The precision of simulation was 88.7% in terms of the final evaluation of the quality index. It is worth noting that the dependent stations are located near the state border with Poland.

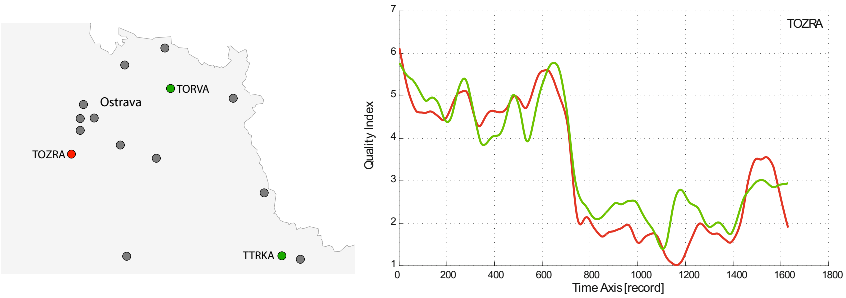


Fig. 5 Locations of dependent stations (TOZRA - TORVA, TTRKA), related dependency function and an illustrative graph of achieved results. The relationship between TOZRA, TORVA, and TTRKA stations can be expressed by: $TOZRA_i = (0.89021 * TORVA_i) + (0.17689 * TTRKA_i)$, where i is the index of measurement in time.

5 Conclusions and Future Work

This article describes a method of measurement of the quality index of air pollution. The proposed method, based on Symbolic Regression, can be used to simulate further progress of the quality index measured by the selected station. This can be very useful especially in such cases where the real station is not able to provide the required data because of hardware failure. Several experiments illustrate the precision of the proposed method. Moreover, some dependences between the stations were observed, which should make form grounds for further research. In future work, wind maps should be taken into account as well as the landscape surface. The experiments will be extended to cover all stations and observed dependencies will be studied in more detail.

References

1. Fang, H., Ross, P., Corne, D.: Genetic algorithms for timetabling and scheduling (1994), http://www.asap.cs.nott.ac.uk/ASAP/ttg/resources.html
2. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley (1989)
3. Huang, C., Li, G., Xu, Z., Yu, A., Chang, L.: Design of optimal digital lattice filter structures based on genetic algorithm. Signal Processing 92(4), 989–998 (2012)
4. hydrometeorological institute, C.: Web page with data specification and access (2012), http://www.chmi.cz/
5. Ishibuchi, H., Nakashima, Y., Nojima, Y.: Performance evaluation of evolutionary multiobjective optimization algorithms for multiobjective fuzzy genetics-based machine learning. Soft Comput. 15(12), 2415–2434 (2011)
6. Juzoji, H., Nakajima, I., Kitano, T.: A development of network topology of wireless packet communications for disaster situation with genetic algorithms or with dijkstra's. In: ICC, pp. 1–5 (2011)
7. Lave, L.B., Seskin, E.P.: Air pollution and human health. Science 169(3947), 723–733 (1970) doi:10.1126/science.169.3947.723, http://dx.doi.org/10.1126/science.169.3947.723,
8. Melanie, M.: An Introduction to Genetic Algorithms. A Bradford Book, MIT-Press (1999)
9. Melanie, M., Stephanie, F.: Genetic algorithms and artificial life. Santa Fe Institute, working Paper 93-11-072 (1994)
10. Pan, S.T.: A canonic-signed-digit coded genetic algorithm for designing finite impulse response digital filter. Digital Signal Processing 20(2), 314–327 (2010)
11. Park, B.J., Choi, H.R.: A genetic algorithm for integration of process planning and scheduling in a job shop. In: Australian Conference on Artificial Intelligence, pp. 647–657 (2006)
12. Sedighi, K.H., Manikas, T.W., Ashenayi, K., Wainwright, R.L.: A genetic algorithm for autonomous navigation using variable-monotone paths. I. J. Robotics and Automation 24(4) (2009)
13. Seinfeld, J.H., Pandis, S.N.: Atmospheric Chemistry and Physics: From Air Pollution to Climate Change, 2nd edn. Wiley-Interscience (2006), http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0471720186
14. Tsang, E.P.K., Warwick, T.: Applying genetic algorithms to constraints satisfaction optimization problems. In: Proc. of 9th European Conf. on AI, Aiello L.C (1990)

15. Vardoulakis, S., Fisher, B.E., Pericleous, K., Gonzalez-Flesca, N.: Modelling air quality in street canyons: a review. Atmospheric Environment 37(2), 155–182 (2003) doi:10.1016/S1352-2310(02)00857-9,
`http://www.sciencedirect.com/science/article/pii/S1352231002008579`
16. Wainwright, R.L.: Introduction to genetic algorithms theory and applications. In: The Seventh Oklahoma Symposium on Artificial Intelligence (1993)
17. Zannetti, P.: Air pollution modeling: theories, computational methods, and available software. Computational Mechanics Publications (1990),
`http://books.google.cz/books?id=FyxSAAAAMAAJ`

The Effect of Sub-sampling on Hyperspectral Dimension Reduction

Ali Ömer Kozal, Mustafa Teke, and Hakkı Alparslan Ilgın

Abstract. Hyperspectral images which are captured in narrow bands in continuous manner contain very large data. This data need high processing power to classify and may contain redundant information. A variety of dimension reduction methods are used to cope with this high dimensionality. In this paper, the effect of sub-sampling hyperspectral images for dimension reduction techniques is explored and compared in classification performance and calculation time.

Keywords: Hyperspectral imaging, dimension reduction, remote sensing, hyperspectral image classification.

1 Introducton

Hyperspectral images are usually acquired through the visible, NIR, SWIR parts of the electromagnetic spectrum with narrow and contiguous wavelength. A typical image consists of 50-250 bands. Since the bands are close to each other, there may be redundancy of data. Due to containing more bands than visible zone, hyperspectral images present much more data than human perception can sense. Therefore they can be used in many fields such as agriculture, mining, astronomy and chemistry. Main purpose of using hyperspectral images is to distinguish the objects by utilizing bands in similar spectra. However numerous bands cause complex computation. Besides, classifying performance decreases due to large number of bands. This is called Hughes effect. As a consequence, it is essential to perform a dimension reduction prior to classification.

Ali Ömer Kozal · Mustafa Teke
TÜBİTAK UZAY (The Scientific And Technological Research Council Of Turkey, Space Technologies Research Institute)
ODTÜ Yerleşkesi, 06531, Ankara, Turkey
e-mail: {omer.kozal,mustafa.teke}@tubitak.gov.tr

Ali Ömer Kozal · Hakkı Alparslan Ilgın
Ankara University, Electrical and Electronics Eng. Dept., Ankara Turkey
e-mail: ilgin@eng.ankara.edu.tr

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 529 – 537.
DOI: 10.1007/978-3-319-00542-3_52

Dimension reduction is defined as the process of representing a high-dimensional data in a lower-dimensional space. The purpose of the dimension reduction is to represent the information in high dimension data with fewer features and in lower dimension, but also to transform the space with higher classifying accuracy.

Dimension reduction techniques can be divided into different groups: Linear and non-linear methods, supervised and non-supervised methods. Well-known linear unsupervised methods are Principal Component Analysis (PCA) and Factor Analysis. On the other hand Linear Discriminant Analysis (LDA) is supervised linear method. Primary non-linear dimension reduction methods are Isomap and Auto Encoder [1].

In this study, the effect of sub-sampling on hyperspectral images with different kinds of dimension reduction techniques is researched. The calculation time and accuracy are considered to evaluate the effect of reduction in input data.

Indian Pine image, captured by Airborne Visible Infrared Imaging Spectrometer (AVIRIS) and its ground truth is used. The image has 145x145 pixels in each 224 bands and contains 16 different classes [2].

For dimension reduction, 12 different dimension reduction methods are utilized and compared for both entire data set and sub-sampled data set.

During classification phase, Support Vector Machine (SVM) is used. SVM is trained with subset of the out sampled set and then classification is performed with the extended set. The accuracy is calculated by comparing the result of classification with the ground truth.

2 Dimension Reduction Methods

Dimension reduction process consists of transforming the data set X with n samples and B dimension/feature $(n*B)$ to data set Y by using transformation matrix W as follows

$$Y = WX \tag{1}$$

The dimensions/features of data set Y is ranked in accordance with the information they contain.

The following dimension reduction methods are used in this study:

2.1 Principal Component Analysis (PCA)

PCA is an unsupervised dimension reduction method using variances to transform the data to uncorrelated spaces [3].

2.2 Linear Discriminant Analysis (LDA)

LDA is a supervised learning method that takes the values of scattering in the class (S_W) and between the classes (S_B) into account and calculates the transformation matrix given below:

$$W = \arg\max_{W} \left| \frac{W^T S_B W}{W^T S_W W} \right| \tag{2}$$

LDA tries to put forward the differences between the classes and projects the data according to that, while PCA does not consider the classes but deals with data values.

2.3 Isomap (ISO)

Isomap is one of the widely used unsupervised dimension reduction method based on preserving the distances among data. The algorithm implements a simple method for estimating intrinsic geometry of data manifold based on a rough estimation of each data point's neighbors on the manifold [4].

2.4 Factor Analysis (FA)

FA is a statistical method that represents the variability among the samples with less number of variables called factors. The samples are modeled as linear combinations of the potential factors.

2.5 Locality Preserving Projection (LPP)

LPP is an unsupervised linear dimension reduction method, which aims to find a linear mapping that minimizes the cost function of Laplacian Eigenmaps [5].

2.6 Neighborhood Preserving Embedding (NPE)

NPE is a linear approximation to the locally linear embedding algorithm [6], which can preserve local neighborhood structure on data manifold [7].

2.7 Neighborhood Components Analysis (NCA)

NCA is a supervised linear method based on learning a Mahalanobis distance measure to be used in the K-Nearest Neighbors (KNN) classification algorithm [8].

2.8 Auto Encoder (AE)

AE is a multi-layer neural network-based dimension reduction method. The method searches encoded data on a compressed projection [9].

2.9 Probabilistic PCA (PPCA)

PPCA is a version of PCA which determines the principal axes of a set of observed data vectors through maximum-likelihood estimation of parameters in a latent variable model [10][11].

2.10 Maximally Collapsing Metric Learning (MCML)

MCML is a supervised linear dimension reduction method using Mahalanobis distance as metric, which assumes points in the same class are simultaneously near each other and far from points in the other classes and tries to collapse classes to a single point [12].

2.11 Fast Maximum Variance Unfolding (FMVU)

Similar to Isomap, FMVU defines a neighborhood graph on the data and retains pairwise distances in the resulting graph. FMVU differs from Isomap in that explicitly attempts to unfold data manifolds [13]. FMVU is also known as Semidefinite Embedding [14].

2.12 Manifold Charting (MANC)

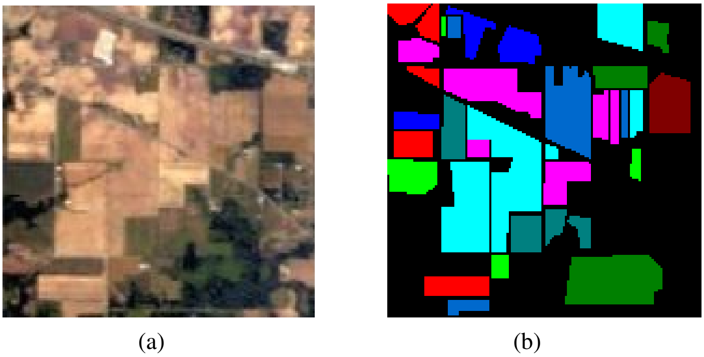
MANC constructs a low-dimensional data representation by aligning a mixture of factor analyzers model [15]. Manifold charting minimizes a convex cost function that measures the amount of disagreement between the linear models on the global coordinates of the data points. The minimization of this cost function can be performed by solving a generalized Eigen problem [14].

3 Experiments and Results

3.1 Data Set and Test Environment

For the experiments, Indian Pine image set is used that was captured using Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) on June 12, 1992 over the Indian Pines test site in Northwestern Indiana. The image consists of 145 × 145 pixels and 224 bands in the wavelength range of 0.4–2.5 µm. The ground truth contains 16 classes.

The RGB representation that is of the image and the ground truth is shown in Fig 1.



(a) (b)

Fig. 1 RGB representation (a) and Ground Truth (b) of Indian Pine Image for 9 classes

Although the image has 16 classes, due to limited representation of some classes, 9 of these with highest samples were chosen to train the SVM classifier effectively [16].

The classes and number of pixels of each class are given in Table 1.

Table 1 Number of pixels per class

Class No	Class Name	Number of Pixels
1	Corn no tillage	1434
2	Corn min. tillage	834
3	Grass/Pasture	497
4	Grass/Trees	747
5	Hay-windrowed	489
6	Soybean no tillage	968
7	Soybean min. tillage	2468
8	Soybean clean	614
9	Woods	1294

All tests are performed in MATLAB. Most of the techniques are part of the Dimension Reduction Toolbox [14]. The PC configuration is 2.3 Ghz Intel® Core™ i7-3770 3.40Ghz PCU with 16GB of RAM.

3.2 Dimension Selection

The lower dimensionality that will be achieved using the dimension reduction methods from the 200 bands is obtained by estimating the intrinsic dimensionality. The intrinsic dimension is the minimum number of features to represent the data [17][18]. For intrinsic dimensionality, Maximum Likelihood Estimation method is used [19].

3.3 Test Method

The test can be separated in two parts: Dimension reduction and classification.

3.3.1 Dimension Reduction

After obtaining the image (145x145x200) a filter is applied to eliminate the water absorption bands and the classes with inadequate samples. Then we have the 9345 pixels data set with 200 bands.

For dimension reduction, entire data set and reduced data set (sub-sampled set) approach is used.

In entire data set approach, filtered data is given to dimension reduction set without any manipulation as an input and the output is a data set with 9345 pixels and 9 dimensions (see Fig 2).

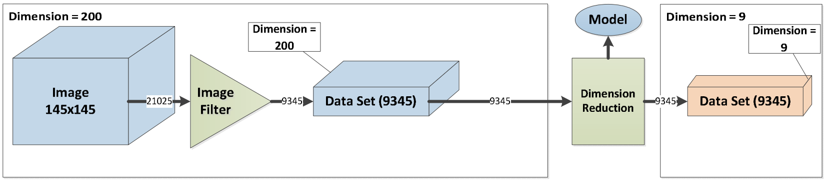


Fig. 2 Dimension reduction with entire data set

In sub-sample set approach; after filtering process, a random elements picker is used to get a subset of 100 samples from each class. The subset is the input for the dimension reduction algorithm and the output is a set with 900 elements and 9 dimensions as displayed in Fig 3.

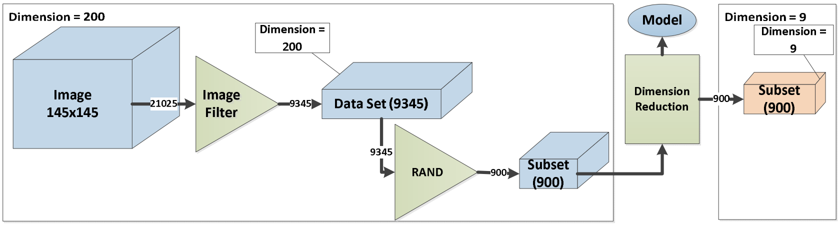


Fig. 3 Dimension reduction with sub-sampled data set

The dimension reduction method also outputs the *model* in each case that is created during the process for reusability.

After dimension reduction, the *model* is used to perform out of sample extension to entire image as shown in Fig 4.

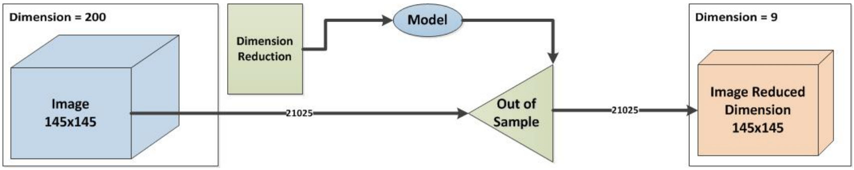


Fig. 4 Out of sample extension

3.3.2 Classification

For classification, SVM is used. Gaussian kernel is utilized as the kernel function of the SVM. For the parameters γ and C, grid search is used to get the optimum parameters for each dimension reduction method . For the grid search, 2^{-10} - 2^{30} and 2^{-60} - 2^{15} intervals are used for C and γ respectively. Then accuracy is measured.

Parameters are also verified with a comparison of the SVM with KNN for *k*=1 classifier.

SVM is trained with 900 sample subset of images with reduced dimension. 100 samples are gathered from each 9 class. That is 9,6% of the complete data set. The training samples are chosen randomly (see Fig 5).

Accuracy is calculated using the ground truth of the Indian Pine and classified image.

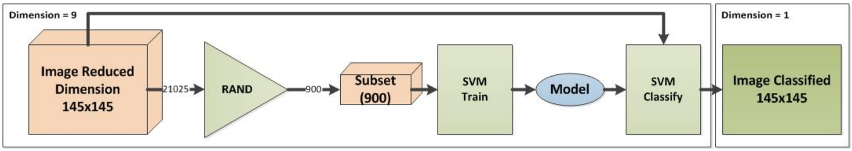


Fig. 5 Classifying phase

3.4 *Experimental Results*

The results are obtained for different dimension reduction techniques by means of accuracy and calculation time.

Calculation time is acquired during dimension reduction and out of sample extension phase. Classification time is not included.

Each method is run and classified 9 times, and majority voting is applied to results of these classifications. Accuracy is calculated comparing the output of the majority voting and ground truth of the image. The results are given in Table 2.

Table 2 Experimental Results with respect to accuracy and time

Linearity	Training	Method	Accuracy (0-1)		Time (s)	
			Sub-Sampled	Entire Data	Sub-Sampled	Entire Data
Linear	Unsupervised	FA	0,88	0,88	1,5	13
		LPP	0,83	0,83	0,11	12
		NPE	0,83	0,81	0,15	12
		PCA	0,87	0,87	0,07	0,28
		PPCA	0,85	0,84	4,36	44
	Supervised	LDA	0,84	0,88	0,04	0,09
		MCML	0,83	0,83	130	18796
		NCA	0,84	0,90	323	38052
Non-Linear	Unsupervised	AE	0,74	0,82	5,5	57
		FMVU	0,68	0,71	108	576
		ISO	0,72	0,76	34	1396
		MANC	0,84	0,78	1,41	15

According to accuracy results, non-linear methods are highly affected from reducing the input sample. Even though their calculation times are better, accuracy gets worse with the decrease in input sample.

For supervised linear techniques, as the number of input samples decreases, the learning set decreases. Hence a drop in the accuracy is expected. The techniques that use Mahalanobis distance have very long calculation time with entire data set and are not feasible to use with hyperspectral images because of the complex process of Mahalanobis metric calculation. But with the sub-sampled data set, the calculation times are appropriate, i.e. 0,6% for MCML and 0,8% for NCA of the entire data version. Furthermore, for MCML method, there is no decrease in accuracy. Therefore that sub-sampling has no negative effect in the aspect of accuracy.

Sub-sampling image has no adverse effect on the unsupervised linear techniques in accuracy manner even there is a slight increase in some methods. But calculation times are far better. Sub-sampling has a certain performance improvement on unsupervised linear techniques.

For non-linear methods, there is a tradeoff between time and accuracy. Some techniques, such as Manifold Charting, have already short implementation time for entire set. However when using methods like Isomap, reduction in the sample size should be considered.

4 Conclusion

This paper presents the effect of reducing the sample of data set on different dimension reduction techniques on hyperspectral image set. The effect is tested on 12 dimension reduction methods with SVM classifier.

For non-linear techniques, there is a trade of between sub-sampling and using the entire image in terms of accuracy and calculation time.

There is an improvement in performance when sub-sampling is used with linear method on hyperspectral images.

References

[1] Fodor, I.K.: A Survey of Dimension Reduction Techniques. UCRL-ID-148494 (May 9, 2002)

[2] AVIRIS NW Indiana's Indian Pines 1992 data set. (data), `ftp://ftp.ecn.purdue.edu/biehl/MultiSpec/92AV3C.tif.zip` (ground truth), `ftp://ftp.ecn.purdue.edu/biehl/MultiSpec/ThyFiles.zip` (access Date: May 07, 2012)

[3] Rodarmel, C., Shan, J.: Principal component analysis for hyperspectral. image classification. Surv. Land Inf. Sci. 62(2), 115–122

[4] Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290(5500), 2319–2323 (2000)

[5] He, X., Niyogi, P.: Locality preserving projections. In: Advances in Neural Information Processing Systems, vol. 16, p. 37. The MIT Press, Cambridge (2004)

[6] Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500), 2323–2326 (2000)

[7] He, X., Cai, D., Yan, S., Zhang, H.-J.: Neighborhood preserving embedding. In: Proc. ICCV (2005)

[8] Jacob, G., Roweis, S., Hinton, G., Salakhutdinov, R.: Neighbourhood components analysis. In: Advances in Neural Information Processing Systems 17 (2004)

[9] Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science 313(5786), 504–507 (2006)

[10] Rosipal, R., Girolami, M.: An expectation-maximization approach to nonlinear component analysis. Neural Comput. 13(3), 505–510 (2001)

[11] Bishop, T.: Probabilistic principal component analysis. T.Journal of the Royal Statistical Society, Series B, 61, Part 3, 611–622 (1999)

[12] Globerson, A., Roweis, S.: Metric Learning by Collapsing Classes. In: NIPS (2005)

[13] Weinberger, K.Q., Saul, L.K.: Unsupervised learning of image manifolds by semidefinite programming. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2004), Washington DC, vol. 2, pp. 988–995 (2004)

[14] van der Maaten, L.J.P.: An introduction to dimensionality reduction using Matlab. Technical Report 07-06, MICC-IKAT, Maastricht University, Maastricht, The Netherlands (2007)

[15] Brand, M.: Charting a manifold. In: Advances in Neural Information Processing Systems, vol. 15, pp. 985–992. The MIT Press, Cambridge (2002)

[16] Parasad, S., Bruce, L., Chanussot, J.: Optical Remote Sensing: Advances in Signal Processing and Exploitation Techniques (Augmented Vision and Reality) ISBN-10: 3642142117

[17] van der Maaten, L.J.P., Postma, E.O., van den Herik, H.J.: Dimensionality Reduction: A Comparative Review, Tilburg University Technical Report, TiCC-TR 2009-005 (2009)

[18] Fukunaga, K.: Introduction to Statistical Pattern Recognition. Academic Press Professional, Inc., San Diego (1990)

[19] Levina, E., Bickel, P.J.: Maximum likelihood estimation of intrinsic dimension. In: Advances in Neural Information Processing Systems, vol. 17. The MIT Press, Cambridge (2004)

Author Index

Akbulut, Ahmet 245
Andrešič, David 475
Andziulis, Arunas 233
Augustová, Petra 305

Bednar, Hynek 333
Beran, Zdeněk 305
Bialic-Davendra, Magdalena 31
Brandejsky, Tomas 181
Bromová, Pavla 467
Bucki, Robert 423
Burguillo, Juan C. 323

Čelikovský, Sergej 313
Chadli, Mohammed 47, 61, 163
Chalupa, Petr 285, 295
Chen, Guanrong 1
Chen, Yi 401
Chramcov, Bronislav 423
Clementis, Ladislav 191

Davendra, Donald 31, 47, 61, 77, 89
Dohnálek, Pavel 487, 497
Dorronsoro, Bernabé 323
Dostál, Petr 155
Dostal, Petr 265
Dubec, Patrik 125
Dzik, Petr 455

Elamvazuthi, I. 147

Gajdoš, Petr 487, 497, 517
Ganesan, T. 147
Gazdoš, František 275

Hanzal, Petr 385
Hotař, Vlastimil 343

Ilgın, Hakkı Alparslan 507, 529

Jakovlev, Sergej 233
Jakša, Rudolf 363
Janošek, Michal 413
Januška, Peter 295
Jašek, Roman 435
Jehlička, Vladimír 257

Kocian, Václav 413
Kolařík, Jaroslav 275
Koščák, Juraj 363
Kozal, Ali Ömer 529
Kříž, Radko 353
Kuznetsov, Nikolay V. 5

Lampinen, Jouni 47, 61
Lebedik, Anastasia Slustikova 201
Leitmanová, Ivana Faltová 385
Leonov, Gennady A. 5
Lozi, René 17
Lynnyk, Volodymyr 313

Marholt, Jiří 275
Marusza, Sabina 423
Matousek, Radomil 137
Miksovsky, Jiri 333
Minar, Petr 137
Moravec, Pavel 487

Novák, Jakub 285, 295

Oplatkova, Zuzana Kominkova 101, 211

Partila, Pavol 221
Peterek, Tomáš 487, 497
Plucar, Jan 125
Pluhacek, Michal 31, 47, 61, 77, 89, 101
Prešovský, Karol 373

Radecký, Michal 517
Raidl, Ales 333
Rapant, Lukáš 125
Ruibys, Kestutis 233
Ruzek, Martin 175

Šaloun, Petr 111, 475
Sanjuan, Miguel A.F. 3
Senkerik, Roman 31, 47, 61, 77, 89, 101, 211
Sepeši, Rudolf 363
Shaari, Ku Zilati Ku 147
Sinčák, Peter 363, 373
Skanderova, Lenka 111
Škoda, Petr 15, 467, 475
Szmit, Anna 435
Szmit, Maciej 435

Teke, Mustafa 529

Vala, Martin 455
Valis, David 445
Vasant, P. 147
Vaščák, Ján 373
Vesely, Michal 455
Vojtesek, Jiri 265
Volná, Eva 413
Vozňák, Miroslav 517
Voznak, Miroslav 221, 233

Walek, Agata 445

Yıldırım, Burak 507
Youssef, T. 163
Yuceol, Ozan Eren 245

Zak, Libor 445
Zelinka, Ivan 31, 47, 61, 77, 89, 101, 111, 163, 201, 401, 475
Zelmat, M. 163
Zendulka, Jaroslav 467
Zhang, Guanfeng 401
Zheng, Bin 401
Zmeskal, Oldrich 455