

Wen Yu
Edgar N. Sanchez (Eds.)

Advances in Computational Intelligence

 Springer

Advances in Intelligent and Soft Computing

Editor-in-Chief

Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series can be found on our homepage: springer.com

Vol. 46. V.-N. Huynh, Y. Nakamori,
H. Ono, J. Lawry,
V. Kreinovich, H.T. Nguyen (Eds.)
*Interval / Probabilistic Uncertainty and
Non-classical Logics*, 2008
ISBN 978-3-540-77663-5

Vol. 47. E. Pietka, J. Kawa (Eds.)
Information Technologies in Biomedicine, 2008
ISBN 978-3-540-68167-0

Vol. 48. D. Dubois, M. Asunción Lubiano,
H. Prade, M. Ángeles Gil,
P. Grzegorzewski,
O. Hryniewicz (Eds.)
*Soft Methods for Handling
Variability and Imprecision*, 2008
ISBN 978-3-540-85026-7

Vol. 49. J.M. Corchado, F. de Paz,
M.P. Rocha,
F. Fernández Riverola (Eds.)
*2nd International Workshop on Practical
Applications of Computational Biology
and Bioinformatics (IWPACBB 2008)*, 2009
ISBN 978-3-540-85860-7

Vol. 50. J.M. Corchado, S. Rodriguez,
J. Llinas, J.M. Molina (Eds.)
*International Symposium on Distributed
Computing and Artificial Intelligence 2008
(DCAI 2008)*, 2009
ISBN 978-3-540-85862-1

Vol. 51. J.M. Corchado, D.I. Tapia,
J. Bravo (Eds.)
*3rd Symposium of Ubiquitous
Computing and Ambient
Intelligence 2008*, 2009
ISBN 978-3-540-85866-9

Vol. 52. E. Avineri, M. Köppen,
K. Dahal,
Y. Sunitiyoso, R. Roy (Eds.)
Applications of Soft Computing, 2009
ISBN 978-3-540-88078-3

Vol. 53. E. Corchado, R. Zunino,
P. Gastaldo, Á. Herrero (Eds.)
*Proceedings of the International
Workshop on Computational
Intelligence in Security for
Information Systems CISIS 2008*, 2009
ISBN 978-3-540-88180-3

Vol. 54. B.-y. Cao, C.-y. Zhang,
T.-f. Li (Eds.)
Fuzzy Information and Engineering, 2009
ISBN 978-3-540-88913-7

Vol. 55. Y. Demazeau, J. Pavón,
J.M. Corchado, J. Bajo (Eds.)
*7th International Conference on Practical
Applications of Agents and Multi-Agent
Systems (PAAMS 2009)*, 2009
ISBN 978-3-642-00486-5

Vol. 56. H. Wang, Y. Shen,
T. Huang, Z. Zeng (Eds.)
*The Sixth International Symposium on Neural
Networks (ISNN 2009)*, 2009
ISBN 978-3-642-01215-0

Vol. 57. M. Kurzynski,
M. Wozniak (Eds.)
Computer Recognition Systems 3, 2009
ISBN 978-3-540-93904-7

Vol. 58. J. Mehnen, A. Tiwari,
M. Köppen, A. Saad (Eds.)
Applications of Soft Computing, 2009
ISBN 978-3-540-89618-0

Vol. 59. K.A. Cyran,
S. Kozielecki, J.F. Peters,
U. Stańczyk, A. Wakulicz-Deja (Eds.)
Man-Machine Interactions, 2009
ISBN 978-3-642-00562-6

Vol. 60. Z.S. Hippe,
J.L. Kulikowski (Eds.)
Human-Computer Systems Interaction, 2009
ISBN 978-3-642-03201-1

Vol. 61. W. Yu, E.N. Sanchez (Eds.)
Advances in Computational Intelligence, 2009
ISBN 978-3-642-03155-7

Wen Yu, Edgar N. Sanchez (Eds.)

Advances in Computational Intelligence

Editors

Wen Yu
Departamento de Control Automatico
CINVESTAV-IPN
Av. IPN 2508
Mexico D.F., 07360
Mexico

Edgar N. Sanchez
CINVESTAV-Guadalajara
Av. Cientifica 1145
Guadalajara, Jalisco, 45015
Mexico

ISBN 978-3-642-03155-7

e-ISBN 978-3-642-03156-4

DOI 10.1007/978-3-642-03156-4

Advances in Intelligent and Soft Computing

ISSN 1867-5662

Library of Congress Control Number: Applied for

©2009 Springer-Verlag Berlin Heidelberg

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable for prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typeset & Cover Design: Scientific Publishing Services Pvt. Ltd., Chennai, India.

Printed in acid-free paper

5 4 3 2 1 0

springer.com

Preface

This book constitutes the proceedings of the second International Workshop on Advanced Computational Intelligence (IWACI 2009), with a sequel of IWACI 2008 successfully held in Macao, China. IWACI 2009 provided a high-level international forum for scientists, engineers, and educators to present state-of-the-art research in computational intelligence and related fields.

Over the past decades, computational intelligence community has witnessed tremendous efforts and developments in all aspects of theoretical foundations, architectures and network organizations, modelling and simulation, empirical study, as well as a wide range of applications across different domains. IWACI 2009 provided a great platform for the community to share their latest research results, discuss critical future research directions, stimulate innovative research ideas, as well as facilitate international multidisciplinary collaborations.

IWACI 2009 received 146 submissions from about 373 authors in 26 countries and regions (Australia, Brazil, Canada, China, Chile, Hong Kong, India, Islamic Republic of Iran, Japan, Jordan, Macao, Malaysia, Mexico, Pakistan, Philippines, Qatar, Republic of Korea, Singapore, South Africa, Sri Lanka, Spain, Taiwan, Thailand, UK, USA, Venezuela, Vietnam, and Yemen) across six continents (Asia, Europe, North America, South America, Africa, and Oceania). Based on the rigorous peer reviews by the Program Committee members, 52 high-quality papers were selected for publication in this book, with an acceptance rate of 36.3%. These papers cover major topics of the theoretical research, empirical study, and applications of computational intelligence.

In addition to the contributed papers, the IWACI 2009 technical program included four plenary speeches by Hojjat Adeli (The Ohio State University, USA), Reza Langari (Texas A&M University, USA), Jun Wang (Chinese University of Hong Kong), and Jacek M. Zurada (University of Louisville, USA). As organizers of IWACI 2009, we would like to express our sincere thanks to the Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV-IPN), Consejo Nacional de Ciencia y Tecnología (CONACyT), and Academia Mexicana de Ciencias (AMC). We would also like to sincerely thank the Advisory Committee Chairs for their guidance in every aspect of the entire conference, and Organizing Committee Chairs for overall organization of the workshop. We want to take this opportunity to express our deepest gratitude to the members of the International Program Committee for their professional review of the papers; their expertise guaranteed the high quality of technical program of IWACI 2009! Furthermore, we thank Springer for publishing the proceedings in the prestigious series of *Advances in Intelligent and Soft Computing*.

Finally, we would like to thank all the speakers, authors, and participants for their great contribution and support that made IWACI 2009 a great success.

October 2009

Wen Yu
Edgar Nelson Sanchez

VIII Organization

Huberto Sossa

Yu Tang

Zhigang Zeng

Jie Zhang

Huaguang Zhang

CIC-IPN, Mexico

Universidad Nacional Autónoma de México, Mexico

Huazhong University of Science and Technology,
China

University of New Castle, Newcastle, UK

Northeastern University, China

Publications Chair

Xiaou Li

CINVESTAV-IPN, Mexico

Registration Chairs

Gary G. Feng

Jose de Jesus Rubio

City University of Hong Kong, Hong Kong

Azcapozalco-IPN, Mexico

Table of Contents

Session 1. Neural Networks

Multi Lingual Speaker Recognition Using Artificial Neural Network	1
<i>Prateek Agrawal, Anupam Shukla, and Ritu Tiwari</i>	
Modeling Huntington's Disease Considering the Theory of Central Pattern Generators (CPG)	11
<i>Masood Banaie, Yashar Sarbaz, Mohammad Pooyan, Shahriar Gharibzadeh, and Farzad Towhidkhal</i>	
Prophelia: Artificial Intelligence for TravelBox® Technology	21
<i>R. Chulaka Gunasekara, B.B. Akila Geethal, Mafaz Hassan, C.D. Tharindu Mathew, A. Shehan Perera, Harsha Subasinghe, Yohan Welikala, and Lasantha Wimalasena</i>	
Application Research of Local Support Vector Machines in Condition Trend Prediction of Reactor Coolant Pump	35
<i>Guohua Yan and Yongsheng Zhu</i>	
Asymptotic Synchronization for Pulse-Coupled Oscillators with Delayed Excitatory Coupling Is Impossible	45
<i>Wei Wu and Tianping Chen</i>	
Missing Data Imputation through the Use of the Random Forest Algorithm	53
<i>Adam Pantanowitz and Tshilidzi Marwala</i>	
Ubiquitous Middleware Using Mobility Prediction Based on Neuro-Association Mining for Adaptive Distributed Object System	63
<i>Romeo Mark A. Mateo, Mabrey Lee, and Jaewan Lee</i>	
A Growing Algorithm for RBF Neural Network	73
<i>Han Honggui and Qiao Junfei</i>	
Fault Tolerance Based on Neural Networks for the Intelligent Distributed Framework	83
<i>Michael Angelo G. Salvo, Jaewan Lee, and Jung-sik Lee</i>	
Learning RNN-Based Gene Regulatory Networks for Robot Control	93
<i>Wei-Po Lee and Tsung-Hsien Yang</i>	
Fault Detection for Networked Control Systems via Minimum Error Entropy Observer	103
<i>Jianhua Zhang, Lei Cai, and Hong Wang</i>	

Discrete-Time Reduced Order Neural Observers 113
Alma Y. Alanis and Edgar Nelson Sanchez

A New Neural Observer for an Anaerobic Wastewater Treatment
 Process 123
*Rubén Belmonte-Izquierdo, Salvador Carlos-Hernández, and
 Edgar Nelson Sanchez*

Prediction of Protein Subcellular Multi-Localization by Using a
 Min-Max Modular Support Vector Machine 133
Yang Yang and Bao-Liang Lu

Application of MultiLayer Perceptron Type Neural Network to Camera
 Calibration 145
Dong-Min Woo and Dong-Chul Park

Hierarchical Neural Network Model for Water Quality Prediction in
 Wastewater Treatment Plants 155
Qiumei Cong, Wen Yu, and Tianyou Chai

Third Generation Neural Networks: Spiking Neural Networks 167
Samanwoy Ghosh-Dastidar and Hojjat Adeli

Session 2. Fuzzy Systems

Choquet Fuzzy Integral Applied to Stereovision Matching for Fish-Eye
 Lenses in Forest Analysis 179
*P. Javier Herrera, Gonzalo Pajares, María Guijarro,
 José J. Ruz, and Jesús M. de la Cruz*

Fuzzy OLAP: A Formal Definition 189
Claudia González, Leonid Tineo, and Angélica Urrutia

Caller Behaviour Classification: A Comparison of SVM and FIS
 Techniques 199
Pretesh B. Patel and Tshilidzi Marwala

A Dual-Model Discrete-Time Jumping Fuzzy System Approach to NCS
 Design 209
Fengge Wu, Fuchun Sun, and Huaping Liu

A Continuous-Time Recurrent Neurofuzzy Network for Black-Box
 Modeling of Insulin Dynamics in Diabetic Type-1 Patients 219
*Marcos Angel González-Olvera, Ana Gabriela Gallardo-Hernández,
 Yu Tang, Maria Cristina Revilla-Monsalve, and Sergio Islas-Andrade*

Vague Query Based on Vague Relational Model 229
Faxin Zhao and Z.M. Ma

Identification and Speed Control of a DC Motor Using an Input-Output Recurrent Neurofuzzy Network	239
<i>Edgar Berrospe, Marcos Angel González-Olvera, and Yu Tang</i>	
Hybrid Intelligent Control Scheme for an Anaerobic Wastewater Treatment Process	249
<i>Rubén Belmonte-Izquierdo, Salvador Carlos-Hernández, and Edgar Nelson Sanchez</i>	
Session 3. Evolutionary Algorithms	
Workability of a Genetic Algorithm Driven Sequential Search for Eigenvalues and Eigenvectors of a Hamiltonian with or without Basis Optimization	259
<i>Subhajit Nandy, Pinaki Chaudhury, and Shankar Prasad Bhattacharyya</i>	
An Improved Quantum Evolutionary Algorithm Based on Artificial Bee Colony Optimization	269
<i>Haibin Duan, Zhihui Xing, and Chunfang Xu</i>	
Rough Approximation Operators with Hedges	279
<i>Xueyou Chen</i>	
An Evolutionary Algorithm with Lower-Dimensional Crossover for Solving Constrained Engineering Optimization Problems	289
<i>Yulong Shi, Sanyou Zeng, Bo Xiao, Yang Yang, and Song Gao</i>	
Gene Regulatory Network Reconstruction of P38 MAPK Pathway Using Ordinary Differential Equation with Linear Regression Analysis	299
<i>Ming Zheng, Gui-Xia Liu, Han Wang, and Chun-Guang Zhou</i>	
A Novel Multi-threshold Segmentation Approach Based on Artificial Immune System Optimization	309
<i>Erik Cuevas, Valentín Osuna-Enciso, Daniel Zaldívar, and Marco Pérez-Cisneros</i>	
Research on Improvement Strategies and Parameter Analysis of Ant Colony Algorithm for One-Dimensional Cutting Stock Problem	319
<i>Lan Huang, Huayi Chen, Bin Yang, and Chun-Guang Zhou</i>	
Mixture of Experts with Genetic Algorithms	331
<i>Laura Cleofas, Rosa Maria Valdovinos, and C. Juárez</i>	
Opposition-Based Particle Swarm Optimization with Velocity Clamping (OVCPSO).....	339
<i>Farrukh Shahzad, A. Rauf Baig, Sohail Masood, Muhammad Kamran, and Nawazish Naveed</i>	

Designing a Compact Genetic Algorithm with Minimal FPGA Resources 349
Alejandro León-Javier, Marco A. Moreno-Armendáriz, and Nareli Cruz-Cortés

Session 4. Intelligent Techniques and Applications

Application of DNA Self-assembly on Maximum Clique Problem 359
Guangzhao Cui, Cuiqing Li, Haobin Li, Xuncaizhang, and Xiaoguang Li

Modeling of the Relative Humidity and Control of the Temperature for a Bird Incubator 369
Jose de Jesus Rubio, Martin Salazar, Raul Lugo, Jaime Pacheco, and Angel D. Gomez

A Novel Automatic Method on Diagnosing Movement Disorders 379
Masood Banaie, Mohammad Mikaili, and Mohammad Pooyan

The Fault Diagnosis of Electric Railway Traction Substation with Model-Based Diagnosis of Integration of FDI and DX Approaches 387
Zhigang Liu, Zhiwei Han, and Wei Zhong

A Single-Hop Active Clustering Algorithm for Wireless Sensor Networks 397
Fengjun Shang

A Transelevator Moving Inside of an Automatic Warehouse in Virtual Reality 407
Jose de Jesus Rubio, Enrique Garcia, and Jaime Pacheco

Improved AFT and Background-Mesh Generation for FEM with Applications in Microwave 415
Xin Li, Changying Wu, Jie Li, and Jiadong Xu

Application of Non-redundant Association Rules in University Library 423
Xiukuan Chen, Xiangjun Dong, Yuming Ma, and Runian Geng

Global Structure Constraint: A Fast Pre-location for Active Appearance Model 433
Jie Ma, Dongwei Guo, Miao Liu, Kangping Wang, and Chun-Guang Zhou

Bio-inspired Architecture for Visual Recognition of Humans Walking ... 443
Pedro Luis Sánchez Orellana, Claudio Castellanos Sánchez, Edgar del Angel-Guerrero, and Tomás Martínez-Arenas

Computational Model for Electric Fault Diagnosis in Induction Motors	453
<i>Rodrigo López-Cárdenas, Luis Pastor Sánchez-Fernández, and Sergio Suárez-Guerra</i>	
Closed-Loop Identification of a Nonlinear Servomechanism: Theory and Experiments	463
<i>Rubén Garrido and Roger Miranda</i>	
Dynamic Classifier Selection with Confidence Intervals	473
<i>Rose Maria Valdovinos, M. Sánchez, and Issachar Ruiz</i>	
Optimal Neuron-Controller for Fluid Triple-Tank System via Improved ADDHP Algorithm	483
<i>Shaojian Song, Gang Cai, and Xiaofeng Lin</i>	
Method of Learning for Life Pet Artificial	493
<i>Rodolfo Romero Herrera, Francisco Gallegos Funes, and Antonio Gustavo Juárez Gracia</i>	
An Sliding Mode Control for an Elbow Arm	503
<i>Jose de Jesus Rubio, Jaime Pacheco, and Gerardo Villegas</i>	
Stabilization on a Physical Pendulum with Moving Mass	509
<i>Oscar Octavio Gutiérrez-Frías, Juan Carlos Martínez-García, and Ruben Garrido-Moctezuma</i>	
Restricted Growth String for Video-Type Classification	519
<i>Pedro Luis Sánchez Orellana, José Torres Jiménez, and Claudio Castellanos Sánchez</i>	
Author Index	527

Multi Lingual Speaker Recognition Using Artificial Neural Network

Prateek Agrawal, Anupam Shukla, and Ritu Tiwari

ABV-Indian Institute of Information Technology and Management, Gwalior, India
{prateek061186, dranupamshukla, tiwari.ritu2}@gmail.com

Abstract. This paper describes a method for speaker identification in multiple languages that is based on Back Propagation Algorithm (BPA). The identification process goes through recording the speech utterances of different speakers in different languages, features extraction, data clustering and system training using BPA. Our database contains one sentence in 8 different Indian regional languages i.e. Hindi, English, Assami, Telugu, Punjabi, Rajasthani, Marathi & Bengali, spoken by 32 speakers in each language. With total size of 904 speech utterances, the Average performance of the system is 95.354%. These applications are mainly used in speaker Authentication, in telephony applications where the conversations can be of short durations and the language could change from one conversation to another etc.

Keywords: Multilingual Speaker Recognition.

1 Introduction

In our daily life there are many forms of communication like textual language, body language and speech. Amongst those forms speech is always regarded as the most powerful form because of its rich dimensional characters. Such information is very important for effective communication. From the signal processing point of view, speech can be characterized in terms of the signal carrying message information. The waveform could be one of the representations of speech [1]. A practical approach to multilingual speech recognition for countries like India where more than 25 languages are spoken across the country would be to have a truly multilingual acoustic model. This multilingual model should then be adapted to the target language with the help of a language identification system. Based on the information extracted from the speech signal, it can have three different recognition system itself: Speaker Recognition, Language Recognition and Speech Text Recognition [12, 13, 15]. Speaker recognition can be divided into speaker verification and speaker identification. The objective of a speaker verification system is to verify whether an unknown voice matches the voice of a speaker whose identity is being claimed. In speaker identification, we want to identify an unknown voice from a set of known voices. Speaker verification systems are mainly used in security access control while speaker identification systems are mainly used in criminal investigation [3]. Speech recognition comprises a large number of complex applications such as speech driven consumer applications, speech

commissioning in logistics, checking & recording in quality assurance work etc.. This paper focuses particularly on the problem of recognizing simple isolated words, speaker and language.

1.1 Back Propagation Training

ANN have emerged as a promising approach to the problem of speech recognition [6, 7]. ANNs have advantages to handle speech and speaker recognition as they are able to execute a high level of parallel computation; they have high level of robustness and fault tolerance capability. They can learn complex features from the data, due to the non-linear structure of artificial neuron [6, 8]. Various ANN training algorithms such as BPA, Radial Basis Function, Recurrent networks etc., are being used for training purpose. In this work, the BPA has been used to train the network. Back Propagation (BP) provides the recognition component in this system. The architecture of BPA is shown in Figure 1.

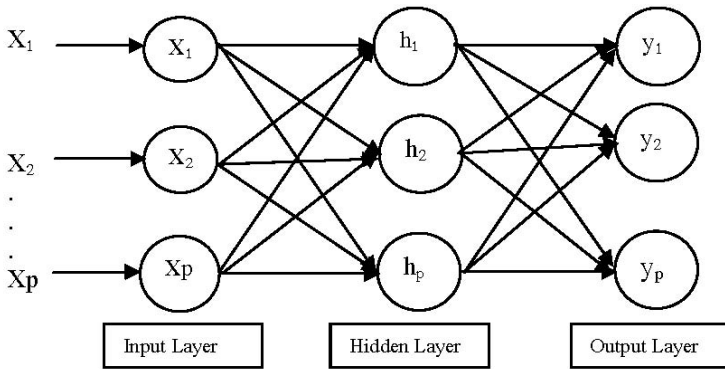


Fig. 1. Architecture of Back Propagation Algorithm

1.2 Clustering

The key task of clustering algorithm is to automatically find groups (clusters) of data in such a manner that the profiles of objects in the same cluster are very similar and the profiles of objects in different clusters are quite distinct [10]. Here we are using this technique to cluster the speech data with similar properties in feature space. The statement about ‘similarity’, a distance measure between two speakers is to be defined. It was earlier shown by the authors [11] that clustering can be an effective tool to break the whole high dimensional input space into reduced input space without any change in dimensionality and also to train ANN with extremely large data. We have taken large data set for experiment, since the structure of the model is complex. We used “Fuzzy C-Means (FCM) Algorithm” for clustering the data [16, 18]. In this clustering algorithm, first we decided, in how many clusters we are clustering the data and then assign to each point coefficients for being in the clusters and repeat this process until the algorithm has converged (i.e. the coefficient change between two iterations is no more than the given sensitivity threshold).

2 Previous Work

Most existing Automatic Speaker Recognition (ASR) systems used for practical application are of the small vocabulary or isolated word type. Medium- and large vocabulary systems perform well in laboratories but not in real life [3]. The research have been done for multilingual speaker identification system is by using statistical methods like Hidden Markov Model (HMMs), Harmonic Product Spectrum (HPS) algorithm [4]. ANNs have been used to identify the speakers in single language [3]. For the pattern classification phase, various methods have been used as vector quantized codebooks to store the features [17]. No research has been done for speaker recognition with eight different languages.

3 Features of Speech

Features of any signal play an important role in pattern recognition. Features are useful to separate one speaker from other[14]. The basic features of speech like average power spectral density, Cepstrum Coefficient, No. of Zero crossings, Length of File have been extracted using MATLAB.

3.1 Cepstrum Coefficient

Cepstrum coefficient is the transform of the real logarithm of the magnitude of the most distinctive feature that helps to differentiate the speakers. The real cepstrum is the inverse Fourier Transform of a sequence. C_x returns the real cepstrum of the real sequence x . The real cepstrum is a real-valued function. The real cepstrum of a signal x , sometimes called simply the cepstrum, is calculated by determining the natural logarithm of magnitude of the Fourier transform of x , then obtaining the inverse Fourier transform of the resulting sequence, as in (1),

$$C_x = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log |X(e^{jw})| e^{jwn} dw \quad (1)$$

3.2 Average PSD

This is another important feature of speech signal. PSD is intended for continuous spectra. The integral of the PSD over a given frequency band computes the average power in the signal over that frequency band. In contrast to the mean-squared spectrum, the peaks in the spectra do not reflect the power at a given frequency. Average PSD of any signal can be calculated as the ratio of total power to the frequency of the signal.

3.3 No. of Zero Crossing

Zero-crossing signals have a direction attribute, which can have three values Rising, Falling and Either. In Rising, when a signal rises to zero to or through zero, or when a signal leaves zero and becomes positive zero crossing occurs. In case of Falling, a

zero crossing occurs when signal falls to or through zero, or when a signal leaves zero and becomes negative. In case of Either, a zero crossing occurs if either a Rising or Falling occurs.

3.4 Length of File

Length of the *.wav file is another feature that has been calculated. The maximum size of the file is being measured that help us to distinguish one file to another. Total no. of counter are being counted to find out the max size of the file.

4 Approach

Proposed system is done in various steps like Collection of speech utterances from different speakers, Preprocessing of speech utterances, Features Extraction, Clustering of Entire Featured Data, Neural Network Training and Simulation as shown in Figure No. 2.

The speech used in *Collection of Speech utterances* experiment consists of word sequences taken from the public domain. Sound files (*.wav) are created by using Microphone hardware interfaced with personal Computer at sampling rate of 44.1 KHz. For developing the Speech database one sentence (“*ab iss baar aap*”) is recorded from 32 speakers (19 male & 13 female) in 8 different languages. The sentence is in such a way that in each word every consonant succeeds a vowel and vice versa. The reason behind this is, whenever we pronounce any letter a vowel sound is always generated. For the sentence considered four words are used in few languages, three in some and two in other and collectively 904 words are recorded in eight different Indian languages.

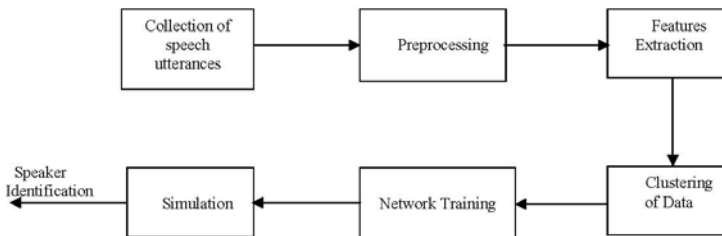


Fig. 2. Processing steps of proposed approach

In *Preprocessing of Speech Utterances* each word is clipped from the sentence of particular language. Silence and noise are removed from speech signal as shown in Figure No. 3. Speech database is created using these clipped speech utterances and these speech signal have been arranged in proper manner for further use. Next step is extracting the features from these speech signals with the help of MATLAB (A computational language). Four types of features are extracted - Cepstrum coefficient, number of zero crossing, average PSD, length of file. In these features, cepstrum coefficient is the main feature which has maximum knowledge. We clustered the

whole data into different groups by using FCM clustering algorithm. FCM clustering is applied on each feature independently because this clustering scheme gives better generalization. FCM is a data clustering technique wherein each data point belongs to a cluster to some degree that is specified by a membership grade. It provides a method that shows how to group data points that populate some multidimensional space into a specific number of different clusters. Each cluster has its cluster center that helps to separate similar patterns into different data sets. These datasets are created on the basis of Nearest Neighbor Method. The Nearest Neighbor Method chooses one of the articulatory clusters based on the Euclidian distance between each of the cluster centroid vectors and each of the articulatory vectors to be trained.

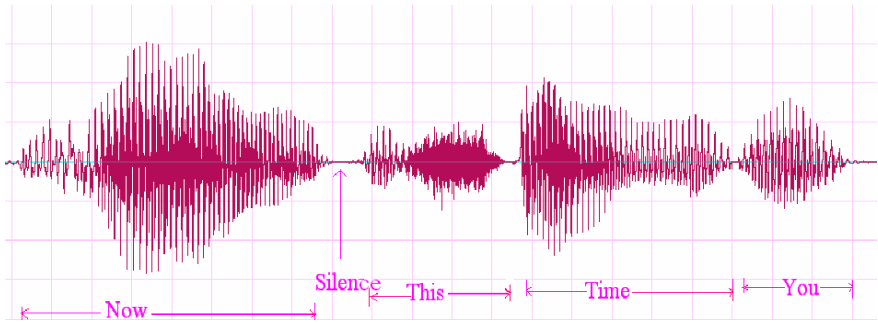


Fig. 3. Preprocessing of speech sentence (in English) into utterances

Each cluster of the system is trained separately by using BPA with single hidden layer and numbers of hidden neurons are being kept less than the target numbers. The training parameters for each network i.e. error goal (δ), momentum (μ), maximum epochs, non linear function, number of hidden layers, number of neurons per layer, number of targets, training parameter (α) are illustrated in Table 1. One lookup table is being created that tells us the appropriate position of each input data in respective cluster. One training curve is shown in Figure 4.

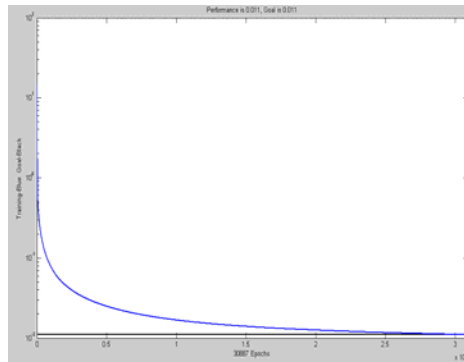


Fig. 4. Training graph for cluster no. 2

Table 1. Various Parameter Values for Training

Cluster No.	Error Goal (δ)	No. of Neurons/Layer	No. of Targets	Training parameter (α)	Max. Epochs
1	0.012	28	31	0.25	40,000
2	0.011	52	58	0.26	40,000
3	0.014	15	19	0.26	25,000
4	0.01	40	44	0.26	50,000
5	0.009	56	58	0.28	50,000
6	0.012	44	55	0.28	25,000
7	0.009	52	56	0.28	50,000
8	0.016	20	25	0.24	50,000
9	0.013	20	28	0.24	50,000
10	0.008	78	80	0.27	50,000
11	0.01	42	45	0.25	50,000
12	0.0118	36	45	0.25	22,000
13	0.0118	40	50	0.25	30,000
14	0.011	27	29	0.28	50,000
15	0.008	71	73	0.26	50,000
16	0.01	36	42	0.25	50,000
17	0.01	33	37	0.25	40,000
18	0.01	24	27	0.24	50,000
19	0.01	52	57	0.27	30,000
20	0.01	38	43	0.26	50,000

Training Momentum (μ) = 0.9.

Non-linear function = tan-sigmoid.

No. of hidden layer/network = 1.

After completion of the training, the next step is to simulate each trained network separately and check whether the class of actual output and target output is same or not. For simulation, a sample data is taken from the input set for testing and it is simulated with the trained network. One data lookup table is maintained that gives us the information about the position of each word in appropriate cluster that are being spoken by different speakers in different languages.

5 Results

When the system is trained with single ANN (traditional approach) as used by other researchers for same problem with limited number of clusters, the performance was 18%. Complete data set is being clustered into 20 different groups numbered from 1 to 20 as shown in Table 2. Each data cluster is trained independently using BPA and tested with the sample taken from database. Error values after testing the trained data have been calculated finding overall performance of the system. Efficiency of each cluster is being calculated separately. In total 20 clusters of 904 input data, average performance of the system is **95.354%** with 42 errors. Also the time taken was much less in the proposed approach as compared to the traditional. Mak, M. W. et. al. [3] described a speaker identification system based on RBF networks with highest

efficiency of 95.7% and their database contains 7 sentences and 10 digits spoken by 20 speakers while we have worked with 904 input data. With a lot number of speakers, languages and words, the average performance of our system is equivalent to their system best performance.

Table 2. Training Result Data

Cluster No.	No. of Input Utterances	No. of Errors	Efficiency (%)
1	31	1	96.77
2	58	5	91.4
3	19	0	100
4	44	2	95.45
5	58	5	91.4
6	55	4	92.73
7	56	5	91.07
8	25	0	100
9	28	0	100
10	80	5	93.75
11	47	2	95.74
12	45	2	95.56
13	50	4	92
14	29	0	100
15	73	3	95.89
16	42	1	97.62
17	37	0	100
18	27	0	100
19	57	3	94.74
20	43	0	100
Total	$\Sigma = 904$	$\Sigma = 42$	95.354

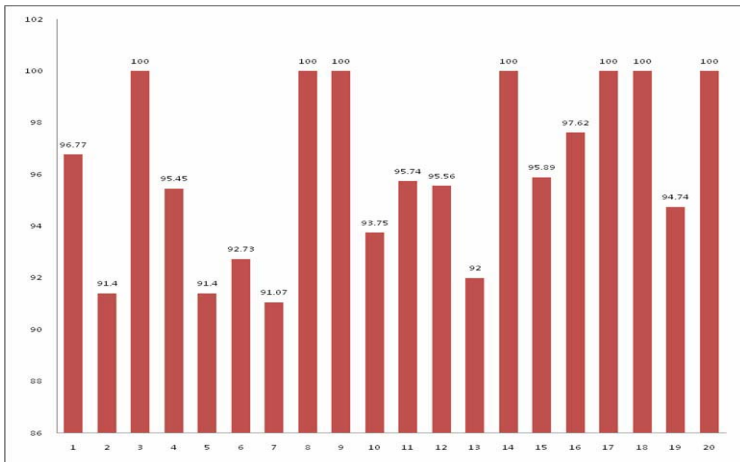


Fig. 5. Cluster vs. Efficiency Graph

The graphical representations of “clusters vs. efficiency” and “input data vs. errors” are shown in Figure 5 and Figure 6 respectively.

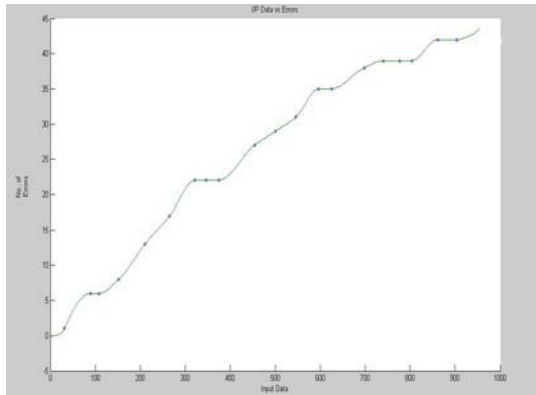


Fig. 6. Input Data vs. Errors Graph

6 Conclusion

A series of cluster based multilingual speaker identification experiments using artificial neural network have been conducted. The result shows that BPA can be used for multi language system. This research focuses on text dependent speaker recognition. The minimum performance of the system is 91.07% while the best performance is being reached upto 100%. Overall performance of the system is 95.354%. The desired goal of a system which can understand a text independent expression uttered by different speakers using various languages in different environments can be the further enhancement of this research.

This system is localized for large input data. Increasing the number of clusters would make the solution more localized. The new databases having more classes and more data per class need to be built for the optimal testing of this approach and testing on different databases need to be done in future. The generalization of the ANN would be lost to some extent. Hence we are able to train an ANN that was in other way not being trained, by some loss of generality. The effect of loss of generality with the increasing number of clusters may be studied in future.

References

1. Love, C., Kinsne, W.: A speech recognition system using a neural network model for vocal shaping, Department of Electrical and Computer Engineering, University of Manitoba Winnipeg, Manitoba, Canada R3T-2N2, 198 p. (March 1991)
2. Chougule, S., Rege, P.: Language independent speaker identification. Ieexplore, 364–368 (May 2006)

3. Mak, M.W., Allen, W.G., Sexton, G.G.: Speaker identification using radial basis functions. In: IEEE 3rd international conference, ieeexplore, University of Northumbria at Newcastle, UK, pp. 138–142 (1993)
4. Bum, D.: Experiments on neural net recognition of spoken and written text. IEEE Trans., on Acoust. Speech and Signal Proc. ASSP-36(7), 1162–1168 (1988)
5. Deiri, Z., Botros, N.: LPC-Based neural network for automatic speech recognition. In: Proc. IEEE Engineering in Medicine and Biology Soc., IEEE Cat. no. 90 CH2936-3, pp. 1429–1430 (1990)
6. Wassermann, P.P.: Neural Computing: Theory and Practice. VNR, New York (1989)
7. Lipmann, R.P.: Review of Neural Networks for Speech Recognition. Neural Computation 1, 1–38 (1989)
8. Zebdum, R.S., Guedes, K., Vellasco, M., Pacheco, M.A.: Short Term Load Forecasting Using Neural Nets. In: Sandoval, F., Mira, J. (eds.) IWANN 1995. LNCS, vol. 930. Springer, Heidelberg (1995)
9. Zurada, J.M.: Introduction to Wavelet Neural Systems. West Publishing Company (1992)
10. Falthauser, R., Ruske, G.: Robust speaker clustering in eigenspace. In: Inst. for Human-Machine-Communication, Technische Universitat Munchen, Munich, Germany. IEEE, Los Alamitos (2002)
11. Kala, R., Shukla, A., Tiwari, R.: Fuzzy Neuro Systems for Machine Learning for Large Data Sets. In: Proceedings of the IEEE International Advance Computing Conference, ieeexplore, Patiala, India, March 2009, pp. 223–227 (2009)
12. Kumar, S.C., Mohandas, V.P., Li, H.: Multilingual Speech Recognition: A Unified Approach. In: Proceedings of Interspeech 2005 (Eurospeech - 9th European Conference on Speech Communication and Technology), Lisboa (September 2005)
13. Kumar, S.C., Li, H.: Language identification System for Multilingual Speech Recognition Systems. In: Proceedings of the 9th International Conference Speech and Computer (SPECOM 2004), St. Petersburg, Russia (September 2004)
14. Sun, H., Ma, B., Li, H.: An Efficient Feature Selection Method for Speaker Recognition. In: International Symposium on Chinese Spoken Language Processing, China, December 16-19 (2008)
15. Ma, B., Guan, C., Li, H., Lee, C.-H.: Multilingual Speech Recognition with Language Identification. In: International Conference on Spoken Language Processing (ICSLP), September 16-20, Denver-Colorado (2002)
16. Ichihashi, H., Honda, K., Notsu, A., Miyamoto, E.: FCM classifier for high-dimensional data. In: IEEE International Conference on FUZZ-IEEE 2008 (IEEE World Congress on Computational Intelligence), June 2008, pp. 200–206 (2008)
17. Soong, F.K., Rosenberg, A., Rabiner, L., Juang, B.: A vector quantization approach to Speaker Recognition. In: Proc. ICASSP, April 1985, pp. 387–390 (1985)
18. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
19. Shukla, A., Tiwari, R.: A novel approach of speaker authentication by fusion of speech and image features using Artificial Neural Networks. Int. J. of Information and Communication Technology 1(2), 159–170 (2008)

Modeling Huntington's Disease Considering the Theory of Central Pattern Generators (CPG)

Masood Banaie^{1,*}, Yashar Sarbaz², Mohammad Pooyan³, Shahriar Gharibzadeh²,
and Farzad Towhidkhah⁴

¹ Laboratory of Biomedical Engineering, Faculty of Engineering, Shahed University, Iran
Banaie@gmail.com

Tel.: +98 935 21 28208; Fax: +98 21 5121 2021

² Neuromuscular Systems Laboratory, Biomedical Engineering Faculty, Amirkabir University
of Technology, Iran

³ Department of Biomedical Engineering, Faculty of Engineering, Shahed University, Iran

⁴ Biological Systems Modeling Laboratory, Biomedical Engineering Faculty,
Amirkabir University of Technology, Iran

Abstract. In this study, we present a novel model for Huntington's disease (HD) gait disorder. We consider the mathematical relations between blocks. The number of inputs and outputs of each block are designated due to the physiological findings. The connection types between blocks are modeled by gains. Inner structure of each block is modeled using a central pattern generator neural network. Our model is able to simulate the normal and HD strides time intervals and shows how diazepam is able to ameliorate the gait disorder; however, we believe that this treatment is somehow irrational. Using GABA blockers recovers the symptoms but it means omitting BG from motor control loop. Our model shows that increment of GABA aggravates the gait disorder. Our novel idea about BG treatment is to reduce glutamate. Experimental studies are needed for evaluating this novel treatment. This validation would implement a milestone in treatment of such a debilitating disease. It seems that synchronization of a number of neurons is the major disturbance in HD. The synchronization was modeled as CPG (Central Pattern Generator) structure. We supposed that the disorder will recover if the wrong synchronization of the neurons is diminished. Therefore, deep brain stimulation may be useful in this regard.

1 Introduction

Neuromuscular diseases are one of the most common problems that human being challenges with them, especially in old ages. These diseases often have no absolute treatment. Basal Ganglia (BG) have a main role in motor controlling. The degeneration of nerve cells in BG produces vast movement disorders. The most known movement disorders concerning BG are Parkinson's disease, Huntington's disease (HD) and Hemiballism. HD is not a wide-spread one, but has enormous movement disorders. In recent years, many researchers focus on HD to introduce an acceptable treatment for it [1, 2].

* Corresponding author.

The brain has a complex structure and this complexity is doubled in disease states. Therefore, introducing new models can be useful for better understanding of the disease. Many conceptual models have been designed to assess the disease behavior [3, 4]. However, a few computational models are present [5, 6]. Studying the disease in a mathematical approach can help us to analyze the system. Moreover, mathematical simulation can provide a better approach to study different states of the disease. It can be used to assess new proposed treatments to find their effectiveness prior to clinical trials. Verifying new treatments through clinical experiments is a time-consuming and costly process. The model can check reliability of different hypothesis.

There is little information about the mechanisms which cause HD symptoms. There are some hypotheses about the mechanisms of HD; but little models are available focusing on them in a mathematical way. In this study, we try to present a novel model for HD gait disorder, i.e. one of the main symptoms of HD. There are few ordinary mathematical models about HD disorder, more of which were introduced by our research group. In this novel research, we generate a model from a new viewpoint. Contrary to old methods for modeling the HD that focuses on detailed neuron behavior, the new model is introduced based on global behavior of system. The model certifies the predictions of previous models as well as modeling the healthy and disease states.

Physiological Background

Losing specific neurons in BG can result in movement disorders that are classified in HD symptoms. Some information about nerve cell loss and connection weight variations in BG is available in scientific texts. The structure of BG is known including BG parts, the kind of neurotransmitters between blocks and the interaction type of connections, i.e. excitatory or inhibitory(Ganong, 2003). This information is presented in figure1.

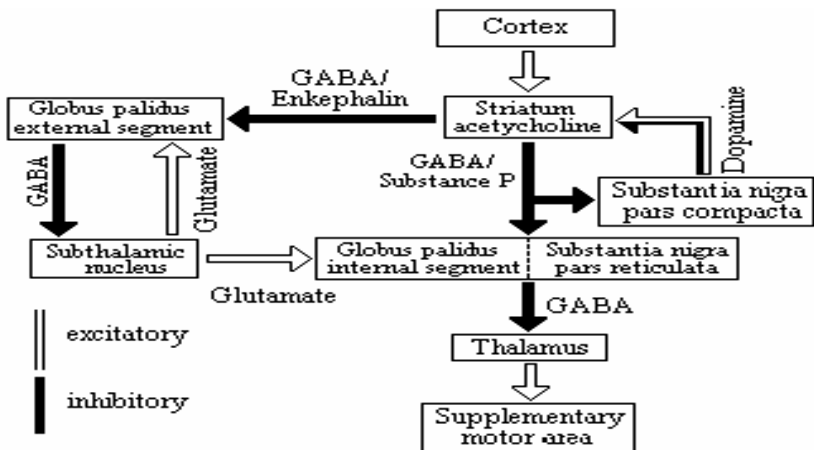


Fig. 1. A schematic diagram of the different neurotransmitters used in the connections of the basal ganglia

Moreover, the differences between connection weights of blocks in healthy and HD states are known. A schematic diagram of these variations is shown in figure 2 (Ganong, 2003).

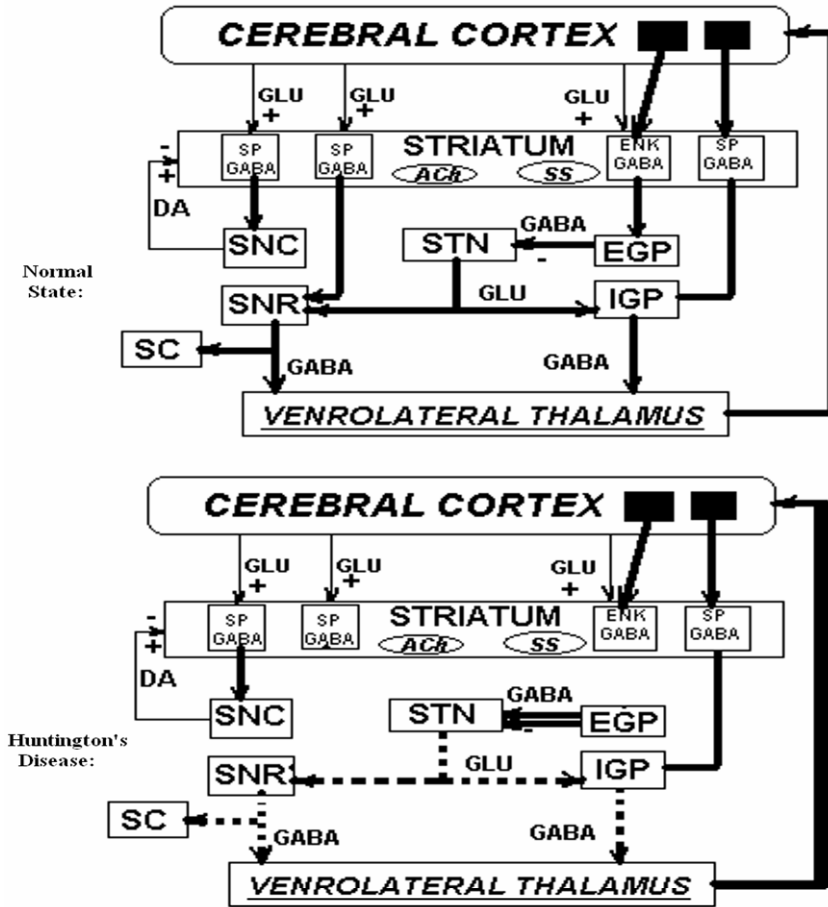


Fig. 2. Connection weights in the BG of normal and HD patients

Our information about the internal function of each block is limited. Each block contains numerous neurons working together. In previous works, we used properties of single neurons for modeling the blocks. However, we focus on the global behavior of system in the recent research. Various studies have showed that different parts of Central Nervous System (CNS) have different circuits that are technically called Central Pattern Generator (CPG). CPG is a set of neurons which produce a special pattern e.g. gait, respiratory system movements and other periodic movements of human body. Some of these CPG circuits are stereotypical ones (like respiratory system) and some others are formed by learning (like swimming). There are some hypotheses claiming that CPG circuits are established in some neuromuscular diseases producing semi-periodic movements. The best example for this hypothesis is Parkinsonian tremor. In HD, the gait disorder is supposed to be in such a way. Variation in gait signal of normal person appears to have a random-like behavior; but in the HD patients, a semi periodic signal is observed. This concept can be deduced

from figure 3. According to the introduced hypothesis, some oscillations of CNS are synchronized which produce the movement symptom. These oscillations are present in healthy state; but they weaken each other because of the balance of the CNS [8, 9, 10]. We suppose that in HD, a CPG is produced. Hence, it could be guessed that a CPG in the BG is organized.

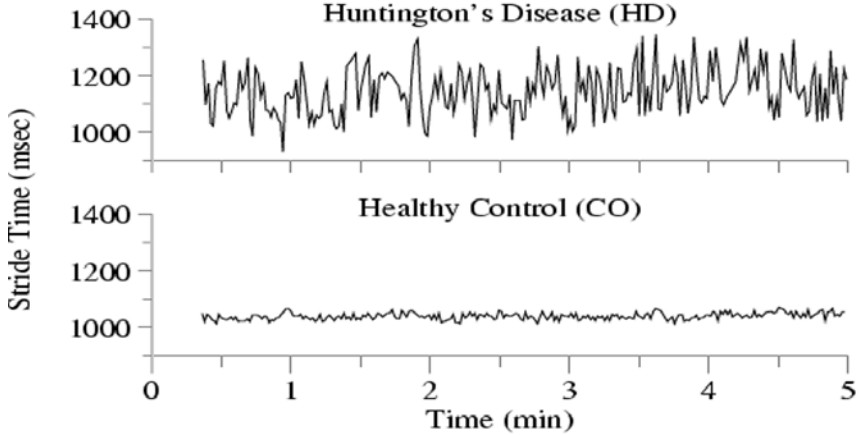


Fig. 3. Comparison of gait stride time

As mentioned above, each block of BG contains numerous neurons. Therefore, the best mathematical method for modeling them could be artificial neural networks. On the other hand, CPG is established in BG. Then, we can consider each block as a CPG neural network. Various neural networks are introduced as CPG. We use one of ordinary CPG models to consider the neurons' behavior. In addition to these considerations, we have noted that glutamate role in HD is in a state that can be called toxicity.

Mathematical Model

In this research, we introduce a mathematical model that simulates the gait disorder (stride interval signal) in HD state. The database is obtained from www.physionet.org [7]. In our modeling, we consider the relations between blocks. The number of inputs and outputs of each block was designated due to the physiological findings. The connection types between blocks are modeled by gains; i.e. excitatory and inhibitory connections are simulated using positive and negative gains, respectively.

Inner structure of each block is modeled using a CPG neural network. A schematic diagram of the CPG is shown in figure 4.

The CPG is established by two equivalent neurons, interaction of which produces the output pattern. The relation of CPG model is presented below:

$$\tau_1 \dot{x}_1 = -x_1 - \beta x_1 - w[x_2]^+ - \sum_{j=1}^n h_j [g_j]^+ + c \quad (1)$$

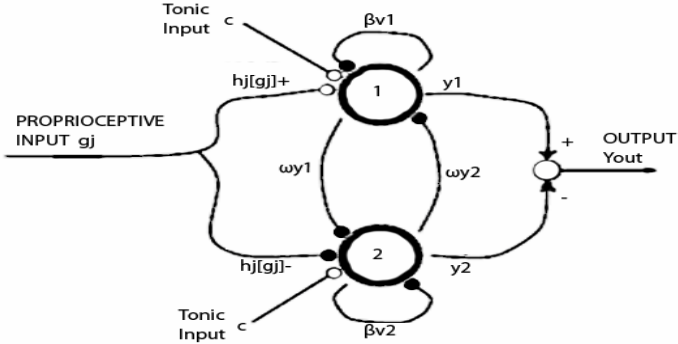


Fig. 4. Structure of a CPG unit

$$\tau_2 \dot{v}_1 = -v_1 + [x_1]^+ \quad (2)$$

$$\tau_1 \dot{x}_2 = -x_2 - \beta x_2 - w[x_1]^+ - \sum_{j=1}^n h_j [g_j]^- + c \quad (3)$$

$$\tau_2 \dot{v}_2 = -v_2 + [x_2]^+ \quad (4)$$

$$y_i = [x_i]^+ = \max(x_i, 0) \quad (5)$$

$$Y_{out} = y_1 - y_2 \quad (6)$$

We also consider a threshold block in the beginning of the CPG in each block to model the activation threshold of CPG neurons; because the CPG neurons need a minimum input to show the oscillatory behavior. The proposed model includes most of available physiological findings in scientific texts. Moreover, using Artificial Neural Network (ANN) has prepared a global view of performance of BG in HD state. Figure 5 shows a schematic diagram of the model structure considering the physiological findings.

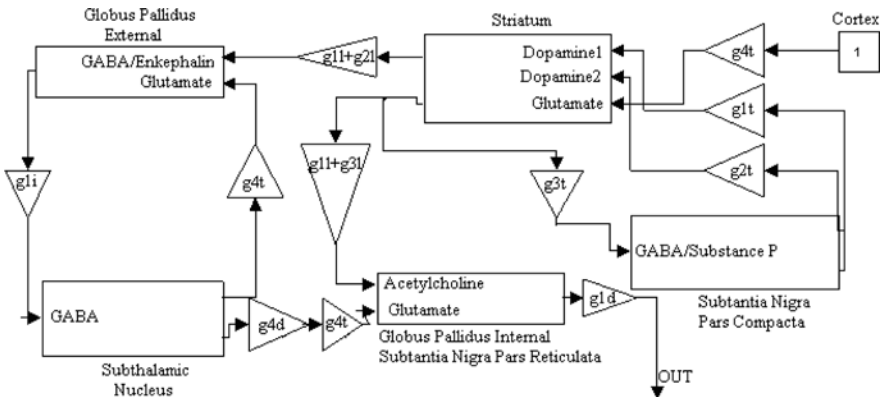


Fig. 5. Schematic diagram of BG model in HD state

The notations for nomenclature of the gains (g) on block connections are as follows: each connection name that ends to “l” is a connection which will be lost in HD state; each name that ends to “i” is a connection weight that increases and each name that ends to “d” is one which decreases in HD. All gains ending to “t” are designated to assess the HD treatment in the model and to research on the new treatments.

In the HD state, in addition to the changes discussed above, one of neurotransmitters in BG, glutamate, becomes toxic; i.e., the glutamate increases abnormally. In our model, we have considered this state with adding a noise with a positive mean to the glutamate signal. The variations of parameters due to physiological and clinical findings are presented in Table 1.

Table 1. Variation of parameters of the model in normal and disease state

Weight	<i>N</i>	<i>gli</i>	<i>gld</i>	<i>g4d</i>	<i>g3l</i>	<i>g2l</i>	<i>g1l</i>
Normal	0	1	1	1	0.5	0.5	0.5
Disease	3	2	0.5	0.5	0	0	0

Results

In our modeling procedure, we have ignored weak variations in stride interval of healthy persons. Hence, the model produces zero output in healthy state; because, we have no abnormal synchronization in healthy state. The response of the model for healthy and the disease (HD) state is shown in figure 6.

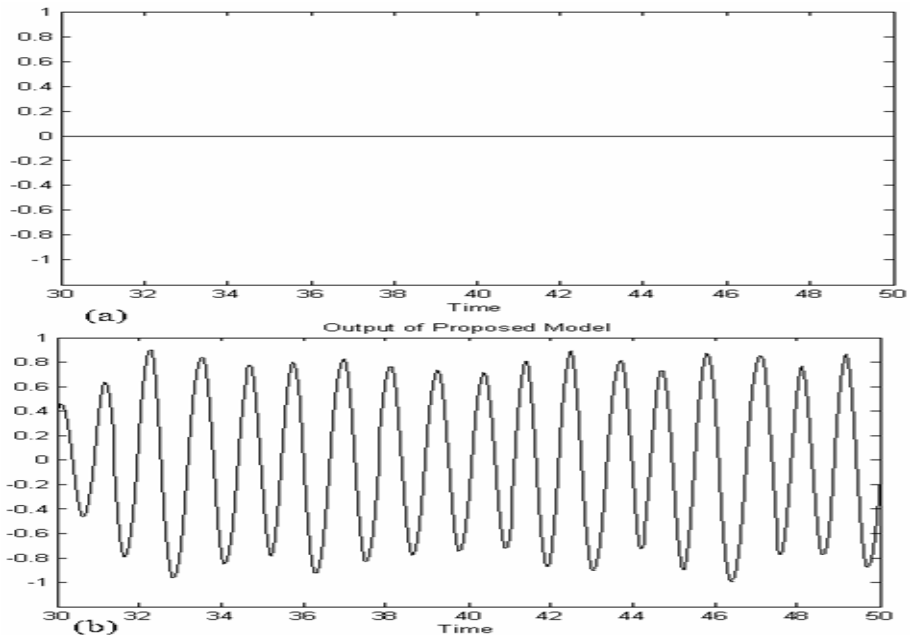


Fig. 6. The model response of model for healthy (a), and disease (b) states

A sample of recorded clinical data of HD is presented in figure 7.

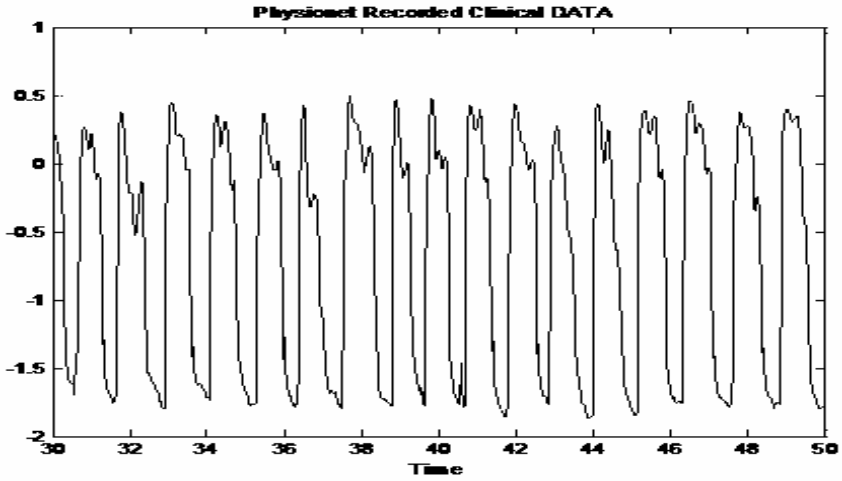


Fig. 7. A sample of recorded clinical data in HD

To compare recorded data and simulation results, we used power spectra criteria. Figure 8 shows the power spectra of model output and clinical data.

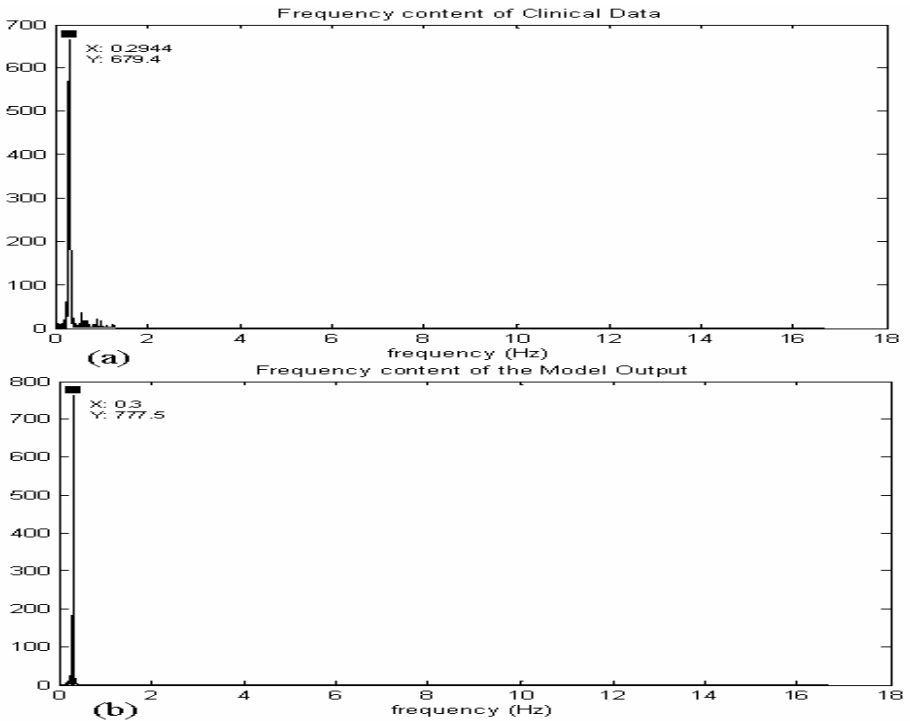


Fig. 8. Comparison of power spectra between simulated (a), and clinical (b) data

The model response to diazepam, a drug that is supposed to increase the threshold of neurons, was simulated. It was obvious in our results that diazepam ceased the response completely; i.e. the disease symptoms are eliminated and the change in stride time intervals becomes zero.

The model response to using GABA blocking agents is shown in figure 9.

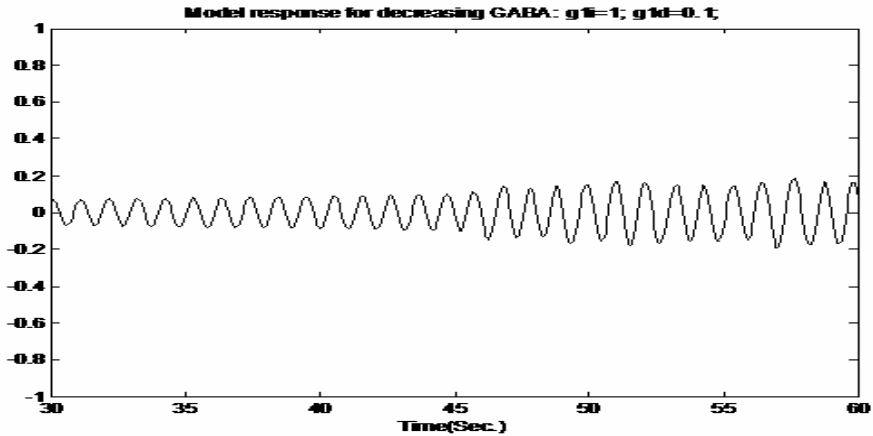


Fig. 9. The model response to GABA blocking agents

We simulated the model response to decreased glutamate level. Our results showed that stride time interval changes becomes zero.

Discussion

Computational modeling of brain performance has a great task in elucidating medical knowledge and has expanded our judgment about the illness states. Effectiveness of computational models is manifested by two factors: The complexity of brain structure itself, and the greater complexity of pathological conditions. Meantime, certain neurological disorders such as BG diseases have a particular importance and have gained attention of many researchers.

However, because of the complexity of HD, there are no many researches about it. Due to the vast range of movement disorders, there are small ranges of successful studies about HD treatment. In addition, there are a few computational simulations prepared by other researchers about the main symptoms of HD. Hence, presenting a computational model will be costly.

As it is shown in the results section, our model is able to simulate the normal and HD stride time intervals (Figure 6). Comparing the model response and clinical data through power spectra criteria shows that our model is valid (Figure 8). Moreover, our model shows how the present treatment, i.e. diazepam, is able to ameliorate the gait disorder.

We believe that the common present treatment, i.e. diazepam, is somehow irrational. Using diazepam, with raising the threshold of BG neurons and inhibiting some BG cells, in effect omits the BG from the neural circuit of motor control. Hence, using diazepam is not a reasonable route.

In this research we assessed the effects of changing some neurotransmitter levels as parameters of our model in order to propose new treatments.

The output of BG is GABA; then, using GABA blockers means omitting BG from motor control loop. In this situation, despite the reduction of HD movement disorder, the patient would experience the side effects of the drug that would be the result of removing the role of BG on movement control. Therefore, it is unjustifiable to use this route for treatment.

Moreover, we evaluated our model response to increment of GABA, alone. We observed that the gait disorder was aggravated. This is also interpretable in the light of our idea; since in this situation, the internal disturbance of BG is augmented.

Our novel idea about BG treatment is to reduce glutamate. It is surprising that despite increment of glutamate in BG of HD patients, rare studies have focused on decreasing glutamate as a route of treatment. Our model was able to show the efficiency of this treatment on HD disturbances.

We propose that experimental studies should be designed in which this novel method of treatment will be evaluated. This validation would implement a milestone in treatment of such a debilitating disease at Huntington.

Up to now, we have assessed the behavior and attributes of the neurotransmitters; but, we can focus on the disease systematically. Our main theory of the disease is synchronization of a number of neurons in a wrong way that will lead to forming the disorder. The synchronization was modeled as CPG structure. The consequences of our modeling have strengthened the validity of the hypothesis. It is supposed that the disorder will recover if the wrong synchronization of the neurons is diminished. In other words, if we remove the abnormally formed CPGs that are resulted from HD, the symptoms of the disease would be ameliorated. It sounds that one of the main mechanisms of removing the abnormal synchronization of neurons is Deep Brain Stimulation (DBS). The method is employed to treat the Parkinson's disease (PD) and showed good results. Some theories about formation of CPG in PD are available [8, 9, 10]. So, we propose that DBS may be useful to treat the HD.

References

1. Kandel, E.R., Schwartz, J.H., Jessell, T.M.: Principles of Neural Science, 4th edn. McGraw-Hill, New York (2000)
2. Ganong, W.F.: Review of medical physiology, 21st edn. McGraw-Hill, New York (2003)
3. Beuter, A., Titcombe, M.S.: Modulation of tremor amplitude during deep brain stimulation at different frequencies. *Brain Cognition* 53(2), 190–192 (2003)
4. Menalled, L.B., Marie-Françoise, C.: Mouse models of Huntington's disease. *Trends Pharmacol. Sci.* 23(1) (2002)
5. Haeri, M., Sarbaz, Y., Gharibzadeh, S.: Modeling the Parkinson's tremor and its treatments. *Journal of Theoretical Biology* 236(3), 311–322 (2005)
6. Sarbaz, Y., Banae, M., Gharibzadeh, S.: A computational model for the Huntington disease. *Medical Hypothesis* 68, 1154–1158 (2007)
7. <http://www.physionet.org>
8. Tass, P.A.: *Phase Resetting in Medicine and Biology: Stochastic Modeling and Data Analysis*. Springer, Berlin (1999)
9. Tass, P.A.: Stochastic phase resetting: a theory for deep brain stimulation. *Prog. Theor. Phys. Suppl.* 139, 301–313 (2000)
10. Williamson, M.M.: Neural Control of Rhythmic Arm Movements. *J. Neural Network* (1998)

Prophetia: Artificial Intelligence for TravelBox[®] Technology

R. Chulaka Gunasekara¹, B.B. Akila Geethal¹, Mafaz Hassan¹,
C.D. Tharindu Mathew¹, A. Shehan Perera¹, Harsha Subasinghe², Yohan Welikala²,
and Lasantha Wimalasena²

¹ Department of Computer Science and Engineering,
University of Moratuwa, Sri Lanka

{chulaka.gunasekara, akila.geethal, mafaz.hassan,
tharindu.mathew, shehan}@uom.lk

² Codegen International (Pvt.) Ltd, Sri Lanka
{harsha, yohan, lasanthaw}@codegen.net

Abstract. In today's fiercely competitive and dynamic market scenario, business enterprises are facing many problems due to increasing complexity of the decision making process. Besides, the amount of data to be analyzed has increased substantially. This has resulted in Artificial Intelligence stepping into decision making to make better business decisions, reduce latency and enhance revenue opportunities. Prophetia is a research project carried out to integrate Artificial Intelligence capabilities into TravelBox[®] technology – a range of solutions developed by Codegen International for Travel Industry. This research paper discusses three areas that were researched for the above purpose. These are, Probability Prediction – the use of Neural Networks for calculating the selling probability of a particular vacation package, Package Recognition – the use of Self Organizing Maps for recognizing patterns in past vacation package records, and Customer Interest Prediction – the use of association rule mining for determining the influence of customer characteristics on the vacation destination.

Keywords: Neural Networks, Self Organizing Maps, Data Mining.

1 Introduction

Prophetia is a research project carried out to integrate Artificial Intelligence capabilities into TravelBox[®] technology – a range of solutions developed by CodeGen.IT for tour operators. The ultimate goal of this research is to add, prediction and pattern recognition capabilities to the armament of features that TravelBox[®] provides to its customers.

The volume of data that is available for today's travel industry to base their decisions upon is overwhelming. In addition to this, the variability of the data available (for example, unknown patterns/relationships in sales data, customer buying habits, and so on) makes the analysis of this data an even more complex task.

Artificial Intelligence has seen a revival in its adaptation for commercial use in many industries such as the finance industry as a result of the advent of web-enabled infrastructure and giant strides made by the AI development community. AI has been widely adopted in such areas of risk management, compliance, and securities trading and monitoring, with an extension into customer relationship management (CRM). The use of AI technologies brings more complex data-analysis features to existing applications [1]. This has resulted in CodeGen.IT also exploring the use of AI technologies to improve the functionalities it can provide to its customers.

Prophetia concentrates on using AI technologies to solve the following three problems related with TravelBox[®].

1.1 Probability Prediction

The problem to be addressed was that a probability that indicates the selling potential of a certain vacation package should be predicted once its parameters are known. A vacation package consists of fifteen parameters. These parameters are; Booking Date, Departure Date, Pax Count, Lead Age, Total Cost, Total Price, Nights, Product Group, Flight Departure City, Flight Destination, City of Accommodation, Flight Cabin Class, Airline, Hotel Code and Room Type. An example of the available data is shown below:

Table 1. A Sample of the data available for Probability Prediction

TOTAL_COST (\$)	BOOKING_DATE	HOTEL_NAME
2045.81	6/12/2008 0:00 A8P8	Hilton Hotel
565.71	8/12/2008 0:00 A8P8	Queen's Hotel
1225.90	9/12/2008 0:00 A8P8	Plaza Hotel

This research was carried out for 10,000 records, keeping room for future developments targeting a much larger data collection.

The expectation of the project was to come up with a mechanism that calculates the predicted probability values in an efficient and accurate way. There was a possibility of coming up with a non linear function that calculates the selling probability of vacation packages when the input parameters are given. But coming up with a regression function for the purpose is really a time consuming task due to the higher number of input parameters and variance of input data [2]. Neural Networks are general and data driven. This means that once the data is fed, the network learns from the data itself without human intervention and the variability and volume of the data does not affect the ability of the neural network to solve the problem [3]. Based on this, Neural Network approach was selected as the suitable method to solve this problem.

1.2 Package Recognition

The second requirement of Prophetia is to recognize “hot spots” or patterns in transaction records of sold vacation packages and recommend vacation package combinations from these identified patterns.

The pattern identification problem was broken down into a typical problem of data mining, specifically into data clustering. For the purpose of data mining, the many techniques that exist were explored through available literature. There was the choice of using classical methods for this purpose versus modern AI related methods.

The data set was the same as in the data used in Section 1.1 of this paper (15 variables, 10000 records). This variability of the data set makes the use of classical methods useless in this process [4]. The AI related techniques were considered, and the Self-Organizing Map was selected, which has the following advantages when clustering data [5] [6]:

Data mining typically deals with high-dimensional data. A record in database typically consists of a large number of items. The data do not have regular multivariate distribution and thus the traditional statistical methods have their limitations and they are not effective. SOMs work with high-dimensional data efficiently.

Kohonen's self-organizing maps provide means for visualization of multivariate data, because two clusters of similar members activate output neurons with small distance in the output layer. In other words, neurons that share a topological resemblance will be sensitive to inputs that are similar. This property has no other algorithm of cluster analysis.

1.3 Customer Interest Prediction

The 3rd and final requirement of Propheta deals with predicting customer's future interests according to their past data. This is done by producing a value for the influence factor of a customer's characteristics on the holiday destination. Anything above the threshold influence will be recommended to the customer as a holiday destination in his holiday destination search.

With prediction coming into play, first the research team was very much inclined to use a neural network oriented approach, either supervised or unsupervised, but given the simple nature of the requirement, associative rule mining seemed to be the most suited AI technique considering its intended application [7]. The aim of using association rule mining in this project is to find interesting and useful patterns in the data given, and come up with the association rules processing those data.

The data consists of 10,000 records with fifteen parameters. These are, Passenger Age, Marital Status, Sex, Booking Status, Option Status, Booking Price, Booking Departure Gap, Departure Month, Booking Month, Travel Together Pax, Holiday Destination, Holiday Type, Holiday Duration, Holiday_Has_Flight, Holiday_Has_Transfer, Holiday_Has_Excursion, Holiday_Has_Tour and 7_Or_Less_Nights. Shown below is an extract of the data:

Table 2. Sample of the available data for Customer Interest Prediction

PASS_AGE	MARITAL_NAME	SEX
85	Married	M
76	Unknown	F
23	Single	M

For the purpose of rule mining, algorithms such as Apriori, Predictive Apriori and Tertius were considered [8]. The decision of going through Predictive Apriori was taken after going through relevant literature and discovering its advantage in using for large item sets. [9]

This paper will discuss about the research carried out by the research team and the results obtained by the research. Section 2 of this paper will explain the underlying theory behind the AI techniques used in this project. Section 3 will look at the implementation details followed by the research team in order to come up with the expected results. The results obtained after the implementation will be discussed in the Section 4. Finally Section 5 will give the conclusion to this paper.

2 Theory

The main AI techniques that are used in this project are Neural Networks, Self Organizing maps and Association Rule Mining. The basic theoretical concepts of these three techniques will be explained in this section.

2.1 Neural Networks

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well. ANNs also have layers of neurons with synapses connecting them. Fig. 1 illustrates the different layers of an ANN and how the synaptic connections are used to interconnect those layers.

Generally there are two methods of learning used in neural networks. These are called supervised learning and unsupervised learning. Those two learning methods are made use in different scenarios where neural networks are used to solve problems. Both learning methods have their own specific application vicinities.

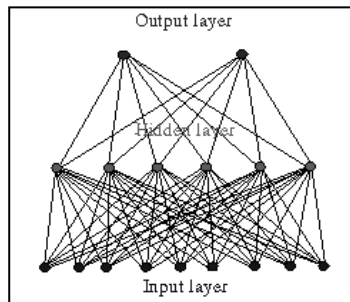


Fig. 1. Interconnections between different layers of an ANN. *Input layer*, *Hidden layer* and the *Output layer* are shown here.

- Supervised learning - which incorporates an external teacher, so that each output unit is told what its desired response to input signals ought to be an important issue concerning supervised learning is the problem of error convergence, i.e. the minimization of error between the desired and computed unit values.
- Unsupervised learning - uses no external teacher and is based upon only local information. It is also referred to as self-organization, in the sense that it self-organizes data presented to the network and detects their emergent collective properties. [10],[11]

2.2 Self Organizing Maps

A self-organizing map (SOM) is a type of artificial neural network that is trained using unsupervised learning to produce a low-dimensional (typically two dimensional) representation of the input space of the training samples. This makes SOM useful for visualizing low-dimensional views of high-dimensional data. Fig. 2 illustrates neuron structure and connections in a Self Organizing Map. The model was first described as an artificial neural network by the Finnish professor Teuvo Kohonen, and is sometimes called a Kohonen Feature map [12].

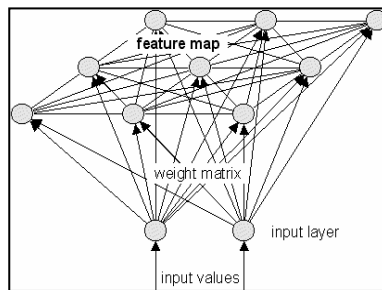


Fig. 2. Neurons and interconnections in a Self Organizing Map. The *weight matrix* which connects the *input layer* and *self organizing feature map* is shown here.

2.3 Association Rule Mining

Association rule mining is to find out association rules that satisfy the predefined minimum support and confidence from a given database. The problem is usually decomposed into two sub problems. One is to find those item sets whose occurrences exceed a predefined threshold in the database; those item sets are called frequent or large item sets. The second problem is to generate association rules from those large items sets with the constraints of minimal confidence [7].

3 Implementation

The following sections discuss the algorithms and implementation details of each of the three problems:

3.1 Probability Prediction

To train the supervised neural network of the Probability Calculation module of the system, the selling probability value of each and every data item had to be calculated first. A simple approach was taken to calculate the probability. The following formula was used:

$$\text{Probability} = \frac{\text{Number of occurrence of a certain package}}{\text{Total Number of records}} \quad (1)$$

Therefore, if a specific parameter combination occurred six times, the probability would be (Total No. of records = 10000): $6 / 10000 = 0.0006$. With this approach, every record would have an associated probability.

A feed forward artificial neural network was trained using the input parameter values as the input values and the derived probability value as the output value. Once the network was trained, when a particular parameter combination is provided, the neural network provided the probability value as the output. Joone (Java Object Oriented Neural Engine) was used to implement the Neural Networks. It is a framework where a neural network can be easily constructed using basic components associated with ANNs, such as synapses and layers [13].

The input parameters of our data set were comprised of several data types like 'Date' values, 'String' values and Numerical values. So, before inputting the data in to the supervised neural network, we had to convert all the data types into numerical values [14]. Different algorithms were used for different data types in order to convert them into numerical values. In dealing with 'Date' data type, first, the day of the year of each and every 'Date' value was calculated and then number of occurrences of that day of the year value in the given data set was taken as the numerical value for the particular 'Date' value. In that manner the highest occurred day of the year in a certain input parameter was given the highest weight compared to other days of the year. For 'String' data type, the number of occurrences of a certain 'String' value was taken as the respective numerical value of the 'String' value. Since, the number of occurrences of a specific 'String' value of data, is a variable with time, the normalized numeric value corresponding to that 'String' value is also a variable with time. So, this causes the selling probability value for same combination of input values a variable with time. But this is acceptable, since the selling probability of a specific vacation package can actually change over time. So, the number of occurrences of data was taken as the basis for calculating the normalized values for 'String' data.

Then, the data set which was made numerical, had to be normalized before inputting to the neural network since the data of almost all input parameters, had a huge range and deviated a lot from the respective mean values. So, all the numerical values were normalized into an acceptable range before inputting into the neural network. Equation (2), which is given below was used to normalize the input value D in to normalized input value I . In (2), I_{max} is the highest value to which the input values were normalized into and I_{min} is the lowest value to which input values were normalized into. In our implementation I_{max} was taken as +1 and I_{min} was taken as -1. So, all the normalized input values were in the range of -1 to +1. Then, highest input value and the lowest input value of each input parameter were calculated. In (2), D_{max} was taken as the highest input value of a particular input parameter and D_{min} was taken as the lowest input value of the same input parameter.

$$I = I_{\min} + (I_{\max} - I_{\min}) * ((D - D_{\min}) / (D_{\max} - D_{\min})) \quad (2)$$

This method of normalization will scale input data into the appropriate range but will not increase its uniformity [15]. The normalized data for the example shown in Table 1 is given in the Table 3 below.

Table 3. Sample of the Normalized data for the Probability Calculation

TOTAL_COST	BOOKING_DATE	HOTEL_NAME
-0.6920073	-0.66025641	-0.35135
-0.9240703	-0.34562982	-0.86486
-0.8205600	0.25985201	-0.67568

After coming up with the input necessary for the Neural Network, the research was extended to find out the most appropriate configuration for the Neural Network to be used in the application. There were some guidelines followed in deciding the number of hidden layers and the number of neurons in the hidden layers [16]:

- The number of hidden neurons should be in the range between the size of the input layer and the size of the output layer.
- The number of hidden neurons should be around 2/3 of the input layer size, plus the size of the output layer.
- The number of hidden neurons should be less than twice the input layer size.

While following the above guidelines, the neural network was trained changing the number of hidden layers and number of neurons in the hidden layers. The Root Mean Squared Error (RMSE) value for each configuration of the neural network was recorded and the neural network which produced the lowest RMSE value was saved. The RMSE value for each configuration was calculated by (3), which is given below. In (3), $f(x_i)$ is the value predicted by the neural network, y_i is the actual value which is in the data set and n is the number of records in the data set.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n ((f(x_i) - y_i))^2}{n}} \quad (3)$$

When input parameters of a certain vacation package was given to the system to find the selling probability, the neural network which was generated earlier is retrieved and the selling probability is calculated using that neural network.

3.2 Package Recognition

The SOM is an unsupervised neural network that would identify and group the data being fed into clusters without the need of human intervention. Therefore, appropriate normalizing of data is paramount for the success of the results that will be obtained.

Since the same data set was used here as the probability prediction module, the same algorithm which was described in Section 3.1, is used in normalizing the data. But since we are concentrating on clustering, assigning numeric values for other data types was done without concerning about their number of occurrences.

Joone was used to implement the SOM with the use of its Kohonen Synapse. A 10x10 map was used to cluster the data. This unsupervised network is trained and then saved to preserve consistency. Then, whenever there was a need to find out new vacation packages, that saved neural network is loaded and clustering of the data is done based on that. When clustering is done, the results are displayed graphically using Java 2D. This Java 2D map displays the number of vacation packages found in the data set and the number of records in each of those vacation packages. The size of the neurons is determined by the number of records on each.

After getting the dataset clustered the next task was to analyze the data and give away probable new package types. As mentioned above, some of the clusters contain records with same product group, which implies that cluster is an existing vacation package type. So those clusters can be neglected for analyzing. So to analyze the other clusters, first, the largest clusters with different product groups are chosen. Although the clustering is done based on normalized data, for analyzing purposes actual records of those normalized data had to be fetched. This was done by extracting the record number of the particular normalized record and retrieving the actual data values corresponding to that record number from the data set. Then in each cluster, the most occurred value for each parameter is determined. Then output objects are created with these values.

So, to predict the vacation packages all the combinations from these highest values can be considered. To filter out the predicted vacation packages which are not practical in the real world, a rule set which is used in the current process of creating the vacation packages was used.

3.3 Customer Interest Prediction

The Predictive Apriori algorithm was followed here to determine association rules. Apriori is designed to operate on databases containing transactions. It uses a breadth-first search strategy to counting the support of itemsets and uses a candidate generation function which exploits the downward closure property of support. [17]

Association Rules were generated using the Destination of a travel package as the target parameter (ex: Colombo, Beijing) from the Predictive Apriori algorithm. The strength or the reliability of an association rule is called the confidence value. The confidence of an association rule is the percentage value that shows how frequently the rule head (right hand side of the association rule) occurs among all the groups containing the rule body (left hand side of the association rule). The confidence value indicates how reliable this rule is. Rules that exceed a threshold confidence value of 0.5 are selected and this value is presented as the influence value of the holiday destination. The WEKA java library is used to carry out the implementation.

Some rules generated in this manner are shown below:

- i. Booking_Month = Jan & Hol_Has_Exec = No & Hol_Has_Tour = No ==> Holiday_Destination =Beijing (Confidence Value: 0.95135)
- ii. Passenger_Age = 60-80 & Sex = Female & Departure_Month = Sep ==> Holiday_Destination =Tokyo (Confidence Value: 0.93782)
- iii. Departure_Month =Sep & Booking_Month =Jan & Hol_Has_Tour = no ==> Holiday_Destination =Moscow (Confidence Value: 0.89752)

Using these rules, a set of preferred destinations for each customer can be presented in the following manner:

- When a customer requests for a package search, booking details are entered and the respective customer details are retrieved from the database.
- Using these details, the rules are searched for matching parameter values.
- Respective destinations are retrieved from the rules that match, sorted according to the confidence value and presented.

4 Results

The following sections discuss the results and analysis of each of the three problems:

4.1 Probability Calculation

The first test was carried out to find out the suitable configuration for the network. From this test following parameters and combinations had to be determined.

- The number of Layers for the network
- Number of neurons in each layer
- Number of training cycles (epochs)

To identify the number of layers and number of neurons, the supervised network was trained with several configurations, while taking RMSE value as the deciding factor.



Fig. 3. The training error of the single layer ANN plotted against the number of neurons used in the hidden layer

Fig. 3 illustrates the RMSE values obtained when the network was tested using one hidden layer and changing the number of neurons from 1 to 17. Then the research was extended by using two hidden layer neural networks. Here, the research team recorded RMSE values changing the number of neurons in layer 1 from 1 to 17, while for each combination in layer 1 the changing number of neurons in layer 2 from 1 to 11.

The combinations that returned the lowest values of the training error were then subjected to validation. This was done by using 80% of the data for training and using the left 20% for validating results. This is done to determine the number of effective training cycles to be used. Fig. 4 illustrates the training error obtained for different

number of cycles that the network was trained for. The number of cycles which produces the minimum validation error was taken as the number of training cycles for the network.

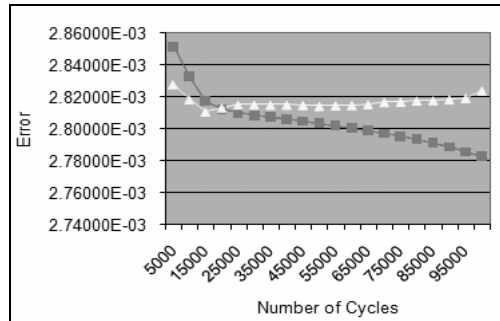


Fig. 4. Validation graph for the supervised Neural Network. Training error and validation error are plotted against the number of cycles trained.

Table 4 illustrates the optimum results obtained by testing:

Table 4. Optimum results for Probability Prediction Neural Network

FACT	RESULT
No. of hidden layers	1
No. of neurons in the hidden layer	11
No. of training cycles	15000

The output of the application, which is the selling probability of the vacation package, is generated from the neural network by feeding in the input parameters of the vacation package and running a single cycle of the neural network. This final method was developed into a web service so that other modules in TravelBox® can access this module.

An example of the functionality is given below:

INPUT - Package Combination:

- BOOKING_DATE - 08/12/2008 0:00 A8P8
- DEPARTURE_DATE - 11/23/2008 0:00 A11P11
- PAX_COUNT - 2
- LEAD_AGE - 72
- TOTAL_COST - 2045.81
- TOTAL_PRICE - 2742
- NIGHTS - 11
- PRODUCT_GROUP - 09
- FLIGHT_DEP - LHR
- FLIGHT_DES - CAI
- ACCOM_CITY - CAI
- FLIGHT_CABIN_CLASS - ECONOMY
- AIRLINE - BA
- HOTEL_CODE - CAL21
- ROOM_TYPE - ECONOMY

OUTPUT – Selling Probability

PROBABILITY

- 0.013983

The RMSE value which corresponds to the optimum result is 0.00205. The accuracy of the result improves as the RMSE value approaches 0 [18]. Since the obtained RMSE value is really close to 0, it can be assumed that the predicted probability values are really accurate.

4.2 Package Recognition

Fig.5 illustrates the clusters generated by the SOM used in Package Recognition module implemented using Joone. The number of clusters that were formed was 47 clusters. These were close to the expected number of results as the package types that exist are 45. The other observation was that the number of clusters that were formed after training the network each time was a constant value.

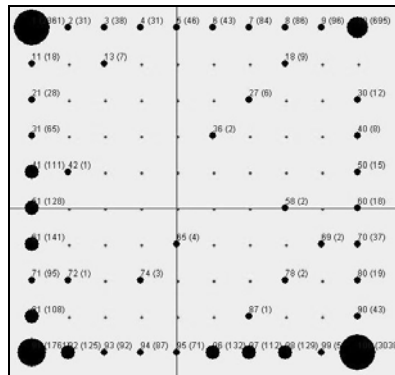


Fig. 5. Data clusters generated by Joone based Self Organizing Map. The thickness of a cluster is proportional to the number of records in the cluster.

A second option was also tried which is still in its research phase known as GSOM (Growing SOM) which is a dynamic SOM which grows its neurons according to the data that is being clustered. The tool was modified according to the requirement of Prophetia and used for data clustering [19]. Fig.6 illustrates the clusters generated by GSOM based clustering.

There were over 100 clusters formed that made it a difficult task to analyze and get a meaningful result. The large number of clusters that were formed caused a problem in analyzing and presenting these as feasible results. The project team experimented in trying to leverage the GSOM to reduce the number of clusters by adjusting the values of GSOM parameters such as Spread Factor, Learning Rate, and Weight Update Neighborhood. Still the number of clusters that were formed was above 100 clusters. Another problem that occurred with the GSOM was that every time it was

trained different numbers of clusters were formed with different compositions. These results lead to abandoning the GSOM approach for this requirement.

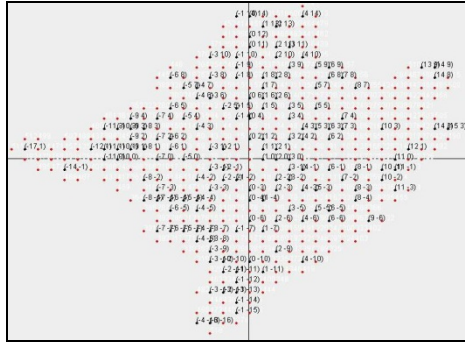


Fig. 6. Data clusters generated by GSOM based clustering. This shows the huge number of clusters generated by this approach.

Therefore, the results generated through the SOM were analyzed and the data extracted by the algorithm mentioned in Section 3.2. Finally those newly identified Package types are displayed in a tabular format where it can be used for decision making. Fig. 7 shows the newly identified vacation packages presented in a tabular format.

Departure Date	Total Cost	Flight	Departure	Flight Destinat...	Accom City	Airline	Hotel Code	Room Type
07/10/2008	1,394.47	LGW	NER	HRK	FCA	G23	N2SVB	
25/09/2008	1,756.00	EMA	ALC	DEC	WW	111	N2B	
13/09/2009	1,094.76	MAN	GRV	BLV	MON	IBD	N1SVB	
19/04/2009	630.42	LHR	POP	PETT	BA	JLC	N1LEB	
15/05/2010	1,922.75	MAN	ZRH	GASC	LX	ORB	ATWV	
06/01/2009	963.14	MAN	AVT	AVT	TCX	OHF	N2SVB	
19/01/2009	1,181.31	MAN	MLA	MLA	WV	CLM	N2SVB	
21/09/2010	1,037.98	LHR	MLJ	WANG	LH	OR7	N2	
22/12/2008	1,836.54	LHR	ATH	ACQ	OA	GA0	N1B	
12/02/2009	1,108.90	HRK	PRG	PURT	ZR	CRB	N2	
10/09/2008	520.40	HRK	PMI	CALD	WW	62B	N2B	
22/09/2009	922.03	LGW	NER	HRK	FCA	G23	N2SVB	
19/10/2008	584.64	LTN	AGP	NJR	IB	N13	N3PSB	
24/02/2009	344.86	MAN	AGP	NJR	WW	N13	N3PSB	
04/11/2008	1,225.90	LGW	PMI	PDY	BA	M3B	N2B	
08/07/2009	937.03	ALA	ACE	ACE	TCX	ISC	N2SVB	
23/04/2010	2,211.45	LHR	DRH	GASC	LX	ORB	ATWV	
06/10/2008	684.37	EMA	PMI	CALD	WW	62B	N2B	
07/09/2010	4,414.68	MAN	MLJ	MUTT	LH	OE3	A1SO	
14/11/2008	1,608.32	BRN	FRS	TCE	TCX	M14	N2SVB	
17/02/2009	2,462.71	LGW	HRG	HRG	TCX	FAE	N4SVB	
28/05/2009	1,672.47	LGW	BRI	ALBE	BA	33K	N1	
08/04/2009	2,643.63	EXT	PFO	PFO	FCA	ABL	N2SVB	
27/09/2008	448.73	LGW	ORO	ORO	TP	V99	N2	
18/02/2009	1,716.16	LGW	RMP	CAI	TOM	NEW RADESON	N1GNB	

Fig. 7. Newly identified Vacation packages using the SOM displayed in a tabular form

4.3 Customer Interest Prediction

When customers enter their details to find out the holiday destinations which are most appropriate to them, the system uses the association rules which are generated using the Predictive Apriori algorithm to present them with the most appropriate holiday destinations. Without the loss of generality, the list of preferred destinations shown below for a customer that has the following parameters:

SEX	-	M
MARITAL_STATUS	-	MARRIED
AGE	-	34
BOOKING_DATE	-	21/05/2009
DEPARTURE_DATE	-	24/09/2009
FLIGHT	-	NO
TOUR	-	NO
EXCURTION	-	NO
7_OR_LESS_DATES	-	YES

List of Destinations

Destination	Confidence Value
Beijing	0.8751
London	0.7400
Sydney	0.7397

5 Conclusion

The effort of integrating AI techniques into TravelBox® Technology has been a success with promising results. With the use of supervised neural networks, Self organized maps and association rule mining, Prophetia has achieved results that would have been hardly achieved with classical statistical methods. Further improvements should be made to these methods to further improve results before adopting it in commercial use but the steps that have been taken in this project in integrating AI in TravelBox® Technology will pave the way for future innovations in the range of products.

6 Confidentiality

The Contents of this paper represent intellectual effort from Codegen International (Pvt.) Ltd, and are confidential and bound by copyright laws. No part or ideas in this paper shall be reproduced or used without the written consent from Codegen International (Pvt.) Ltd.

References

1. Xindong, W.: Data Mining: Artificial Intelligence in Data Analysis. In: Proceedings. IEEE/WIC/ACM International Conference on Web Intelligence, WI 2004 (September 2004)
2. White, H.: Economic Prediction Using Neural Networks: The Case of IBM Daily Stock Returns. In: Proceedings of the Second Annual IEEE Conference on Neural Networks, II, pp. 451–458 (1988)

3. Delgado, F.J., Santín, D., Valiño, A.: The measurement of technical efficiency: a neural network approach. *Applied Economics* 36(6), 627–635 (2004)
4. Kumar, V.R., Mani, N.: The application of artificial intelligence techniques for intelligent control of dynamical physical systems. *International Journal of Adaptive Control and Signal Processing* 8(4), 379–392 (2007)
5. Zhang, X., Li, Y.: Self-organizing map as a new method for clustering and data analysis. In: *Proceedings of 1993 International Joint Conference on Neural Networks, IJCNN 1993-Nagoya*, vol. 3, pp. 2448–2451 (1993)
6. Craven, M., Shavlik, J.: Using Neural Networks for Data Mining. *Future Generation Computer Systems* 13, 211–229 (1997)
7. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: *Proceedings of the 1993 International Conference on Management of Data (SIGMOD 1993)*, May 1993, pp. 207–216 (1993)
8. Tunc, B., Dag, H.: Generating classification association rules with modified Apriori algorithm. In: *Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Databases*, pp. 384–387 (2006)
9. Hipp, J., Güntzer, U., Nakhaeizadeh, G.: Algorithms for association rule mining — a general survey and comparison. *ACM SIGKDD Explorations Newsletter* 2(1), 58–64 (2000)
10. Sanger, T.D.: Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks* 2, 459–473 (1989)
11. Haykin, S.: *Neural Networks: A Comprehensive Foundation*, 2nd edn. Prentice Hall, Englewood Cliffs (1999)
12. Fröhlich, J.: Kohenan Feature Map. *Neural Networks with Java 2004 Edition* (2009), <http://www.nnwj.de/> (accessed: February 12, 2009)
13. Marrone, P.: *The Complete Guide: All you need to know about Joone*. Joone: Java Object oriented Neural Engine (January 17, 2007), <http://biznetnetworks.dl.sourceforge.net/sourceforge/joone/JooneCompleteGuide.pdf> (accessed: 13.06.2008)
14. Klevecka, I., Lelis, J.: Pre-Processing of Input Data of Neural Networks: The Case of Forecasting Telecommunication Network Traffic. *Elektronikk* 3/4.2008, pp. 168–178 (2008)
15. Mendelsohn, L.: *Preprocessing Data for Neural Networks. Technical Analysis of Stocks and Commodities* (1993), http://www.tradertech.com/preprocessing_data.asp (accessed: 21.06.2008)
16. Heaton, J.: *Introduction to Neural Networks for Java*, 2nd edn. Heaton Research (October 2008)
17. Kotsiantis, S., Kanellopoulos, D.: Introduction, Association Rules Mining: A Recent Overview, Educational Software Development Laboratory, Department of Mathematics, University of Patras, Greece (2006)
18. Mandal, S.: Prediction of Tides using back-Propagation Neural Networks. In: *International Conference in Ocean Engineering* (December 2001)
19. Alahakoon, D., Halgamuge, S., Sirinivasan, B.: A Self Growing Cluster Development Approach to Data Mining. In: *Proceedings of IEEE International Conference on Systems, Man and Cybernetics, San Diego, USA*, pp. 2901–2906 (1998)

Application Research of Local Support Vector Machines in Condition Trend Prediction of Reactor Coolant Pump

Guohua Yan and Yongsheng Zhu

Theory of Lubrication and Bearing Institute
School of Mechanical Engineering
Xi'an Jiaotong University
Xi'an, Shaanxi 710049, China
yanguohua@gmail.com

Abstract. The difficulty in parameters selection of support vector machines (SVMs), which determines the performance of SVMs, limits the application of SVMs. In this paper, a directly determination (DD) method, which combines the existing practical approach used to compute parameters \mathcal{E} and C with another method used to compute λ , is introduced. This method determines the values of parameters directly from analyzing training data without running SVMs training process. The results show it gets better performance than usual grid search method in terms of predicting accuracy. Moreover, it reduces the spent time to a minimum. For predicting the condition trend of reactor coolant pump (RCP), a forecasting model which combines Local SVMs, whose parameters are determined by DD method, and Time Series is used. The results of experiments show that the model is able to predict the developing trend of time series of features reflecting the pump running condition preferably.

Keywords: Support Vector Machine, Parameters Selection, Condition Trend Prediction.

1 Introduction

The reactor coolant pump is a critical part of nuclear power plant, which is in charge of conveying heat generated in nuclear island to steam generator. If the pump failed, a hazard would be caused consequently. So, predicting the developing trend of the running condition of the pump and performing predictable maintenance is quite necessary for achieving high reliability.

Because of the noisy, non-stationary and chaotic characteristic of signal of mechanical equipment, the prediction method [1], which based on the stationary signal, such as AR model, MA model, and ARMA model, can not be efficiently used to predict the condition of mechanical equipment. In recent years, Support Vector Machines (SVMs) gradually become the hot research point in the field of artificial intelligence for its favorable generalization ability and have been successfully applied to predict non-stationary time series [2-4]. However, the hard selection of parameters of SVMs, which determine the performance of SVMs, in some extent limits the application of SVMs. Existing software implementations of SVMs usually treat these

parameters as user-defined inputs, or use grid search method based on k -folds cross validation to optimize these parameters. The former requires the user have a priori knowledge to the faced problem or user expertise. Obviously, this approach is not appropriate for non-expert users. The later is very time-consuming and not meets the time requirement of industry.

In this paper, the optimization of parameters of SVMs is studied. and Local SVMs, combining the local algorithm and SVMs, and getting better performance than traditional SVMs in practice [5,6], is used to forecast the condition trend of reactor coolant pump.

2 Principle of SVMs Regression

SVMs developed by Vapnik and his co-works are used for pattern recognition problem initially. In order to generalize the results obtained for pattern recognition problem to regression problem, ε -insensitive cost function is introduced [7]. To given training data set

$$(y_1, \mathbf{x}_1), (y_2, \mathbf{x}_2), \dots, (y_l, \mathbf{x}_l) \quad \mathbf{x} \in \mathbb{R}^n, y \in \mathbb{R} \quad (1)$$

The corresponding optimization problem is

$$\begin{aligned} \min \quad & \frac{1}{2} \|\omega\|^2 + C \frac{1}{l} \sum_{i=1}^l (\xi_i + \xi_i^*) \\ \text{s.t.} \quad & \begin{cases} y_i - (\omega \bullet \varphi(\mathbf{x}_i)) - b \leq \varepsilon + \xi_i & i = 1, \dots, l \\ (\omega \bullet \varphi(\mathbf{x}_i)) + b - y_i \leq \varepsilon + \xi_i^* & i = 1, \dots, l \\ \xi_i, \xi_i^* \geq 0 & i = 1, \dots, l \end{cases} \end{aligned} \quad (2)$$

where the C is a positive constant (penalty parameter), and coefficients ξ_i and ξ_i^* are slack factors.

This optimization formulation can be transformed into the dual problem and its solution is given by

$$f(\mathbf{x}) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) K(\mathbf{x}, \mathbf{x}_i) + b \quad (3)$$

where $K(\mathbf{x}, \mathbf{x}_i)$ is kernel function equaling $\varphi(\mathbf{x}) \bullet \varphi(\mathbf{x}_i)$ and α_i and α_i^* are lagrangian multipliers. The training points that appear with non-zero multipliers are called Support Vectors (SVs). In this paper, the radial basis kernel function (RBF) widely used in practice is selected:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\lambda \|\mathbf{x}_i - \mathbf{x}_j\|^2) \quad (4)$$

From Eq. (2) to Eq. (4), it can be known that SVMs generalization performance (estimation accuracy) depends on a good setting of hyperparameters \mathcal{E} , C and λ .

3 Optimization of Hyperparameters

Selection of parameter \mathcal{E} . Parameter \mathcal{E} controls the width of the \mathcal{E} -insentive zone. Too large or too small \mathcal{E} -value, corresponding to underfit or overfit learning respectively, doesn't make the SVM exhibits good performance.

It is well-known that the value of \mathcal{E} should be proportional to the input noise level, that is $\mathcal{E} \propto \sigma$, where σ is standard deviation of training targets. Based on this, Cherkassky [8] found the following practical prescription for \mathcal{E} :

$$\mathcal{E} = 3\sigma \sqrt{\frac{\ln l}{l}} \tag{5}$$

where σ is the standard deviation of input noise and l is the number of training samples. This expression provides good performance for various data set sizes, noise levels and target functions for SVMs regression.

However, in practice, the input noise variance is always not known. It should be estimated via k -nearest-neighbor's method:

$$\hat{\sigma}^2 = \frac{l^{1/5}k}{l^{1/5}k-1} \frac{1}{l} \sum_{i=1}^l (y_i - \hat{y}_i)^2 \tag{6}$$

Since accurate estimation of noise variance is not affected much by specific k -value, we set $k=3$.

Selection of parameter C . Parameter C determines the trade off between empirical risk and model complexity. If C is too large (infinity), then the objective is to minimize the empirical risk only, without regard to model complexity in the optimization formulation (Eq. 2). If C is too small (zero), then the objective does not consider the penalty on deviations which is larger than \mathcal{E} . Under these two situations, it is impossible for SVMs to obtain good regularization performance. A good value of C should be chosen according to the range of output values of training data [9,10]. The prescription for parameter C is given by:

$$C = \max(|\bar{y} + 3\sigma_y|, |\bar{y} - 3\sigma_y|) \tag{7}$$

where \bar{y} and σ_y are the mean value and the standard deviation of training targets respectively.

Selection of parameter λ . The role of kernel function in the problem of SVMs is to map data in the low dimensional space into the high dimensional space. Different λ mean different distribution complexity of data in high dimensional space. Parameter λ can be computed by the following kernel alignment method [11-13]:

$$\hat{A}(K, YY^T) = \frac{\langle K, YY^T \rangle}{\sqrt{\langle K, K \rangle \langle YY^T, YY^T \rangle}} \tag{8}$$

where $\hat{A}(K, YY^T)$ is the value of kernel alignment, K is kernel matrix, and Y is the vector of $(y_1 - \bar{y}, y_2 - \bar{y}, \dots, y_l - \bar{y})^T$. $\langle K, YY^T \rangle$ is the Frobenius inner product between matrices K and YY^T , which is denoted as $\langle K, YY^T \rangle = \sum_{i,j=1}^l K_{ij} \cdot YY^T_{ij}$. The

optimal λ makes Eq. (8) get the maximum value.

To evaluate the performance of the method described above, we compare it with the usual method of two-dimensional grid search, used to select C and λ while \mathcal{E} is determined by Eq. (5). For convenience, we definite it as directly determination (DD) method.

Through a amount of experiments, we found that for different \mathcal{E} , the curves of predicting error with λ and C are similar, which can be illustrated by Fig. 1, and the \mathcal{E} ($\log_2 \mathcal{E} = -0.4816$) selected by Eq. (5) results in small predicting error and a small quantity of SVs, as shown in Fig. 2. The similar results can be obtained on other datasets.

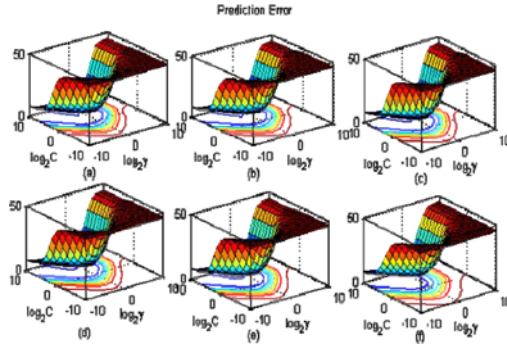


Fig. 1. On mpg dataset, prediction error as a function of paramters λ and C for given \mathcal{E} . From subfigure (a) to (f), $\log_2 \mathcal{E}$ is -8, -6, -4, -2, 0, 2, respectively

For the Grid Search(GS) method, we uniformly discretize the $[-10,10] \times [-10,10]$ region to $21^2 = 441$ points. At each point, a fivefold cross-validation is conducted. The point with the best CV accuracy is chosen and used to predict the test data. All the experiments are based on the software LibSVM [14]. For each dataset [15], 80 percent of the data is used for training and the rest is used for testing. The prediction performance is evaluated by Mean Square Error (MSE).

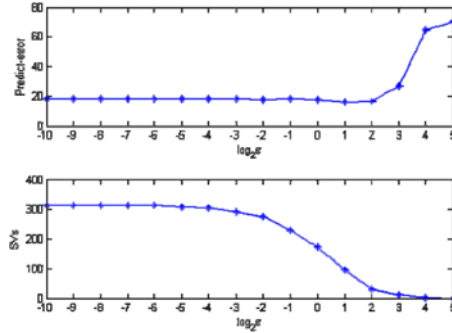


Fig. 2. On mpg dataset, the predict-error and SVs as a function of \mathcal{E}

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - y'_i)^2 \tag{9}$$

The results of the experiment are presented in Tabel 1. It can be clearly seen that the DD method is very competitive with the usual grid search method in terms of MSE and cost much less time, which is because DD method optimizes parameters C and λ without experiencing training process, compared with GS method 5×441 training times.

Table 1. Comparison of the two parameters selection methods

Dataset	Grid Search Method		Directly Determine Method	
	MSE	Time (s)	MSE	Time (s)
pyrim	0.0031	4.1719	0.0062	0.0938
triazines	0.0249	20.375	0.0301	0.0625
bodyfat	3.29E-06	23.1719	1.16E-06	0.1094
mpg	22.863	52.3906	18.6925	0.1094
housing	127.7668	73.0156	113.8565	0.1563
mg	0.0145	2699.4	0.0146	0.4375
spacega	0.0452	22677	0.0228	1.3125
abalone	3.2527	9743.6	3.2222	2.9531

4 Application of SVMs in Predicting Behavior of RCP

The signal collected from RCP is a time series $\{x_i, i=1, \dots, n\}$, such as vibration amplitude, temperature, to name a few, which violates with the data format required by SVM and needs transformation. The concept of phase space reconstruction is introduced to solve this problem [16,17]. It is presented as follows:

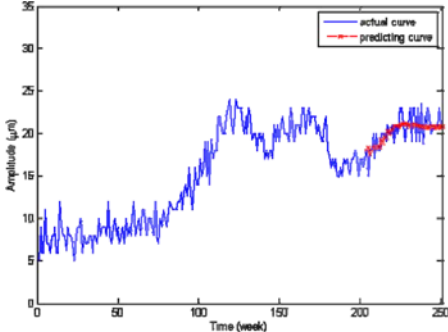


Fig. 4. Prediction of motor vibration

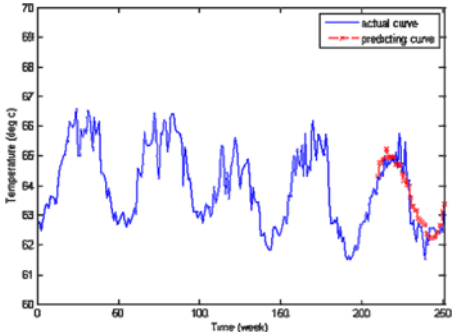


Fig. 5. Prediction of temperature of motor radial bearing

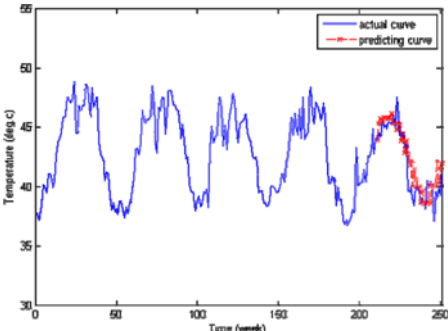


Fig. 6. Prediction of temperature of motor thrust bearing

5 Conclusions

Condition predicting is very important for predictive maintenance. In order to apply SVMs into real practice effectively, we determine the parameters directly from analysing train data without running SVM training process. This method not only guarantees the predicting accuracy, but also reduces the consumed time into the minimum. Combining the phase space reconstruction, Local SVMs are used in the trend prediction of field data of reactor coolant pump. It obtains good performance and is able to predict the developing trend correctly. Local SVMs can predict the condition trend automatically, avoiding the interference of human, and offer one choice for predictive maintenance.

Acknowledgments. This work was supported by the National High Technology Research and Development Program of China under the grant No. 2006AA04Z420.

References

1. Yang, S.Z., Wu, Y.: Time Series Analysis in Engineer Application (in Chinese). Huazhong University of Science and Technology Press, Wuhan China (1999)
2. Kin, T.Y., Oh, K.J., Kim, C., et al.: Artificial neural networks for non-stationary time series. *Neurocomputing* 61, 439–447 (2004)
3. Zhang, X.N.: Research on the condition monitoring and forecasting for large rotating machine (in Chinese). Ph.D thesis of Xi'an Jiaotong University, Xi'an China (1998)
4. Shu, Y.P.: Forecasting theory based on neural network and it's application (in Chinese). Ph.D thesis of Xi'an Jiaotong University, Xi'an China (1994)
5. Bottou, L., Vapnik, V.N.: Local Learning Algorithms. *Neural Computation* 4, 888–900 (1992)
6. Fernandez, R.: Predicting time series with a local support vector regression machine. In: *ACAI 1999* (1999)
7. Vapnik, V.N.: Statistical Learning Theory (in Chinese). In: Xu J.H., Zhang, X.G. (eds.). Publishing House of Electronics Industry, Beijing (2004)
8. Cherkassky, V.C., Ma, Y.Q.: Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks* 17, 113–226 (2004)
9. Matterna, D., Haykin, S.: Support vector machines for dynamic reconstruction of a chaotic system. In: *Advances in kernel methods: Support vector machines*. MIT Press, Cambridge (1999)
10. Tay, F.E.H., Cao, L.J.: Application of Support Vector Machines in Financial Time Series Forecasting. *Omega* 29, 309–317 (2001)
11. Cristianini, N., Kandola, J., Elisseeff, A., et al.: On kernel target alignment. MIT Press, Cambridge (2002)
12. Kandola, J., Taylor, J.S., Cristianini, N.: On the extensions of kernel alignment. Technical Report (2002), <http://www.neurocolt.org>
13. Liu, X.D., Luo, B., Chen, Z.Q.: Optimal model selection for support vector machines. *Journal of Computer Research and Development* 42(4), 576–581 (2005)
14. Chang, C.C., Lin, C.J.: LibSVM: A library for support vector machines (2001), <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

15. Chang, C.C., Lin, C.J.: LibSVM Data: Regression (2001), <http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html>
16. Müller, K.R., Smola, A.J., Räsch, G., et al.: Predicting time series with support vector machines. In: Artificial Neural Networks - ICANN 1997 (1997)
17. Muller, K.-R., et al.: Using support vector machines for time series prediction. In: Advances in kernel methods: support vector learning, pp. 243–253. MIT Press, Cambridge (1999)
18. Cao, L.: Practical method for determining the minimum embedding dimension of a scalar time series. *Physica D* 110, 43–50 (1997)
19. Fraser, A.M., Swinney, H.L.: Independent coordinates for strange attractors from mutual information. *Phys. Rev. A* 33(2), 1134–1140 (1986)

Asymptotic Synchronization for Pulse-Coupled Oscillators with Delayed Excitatory Coupling Is Impossible*

Wei Wu¹ and Tianping Chen^{2,**}

¹ Lab. of Nonlinear Mathematics Science, Institute of Mathematics, Fudan University, Shanghai, 200433, P.R. China
wuweifd@vip.sina.com

² Lab. of Nonlinear Mathematics Science, Institute of Mathematics, Fudan University, Shanghai, 200433, P.R. China
tchen@fudan.edu.cn

Abstract. Fireflies, as one of the most spectacular examples of synchronization in nature, have been investigated widely. Mirollo and Strogatz (1990) proposed a pulse-coupled oscillator model to explain the synchronization of South East Asian fireflies (*Pteroptyx malaccae*). However, transmission delays were not considered in their model. In fact, when transmission delays are introduced, the dynamic behaviors of pulse-coupled networks change a lot. In this paper, pulse-coupled oscillator networks with delayed excitatory coupling are studied. A concept of synchronization, named weak asymptotic synchronization, which is weaker than asymptotic synchronization, is proposed. We prove that for pulse-coupled oscillator networks with delayed excitatory coupling, weak asymptotic synchronization cannot occur.

Keywords: Synchronization, Desynchronization, Pulse-coupled oscillators.

1 Introduction

Since Buck's pioneer work [1] on the synchronization of fireflies was published in 1988, many kinds of biological models have been proposed for studying flashing behaviors of fireflies. In particular, inspired by Peskin's model for self-synchronization of the cardiac pacemaker [2], Mirollo and Strogatz (MS) proposed a pulse-coupled oscillator model with undelayed excitatory coupling to explain the synchronization of huge congregations of South East Asian fireflies (*Pteroptyx malaccae*) [3]. The main result in [3] is that for almost all initial conditions, the system eventually becomes completely synchronized. With the framework of MS model, many results on pulse-coupled networks with undelayed excitatory coupling have been obtained [4,5,6,7,8].

However, transmission delays are unavoidable in real biological systems. For example, experiments show that normally the transmission delays of most fireflies from sensors to motor actions of flashing are around 200ms (see [1]). In fact, it has been shown in

* This work is supported by National Science Foundation of China 60774074.

** Corresponding author.

previous research [9,10,11,12] that the system becomes desynchronized as transmission delays are introduced into MS model. To the best of our knowledge, before the publication of our previous paper [13], this desynchronization was proved only for the case of two pulse-coupled oscillators. Instead, for the case that the number of pulse-coupled oscillators exceeds 2, the phenomenon of desynchronization was revealed only by simulations. In [13], we proved that under reasonable assumptions, $N \geq 2$ pulse-coupled oscillators with delayed excitatory coupling cannot achieve complete synchronization.

The complete synchronization defined in [13] (see Definition 1 in Section 3) requires that all the oscillators reach perfect synchronization in finite time. It is natural to raise the following question: Would the phase differences among oscillators converge to zero as time goes to infinity? Equivalently, can asymptotic synchronization occur? We will address this question in this paper. More precisely, we will give a negative answer to the question. Furthermore, we even prove a stronger result: synchronization in a weaker sense also can not occur. For this purpose, instead of discussing asymptotic synchronization directly, we discuss a weaker form of synchronization, called weak asymptotic synchronization, in which the differences of firing times for oscillators converge to zero as time goes to infinity (see Definition 3 in Section 3). This definition originates from the observation that usually synchronization of fireflies only requires that the differences of firing times converge to zero. Since, at night fireflies can be observed only when they flash.

In this paper, we not only exclude the possibility of achieving weak asymptotic synchronization, but also reveal, to some extent, the regularity of flashing behaviors of pulse-coupled networks with delays. Firstly, we prove an important lemma (Lemma 2), which shows that the differences of firing times for oscillators will increase exponentially only if the system evolves to a state in which all the differences of firing times are less than the delay τ . It is well known that for MS model, if the phase differences are sufficiently small, complete synchronization will be achieved after the next round of firings (see [3]). Instead, from the lemma, one can see that when transmission delays are introduced, the dynamic behaviors of pulse-coupled networks change a lot. Then, by the lemma, we prove a theorem, which describes the limit behavior of sequences of firing times. Using this theorem, we can easily obtain that pulse-coupled oscillator networks with delayed excitatory coupling cannot achieve weak asymptotic synchronization.

The rest of the paper is organized as follows: In Section 2, we describe the network model. In Section 3, some definitions are given, and several results in [13], which will be used in the proof of the main results, are recalled. In Section 4, the main results are proved. We conclude the paper in Section 5.

2 Model

The network consists of $N \geq 2$ pulse-coupled identical oscillators. The coupling is all-to-all. Each oscillator is characterized by a state variable x_i which is assumed to increase toward a threshold at $x_i = 1$ according to $\dot{x}_i = f(\varphi_i)$, where $f : [0, 1] \rightarrow [0, 1]$ is smooth, monotonic increasing, and concave down, i.e., $f'(\theta) > 0$ and $f''(\theta) < 0$ for all $\theta \in (0, 1)$. Here, $\varphi_i \in [0, 1]$ is a phase variable such that (i) $d\varphi_i/dt = 1/T$, where T is the cycle period (without loss of generality, we assume $T = 1$); (ii) $\varphi_i = 0$ when

the i th oscillator is at its lowest state $x_i = 0$, and (iii) $\varphi_i = 1$ at the end of the cycle when the i th oscillator reaches the threshold $x_i = 1$. When x_i reaches the threshold, the i th oscillator fires and x_i jumps back instantly to zero, after which the cycle repeats. That is,

$$x_i(t) = 1 \Rightarrow x_i(t^+) = 0. \quad (1)$$

Because of the presence of the transmission delay, the oscillators interact by the following form of pulse-coupling: when an oscillator fires at time t , it emits a spike; after a transmission delay τ , the spike reaches all other oscillators at time $t + \tau$ and pulls them up by an amount ε , or pulls them up to firing, whichever is less than ε . That is,

$$x_i(t) = 1 \Rightarrow x_j(t + \tau) = \min[1, x_j((t + \tau)^-) + \varepsilon], \quad \forall j \neq i. \quad (2)$$

Note that if $n \geq 2$ oscillators fire at time t simultaneously, then at time $t + \tau$, the state variables of the n oscillators increase by an amount $(n - 1)\varepsilon$ and the state variables of all other oscillators increase by an amount $n\varepsilon$. That is,

$$\begin{aligned} x_{i_1}(t) = x_{i_2}(t) = \dots = x_{i_n}(t) = 1 \\ \Rightarrow x_j(t + \tau) = \begin{cases} \min[1, x_j((t + \tau)^-) + (n - 1)\varepsilon], & j \in \{i_1, \dots, i_n\} \\ \min[1, x_j((t + \tau)^-) + n\varepsilon], & j \notin \{i_1, \dots, i_n\} \end{cases} \end{aligned} \quad (3)$$

In addition, as in [13], we make two assumptions:

- (A1) The system starts at time $t = 0$ with a set of initial states $0 < x_i(0) \leq 1$, and there are no firings in the time interval $[-\tau, 0)$.
(A2) The transmission delay τ and the coupling strength ε satisfy $f(2\tau) + N\varepsilon < 1$.

With the monotonicity of f , the state variable x_i and the phase variable φ_i are one-to-one correspondence. Therefore, the synchronization of the state variables x_1, \dots, x_N is equivalent to the synchronization of the phase variables $\varphi_1, \dots, \varphi_N$. In the following, instead of investigating x_i , we investigate dynamical behaviors of φ_i directly.

From the above description, the phase variable possesses the following properties.

Proposition 1. *The phase variable φ_i satisfies:*

- (a) *If $n \geq 1$ spikes reach the i th oscillator at time t , then $\varphi_i(t) = f^{-1}(\min[1, f(\varphi_i(t^-)) + n\varepsilon])$;
If no spikes reach the i th oscillator at time t , then $\varphi_i(t) = \varphi_i(t^-)$.*
(b) *If $\varphi_i(t) = 1$, then $\varphi_i(t^+) = 0$;
If $\varphi_i(t) < 1$, then $\varphi_i(t^+) = \varphi_i(t)$.*
(c) *If no spikes reach the i th oscillator in the time interval (t_1, t_2) and the i th oscillator do not fire in (t_1, t_2) , then $\varphi_i(t_2^-) = \varphi_i(t_1^+) + (t_2 - t_1)$.*

For the convenience of later use, we introduce the notation $F_n(\theta) = f^{-1}(\min[1, f(\theta) + n\varepsilon])$, where $0 \leq \theta \leq 1$ and $n \in \mathbb{Z}^+ = \{z \in \mathbb{Z} | z \geq 0\}$. Then, Proposition 1(a) becomes that if $n \geq 1$ spikes reach the i th oscillator at time t , then $\varphi_i(t) = F_n(\varphi_i(t^-))$; if no spikes reach the i th oscillator at time t , then $\varphi_i(t) = \varphi_i(t^-) = F_0(\varphi_i(t^-))$.

3 Preliminaries

In this section, we give some definitions, and present some results obtained in [13].

Firstly, we give several concepts of synchronization in pulse-coupled oscillator networks.

Definition 1. *If for a given set of initial phases $[\varphi_1(0), \varphi_2(0), \dots, \varphi_N(0)]$, there exists a $t_0 \geq 0$ such that the phase variables of arbitrary oscillators i and j satisfy*

$$\varphi_i(t) = \varphi_j(t), \text{ for all } t \geq t_0, \quad (4)$$

*then we say that for this set of initial phases, the pulse-coupled oscillator network can achieve **complete synchronization**.*

Definition 2. *If for a given set of initial phases $[\varphi_1(0), \varphi_2(0), \dots, \varphi_N(0)]$, the phase variables of arbitrary oscillators i and j satisfy*

$$\lim_{t \rightarrow +\infty} d(\varphi_i(t), \varphi_j(t)) = 0, \quad (5)$$

where $d(\cdot, \cdot)$ is defined by

$$d(\theta_1, \theta_2) = \min [|\theta_1 - \theta_2|, \theta_1 + 1 - \theta_2, \theta_2 + 1 - \theta_1], \quad (6)$$

*then we say that for this set of initial phases, the pulse-coupled oscillator network can achieve **asymptotic synchronization**.*

Definition 3. *If for a given set of initial phases $[\varphi_1(0), \varphi_2(0), \dots, \varphi_N(0)]$, there exist non-negative integers k_1, k_2, \dots, k_N such that the firing times of arbitrary oscillators i and j satisfy*

$$\lim_{m \rightarrow +\infty} |t_{m+k_i}^i - t_{m+k_j}^j| = 0, \quad (7)$$

where t_m^r is the time at which the r th oscillator fires its m th time, then we say that for this set of initial phases, the pulse-coupled oscillator network can achieve **weak asymptotic synchronization**.

Remark 1. It is clear that complete synchronization is a special case of asymptotic synchronization, and asymptotic synchronization is a special case of weak asymptotic synchronization. The relations can be expressed as follows:

Complete synchronization \subseteq Asymptotic synchronization \subseteq Weak asymptotic synchronization

Secondly, we present several results in [13], which will be used in the sequel.

Proposition 2. (Proposition 2(e)-(f) in [13])

The function $F_n(\theta)$ has the following properties:

- (a) *If $0 \leq \theta, \delta < 1, n \geq 0$ and $F_n(\theta + \delta) < 1$, then $F_n(\theta) + \delta \leq F_n(\theta + \delta)$, and the equality holds if and only if $\delta = 0$ or $n = 0$;
equivalently, if $0 \leq \theta_1 \leq \theta_2 < 1, n \geq 0$ and $F_n(\theta_2) < 1$, then $\theta_2 - \theta_1 \leq F_n(\theta_2) - F_n(\theta_1)$,*

and the equality holds if and only if $\theta_1 = \theta_2$ or $n = 0$.

(b) $F_{n_1}(F_{n_2}(\theta)) = F_{n_1+n_2}(\theta)$ for all $0 \leq \theta \leq 1$ and all $n_1, n_2 \in \mathbb{Z}^+$.

Proposition 3. (Lemma 1 in [L3]) Under the assumptions (A1) and (A2), if an oscillator fires at time t_1 and t_2 with $t_1 \neq t_2$, then $|t_1 - t_2| > 2\tau$.

Proposition 4. (Theorem 1 in [L3]) Under the assumptions (A1) and (A2), from any initial phases (other than $\varphi_1(0) = \dots = \varphi_N(0)$), the pulse-coupled oscillator network with delayed excitatory coupling cannot achieve complete synchronization.

4 Main Results

Before deriving the main results, we prove the following two lemmas.

Lemma 1. There exists a constant $\alpha > 1$ such that the inequality

$$\alpha(\theta_2 - \theta_1) < F_n(\theta_2) - F_n(\theta_1)$$

holds for any $0 \leq \theta_1 < \theta_2 < 1$ and $n \geq 1$ satisfying $F_n(\theta_2) < 1$.

Lemma 2. Suppose that the assumptions (A1) and (A2) are satisfied, and $m_1, \dots, m_N > 1$ are N integers. For some set of initial phases, if the firing times $t_{m_1}^1, \dots, t_{m_N}^N$ and $t_{m_1+1}^1, \dots, t_{m_N+1}^N$ satisfy

$$|t_{m_i}^i - t_{m_j}^j| < \tau, \quad i, j = 1, \dots, N, \quad (8)$$

$$|t_{m_i+1}^i - t_{m_j+1}^j| < \tau, \quad i, j = 1, \dots, N, \quad (9)$$

then for any pair of firing times $t_{m_i}^i$ and $t_{m_j}^j$ satisfying $t_{m_i}^i \leq t_{m_j}^j$, we have

$$\alpha(t_{m_j}^j - t_{m_i}^i) \leq t_{m_j+1}^j - t_{m_i+1}^i, \quad (10)$$

and the equality holds if and only if $t_{m_i}^i = t_{m_j}^j$, where $\alpha > 1$ is the constant in Lemma 1.

Proof of Lemma 1 and Lemma 2 are omitted here.

Lemma 2 shows that the differences of firing times for oscillators will increase exponentially only if the system evolves to a state in which all the differences of firing times are less than the delay τ . Using this lemma and Propositions 3-4, we can prove the following theorem, which describes the limit behavior of sequences of firing times.

Theorem 1. Under the assumptions (A1) and (A2), from any initial phases (other than $\varphi_1(0) = \dots = \varphi_N(0)$), the firing times of the oscillators satisfy

$$\limsup_{m \rightarrow +\infty} \max_{1 \leq i < j \leq N} |t_{m+k_i}^i - t_{m+k_j}^j| \geq \tau, \quad (11)$$

where $k_i, i = 1, \dots, N$, are arbitrary non-negative integers.

Proof: We employ “proof by contradiction”. Suppose that there exists a set of initial phases $[\varphi_1(0), \dots, \varphi_N(0)]$, which are not all equal, such that

$$\limsup_{m \rightarrow +\infty} \max_{1 \leq i < j \leq N} |t_{m+k_i}^i - t_{m+k_j}^j| < \tau \quad (12)$$

holds for some set of non-negative integers k_1, \dots, k_N . It means that there exists an integer $M > 0$ such that for all $i, j = 1, \dots, N$ and $m > M$, there holds

$$|t_{m+k_i}^i - t_{m+k_j}^j| < \tau. \quad (13)$$

In the following, we give the contradiction.

From Propositions 3 and 4, it can be seen that the firing times $t_{M+1+k_1}^1, \dots, t_{M+1+k_N}^N$ will never be all the same. In fact, if $t_{M+1+k_1}^1 = \dots = t_{M+1+k_N}^N = T$, then by Proposition 3, in the time interval $[T-2\tau, T]$, no oscillators fire. Combining this with the equalities $\varphi_1(t_{M+1+k_1}^1) = \dots = \varphi_N(t_{M+1+k_N}^N) = 1$, i.e., $\varphi_1(T) = \dots = \varphi_N(T) = 1$, we conclude $\varphi_1(t) = \dots = \varphi_N(t)$ for all $t \geq T$, which contradicts Proposition 4.

Denote

$$\delta_0 = \max_{1 \leq i < j \leq N} |t_{M+1+k_i}^i - t_{M+1+k_j}^j|,$$

which is positive by previous arguments. From Lemma 2 and (13), it follows that for all $m > M$,

$$\max_{1 \leq i < j \leq N} |t_{m+k_i}^i - t_{m+k_j}^j| > \alpha^{m-M} \delta_0.$$

It means that

$$\lim_{m \rightarrow +\infty} \max_{1 \leq i < j \leq N} |t_{m+k_i}^i - t_{m+k_j}^j| = +\infty,$$

which contradicts (12).

Theorem 1 is proved. \square

It can be easily seen that Theorem 1 excludes the possibility that pulse-coupled networks with delayed excitatory coupling achieve weak asymptotic synchronization. So, we have the following theorem.

Theorem 2. *Under the assumptions (A1) and (A2), from any initial phases (other than $\varphi_1(0) = \dots = \varphi_N(0)$), the pulse-coupled oscillator network with delayed excitatory coupling cannot achieve weak asymptotic synchronization.*

By Theorem 2 and Remark 2, the following theorem is clear.

Theorem 3. *Under the assumptions (A1) and (A2), from any initial phases (other than $\varphi_1(0) = \dots = \varphi_N(0)$), the pulse-coupled oscillator network with delayed excitatory coupling cannot achieve asymptotic synchronization.*

5 Conclusions

In this paper, the pulse-coupled network model is further studied. A concept of synchronization, called weak asymptotic synchronization, which is weaker than asymptotic synchronization, is proposed. An important lemma is given. With this lemma, it is

proved that under the assumptions (A1) and (A2), from any initial phases (other than $\varphi_1(0) = \dots = \varphi_N(0)$), the network cannot achieve weak asymptotic synchronization.

References

1. Buck, J.: Synchronous rhythmic flashing of fireflies, Part II. *Q. Rev. Biol.* 63(3), 265–289 (1988)
2. Peskin, C.S.: *Mathematical aspects of heart physiology*, Courant Institute of Mathematical Sciences, pp. 268–278. New York University, New York (1975)
3. Mirollo, R.E., Strogatz, S.H.: Synchronization of pulse-coupled biological oscillators. *SIAM J. Appl. Math.* 50, 1645–1662 (1990)
4. Vanreeswijk, C., Abbott, L.F.: Self-sustained firing in populations of integrate-and-fire neurons. *SIAM J. Appl. Math.* 53, 253–264 (1993)
5. Chen, C.C.: Threshold effects on synchronization of pulse-coupled oscillators. *Phys. Rev. E* 49, 2668–2672 (1994)
6. Corral, A., Perez, C.J., Diazguilera, A., et al.: Self-organized criticality and synchronization in a lattice model of integrate-and-fire oscillators. *Phys. Rev. Lett.* 74, 118–121 (1995)
7. Mathar, R., Mattfeldt, J.: Pulse-coupled decentral Synchronization. *SIAM J. Appl. Math.* 56, 1094–1106 (1996)
8. Goel, P., Ermentrout, B.: Synchrony, stability, and firing patterns in pulse-coupled oscillators. *Phys. D* 163, 191–216 (2002)
9. Nischwitz, A., Glunder, H.: Local lateral inhibition—a key to spike synchronization. *Biol. Cybern.* 73, 389–400 (1995)
10. Ernst, U., Pawelzik, K., Geisel, T.: Synchronization induced by temporal delays in pulse-coupled oscillators. *Phys. Rev. Lett.* 74, 1570–1573 (1995)
11. Knoblauch, A., Palm, G.: Scene segmentation by spike synchronization in reciprocally connected visual areas. II. Global assemblies and synchronization on larger space and time scales. *Biol. Cybern.* 87, 168–184 (2002)
12. Coombes, S., Lord, G.J.: Desynchronization of pulse-coupled integrate-and-fire neurons. *Phys. Rev. E* 55, 2104–2107 (1997)
13. Wu, W., Chen, T.P.: Desynchronization of pulse-coupled oscillators with delayed excitatory coupling. *Nonlinearity* 20, 789–808 (2007)

Missing Data Imputation Through the Use of the Random Forest Algorithm

Adam Pantanowitz and Tshilidzi Marwala*

School of Electrical & Information Engineering
University of the Witwatersrand, Johannesburg
Private Bag 3, Wits, 2050, South Africa
adam.pantanowitz@wits.ac.za

Abstract. This paper presents a comparison of different paradigms used for missing data imputation. The data set used is HIV seroprevalence data from an antenatal clinic study survey performed in 2001. Data imputation is performed through five methods: Random Forests; auto-associative neural networks with genetic algorithms; auto-associative neuro-fuzzy configurations; and two random forest and neural network based hybrids. Results indicate that Random Forests are superior in imputing missing data for the given data set in terms of accuracy and in terms of computation time, with accuracy increases of up to 32 % on average for certain variables when compared with auto-associative networks. While the concept of hybrid systems has promise, the presented systems appear to be hindered by their auto-associative neural network components.

Keywords: auto-associative, imputation, missing data, neural network, random forest.

1 Introduction

Real world studies are often impacted negatively due to data that are missing. This common problem creates difficulty with data analysis, study and visualisation [1,2]. Insights into characteristics of the data may be reduced due to missing information and, furthermore, the underlying cause for the missing data may make the missing data particularly interesting or of significance to the study. Cascaded systems, such as those responsible for decision making based on certain decision making policies, may be hindered by the missing information and rendered unusable. For these reasons, it is important to find effective and viable methods to impute missing data.

This paper evaluates the concept, classification, problem and treatment of missing data. A background on the methods and paradigms used is provided, followed by a description of the implementation. The data set is considered, and thereafter, comparisons are drawn between the implemented paradigms. Finally a discussion is presented and conclusions are reached.

* Tshilidzi Marwala has since become Executive Dean of the Faculty of Engineering and the Built Environment at the University of Johannesburg. P.O. Box 524, Auckland Park, 2006, Johannesburg, South Africa. tmarwala@uj.ac.za

2 Missing Data

Missing data are a problem inherent and common in data collection, especially when dealing with large, real world data sets. Missing data impact on decision support systems and make statistical evaluation methods difficult to perform. Results are degraded through the use of arbitrary or random assignment to the missing data elements [3]. In surveys in particular, information may be missing due to incomplete variable collection and non-response from subjects, poorly defined surveys, and data being removed for reasons such as confidentiality [12]. These explanations for missing survey data may provide insight into the large proportion of missing data in the set used in this study, as discussed in section 5.

2.1 Missing Data: Categorisation and Mechanism

Missing data can be categorised based on the pattern and mechanism of absence. The methods with which the missing data are dealt are dependent on this categorisation. Three broad categories for pattern absence are defined: *monotone missingness*, *file matching*, and *general missingness* [4,5].

Missing data are also often classified into one of three mechanisms, as defined by Little and Rubin [4]. The mechanisms, in order from least to most dependent on other information, are: missing completely at random (MCAR); missing at random (MAR); and the non ignorable case [14]. In the MCAR case, data cannot be predicted using any information in the set, known or unknown. For the MAR mechanism, there is a correlation between the missing data and the observed data, but not necessarily on the values of the missing data [6].

2.2 Dealing with Missing Data

A number of strategies have been devised for dealing with missing data. The simplest means is discarding the instances for which data are missing (a complete-case method), which is inefficient and potentially leads to biased observations and information waste [1]. Despite this, the method is used commonly in practice [2]. Other techniques include *available-case procedures*, *weighting procedures* and *imputation-based* procedures [6]. We consider imputation methods which involve predicting the values of the missing data and can be applied to MCAR and MAR cases [7]. Two categories of techniques exist, non-model based and model-based. Non-model based approaches include mean imputation and hot-deck imputation, techniques which are said to decrease the variance in statistical procedures, lead to standard errors and to result bias [6]. Model-based approaches include regression-based techniques, multiple imputation [7], expectation maximisation [8] and neural network (NN) based approaches [18,9].

3 Background

A number of learning paradigms are considered which form networks and hybrid networks for comparative purposes in this work. These are generally connected in auto-associative configurations [9], as discussed in section 4.

3.1 Random Forests

Ensemble or network committees are algorithms in machine learning that combine individual paradigms to form combinations that are often more accurate than the individual classifier alone [10]. In the classification case, overall predictions can be obtained from such a network using a weighted or an unweighted voting system; in the regression case, overall predictions can be chosen through an averaging technique. Obtaining a general understanding of why such methods succeed is an active area of research [10,11].

A *decision tree* is a tree with nodes that contain information corresponding to attributes in the input vectors. This information is used to follow a decision path for a given set of input attributes, depending on either thresholding nodes (as in the case of a continuous variable) or categorical nodes (as in the case of categorical data) [12]. Even though decision trees have appeal for being straightforward and fast, they are prone to being overly adapted to the training data or to a loss in accuracy for generalisation through tree pruning [13].

“Random Forest” (RF) is an algorithm that generalises ensembles of decision trees [14], with the ability to perform both regression and classification. RFs make use of bagging (bootstrap aggregation) to combine multiple random predictors in order to aggregate predictions [15], allowing for high complexity without over-generalising and over-fitting to the training data [13]. RF has been used with success in the context of missing data [12]. RFs were first introduced in 2000 and are a trademark of Cutler and Breiman [10].

If Θ is the possible variables, and $h(x, \Theta)$ denotes a tree grown using Θ to classify a vector x , then an RF can be defined as $f = \{h(x, \Theta_k)\}$, $k = 1, 2, \dots, K$, in which $\Theta_k \subseteq \Theta$ [12,15]. Each tree contains an individually selected *subset* of the overall attribute collection. The algorithm for RF growth is presented in [14].

RFs are a good candidate for a missing data study [10,14] and have recently been an area of active research since they have advantageous features and high success rates [10]. They are fast and have accuracy greater than that of single classification and regression trees (CART). They are furthermore impervious to over-fitting the data and do not have dimensionality problems - running effectively on thousands of variables. RFs give a self-assessment and have built in variable importance assessment capabilities [14].

3.2 Other Paradigms

Multilayer Perceptron (MLP) Artificial Neural Networks (ANNs) are ANNs that consist of an interconnection of the processing elements, generally placed in three classes: the input layer, the output layer and the hidden layer [16]. A process of supervised learning allows the weights of the network to be adjusted yielding a feed-forward network capable of modelling complex and non-linear relationships [17]. A number of different optimisation strategies are available in training the network, such as conjugate gradient descent [16].

Fuzzy Inference Systems (FISs) operate through a process of fuzzification, operation and implication [18]. FISs are well suited to knowledge of linguistic if-then rules, offering an advantage in terms of learning capability, while data based

learning is better suited to ANNs [19]. Neuro-fuzzy systems provide the benefit of both subsystems and consist of rule sets and inference systems combined with or governed by a connectionist structure for optimisation and adaptation to given data. Adaptive neuro-fuzzy inference system (ANFIS) implements a Takagi Sugeno (TS) FIS and consists of five layers, as indicated by the schematic architecture of the system presented in [20,21].

Genetic Algorithms (GAs) are search methods that are widely used to solve optimisation problems. Genetic algorithms employ their heuristic search by modelling properties of biological evolution including: crossover; inheritance; mutation; and selection, to solve optimisation problems [22]. Convergence exists due to the fitness of an individual (representing an element in the search space that may be an appropriate solution to the problem) in a given population dominating over another individual [22,23]. Through an evolutionary process, survival of the fittest ensues.

Auto-Associative NN Encoder Networks are system models which are trained to recall an input. The number of outputs is thus equal to the number of inputs [9]. The networks usually have fewer hidden nodes than inputs (or outputs), creating a *bottleneck*. The auto-encoder network detects missing data by forward propagating known elements and a predicted value for the unknown elements, and minimising the overall error between the input and the output using an intelligent search method, such as a GA [8,9].

4 Methodology and System Topologies

The RFs used throughout the analysis generally have 70 trees. The parameter for minimum size of terminal nodes set at 7, and number of variables to be sampled randomly at each split (m) is set to 3 (much less than the total number of inputs M which is 14 [24]). This combination was determined experimentally to be the optimal set of parameters through maximisation of the number of hits (i.e. the number of correct predictions). Regression RFs are used when predicting ordinal variables, which are encoded to be continuous values ranging from 0 - 1, and classification RFs when predicting categorical variables, such as HIV status (indicated in table 1). Since each RF predicts a single variable, an attempt was made to form RFs to predict each of the fourteen variables, and combine this with a GA to form an RF based auto-associative network. This method does not yield favourable results, but the resulting RFs are used to impute the different missing variables. For multiple missing values, the methodology presented in figure 1 is employed.

The **Auto-associative Neural Network** combined with a **GA (AANN-GA)** is used as in [9]. For the 14 input/output system, the optimal number of hidden nodes, determined experimentally, is 11. The number of training cycles is 400, based on the minimum point of the validation curve and a linear activation function is used with scaled-conjugate descent training.

The **Auto-associative Adaptive Neuro-Fuzzy Inference System** with a **GA (AANF-GA)** implements a network of 14 ANFIS networks. Since each

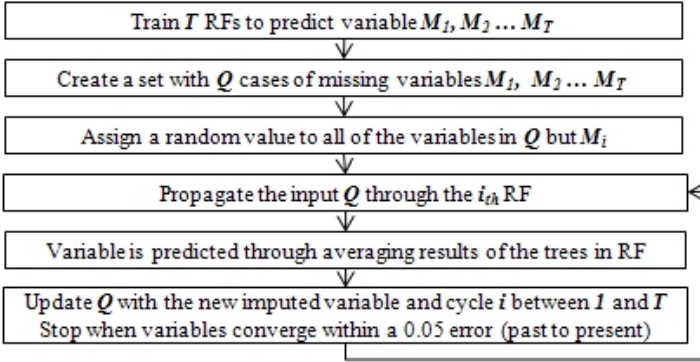


Fig. 1. Flow chart indicating the concept used in imputing multiple missing variables with regression and classification RFs

network predicts a single value, a system of 14 is set up in order to minimise the error between the input and the output in an auto-associative configuration. Each of the ANFIS systems uses subtractive clustering to train, with a training radius of 0.2, 20 training epochs, and a stopping criterion of 0.01.

In the **Random Forest & Auto-associative Neural Network Hybrid** topology, the RF is placed in cascade with an AANN-GA, forming the **RF-AANN-GA**. The RF is used to predict a set of missing variables in an experiment set, and the predictions are recorded. These predictions are then used as limits for the search space of the GA in the AANN-GA system. Since the variable range is 0 - 1, a tolerance of 10 % is added the variable such that the GA has a slightly broader search space. The principle is that by limiting the search space, the AANN-GA will have improved efficiency. A similar principle is successfully applied in [1], in which C4.5 decision trees are used to limit the search space. The AANN-GA and RF have the same structures and parameters as the aforementioned standalone optimised structures.

The **Auto-Associative Neural Network & Random Forest Hybrid** topology combines the AANN-GA system in cascade with an RF, forming the **AANN-GA-RF**. The principle behind the operation is that the RF learns the underlying problems with the AANN-GA system and compensates for them. In order to achieve this, the data are divided into four sets: training; validation; testing and experimental. The training and validation data are used to train and obtain the best model for the AANN-GA using *early stopping* [25]. Thereafter, data are removed from the testing and experimental sets to yield artificially incomplete sets, and these incomplete sets are propagated through the AANN-GA to obtain missing data predictions from the AANN-GA. The testing data and imputed values are made to form a complete set. This testing set is then used as a training set for the RF, with the target being the *original* correct data. In this way, the RF is made to compensate for the error introduced by the AANN-GA. The experimental set is then used to test the RF.

5 Data Evaluation and Preprocessing

Preprocessing of data allows for the data to be of appropriate form for the machine learning paradigms. The data set used is based on a study performed in 2001 for national human immunodeficiency virus (HIV) and syphilis seroprevalence in women attending antenatal clinics in South Africa [26]. The variables contained in the data set are outlined in table 1. Note that the data ranges given are for once the variables have been processed according to the given rules. Gravidity refers to the number of times a woman has been pregnant, and parity the number of times the woman has given birth. Father’s age refers to the age of the father responsible for the current pregnancy. Education is specified as 0 (no education); 1 - 12 (for grades 1 through to 12); or 13 (tertiary education). Province categorises a person in to one of the nine South African provinces, and race categorises a person in to one of six race categories [27].

Table 1. Outline of data set variables (adapted from [27])

Variable	Data Type	Range	Variable Type
Province (location)	Integer	1 - 9	Categorical
Age	Integer	12 - 50	Ordinal
Education	Integer	0 - 13	Ordinal
Gravidity	Integer	1 - 12	Ordinal
Parity	Integer	0 - 9	Ordinal
Father’s age	Integer	12 - 90	Ordinal
HIV status	Binary	0/1	Categorical
Rapid Plasma Reagin (RPR) test status	Binary	0/1	Categorical
Race	Integer	0 - 5	Categorical

Since we are dealing with a real-world study involving missing data, the original data contain inherent errors. There were 16 743 pregnant women involved in the study, of which just under 12 000 instances are regarded as complete and/or valid according to the rules applied. A number of those regarded as outliers contained spurious data or missing data. The problem of missing data is immediately apparent from this statistic. In order to yield a complete set, all fields must be valid as specified according to the ranges indicated in table 1 and in accordance with the logical rules that data cannot be negative and that gravidity cannot be less than parity. The data are preprocessed to ensure this and for normalisation of the data. The categorical data of race and province are binary encoded, since these variables are not ordinal, and non-encoded variables may interfere with the performance of the learning paradigms [27].

6 Comparison and Results

Table 2 indicates the results on testing the missing data prediction ability of the various systems. The results are found by predicting missing data of the

indicated variables, and calculating the percentage of values accurately predicted within specified ranges. MAR and MCAR are not distinguished. Note that the C4.5 AANN-GA results, provided as benchmark results for comparison, may be biased due to the experiments being performed under different conditions, since they are obtained for the appropriate ranges from [1]. The ranges are indicated in the table (for example, age prediction is assessed for percentages of correct prediction within 1, 2, 4, 6 and 10 years).

Testing is performed to determine the best of the techniques specified in section 4. It is evident from the result that the RF and RF hybrids outperform the other methods of missing data prediction. There is significant improvement in the RF from the commonly used AANN-GA method, with an average percentage increase of 7.6 % for the indicated categories. The RF prediction of education increases by an average of 31.2 % when compared with the AANN-GA across the

Table 2. Results of percentage prediction accuracy for given methods within the specified ranges

Quantity	Range (Within)	RF	RF- AANN- GA	AANN- GA RF	AANF- GA	AANN- GA	C4.5, AANN- GA [1]
Age (years)	1	41.1	35.1	40.2	22.0	34.7	-
	2	62.3	55.9	60.8	36.7	54.7	52.3
	4	85.7	83.0	85.0	54.0	81.1	79.4
	6	95.0	92.8	93.9	68.7	90.5	89.6
	10	99.2	98.5	99.0	80.7	96.5	97.9
Education (grades)	0	16.7	19.7	15.4	6.5	5.5	-
	1	53.5	48.1	51.5	18.5	24.3	52.1
	2	76.9	69.4	75.7	34.5	35.8	69.5
	3	88.3	83.7	88.3	46.5	46.4	79.4
	5	93.1	90.8	93.2	70.0	54.8	91.8
Gravidity (Instances)	0	88.0	88.1	88.1	0	88.0	80.4
	1	98.3	98.2	98.2	13.7	98.2	97.1
	2	99.5	99.4	99.4	35.7	99.4	-
	3	99.8	99.7	99.7	67.0	99.8	99.6
	5	100.0	100.0	100.0	95.0	100.0	100.0
Parity (Instances)	0	89.4	87.6	89.5	0	87.9	60.8
	1	98.5	98.3	98.4	21.5	98.2	92.9
	2	99.6	99.5	99.6	52.0	99.4	-
	3	100.0	99.9	100.0	74.0	99.8	89.6
	5	100.0	100.0	100.0	94.0	100.0	97.9
Father's Age (years)	1	28.8	27.9	28.3	3.5	27.7	-
	2	45.7	45.6	46.2	11.5	45.9	41.7
	4	74.1	72.8	73.6	22.5	72.1	68.6
	6	86.3	86.2	86.7	32.0	86.1	82.7
	10	95.3	94.4	95.0	53.0	94.2	93.2

Table 3. Relative computation time taken for the various indicated methods for propagation through 5000 instances with missing data

Method	Training Time (s)	Propagation Time (s)
RF	0.04	0.5
AANF-GA	797.4	50964
AANN-GA	20.7	628.3

specified ranges. The improvement from the AANN-GA to the AANN-GA-RF is a significant one, indicating that the hybrid method of section 4 is working, but the results are comparable to the standalone RF. Furthermore, in the case of the RF-AANN-GA it is observable through experimentation that narrowing the search bounds of the GA improves the performance. Thus, introducing the AANN-GA with larger search bands starts to degrade the performance of the hybrid, indicating that this hybrid’s results are suffering from problems within the AANN-GA and not within the RF. The AANN-GA and AANF-GA perform relatively badly in different aspects: age prediction (for the AANF-GA) and education prediction (for the AANN-GA). The RF performs well in all respects and does not suffer the drawbacks of the other paradigms in predicting age or education.

While the hybrid methods appear to show potential, the computational time trade-off for the use of these methods (due to the cascade with AANN-GAs) is not warranted for the performance improvement. This is especially so in lieu of the relative computation time taken, as indicated in table 3. Note that the study to obtain this table was performed in MATLAB and the programming is therefore not standardised. This result should thus be treated as a basic evaluation. That said, it is to be noted that RF is generally documented as being a relatively fast machine learning tool [10,13,14,24], and this is reflected by the table. It is evident that there are vast improvements in the computational efficiency of the RF algorithm.

The HIV status is predicted by an RF classifier and the results are presented in table 4. The other configurations were also used to predict HIV status. The AANN-GA obtains prediction accuracy of 64.2 % with an F-measure of 0.43. This is not, however, discussed further in this work. The classification results obtained for the RF are lower than those found in [9].

Table 4. HIV Status Prediction Confusion Matrix for RF Classifier run on experiment set

Confusion Matrix	Predicted Negative	Predicted Positive	Percentage Error (%)
Actual Negative	2902	1732	37.4
Actual Positive	449	875	33.9

7 Discussion and Recommendations for Future Work

This work is extended in a missing data impact assessment [28]. It is notable that the AANN-GA method is relatively computationally expensive and are shown to be outperformed when compared with RFs used for the purpose of estimating missing data for this data set [8,9]. In this paper, either regression or classification RFs are used. RF does, however, have built in functionality to estimate missing data through computing terminal node proximities [14]. This functionality was not tested within this work, but RFs implementing this method may show further improvement. The RF HIV classifier does not perform well when compared with classifiers using the same data [9]. The results obtained from the AANF-GA are relatively poor. Despite the fact that ANFIS struggles with high dimensionality data, training was possible through subtractive clustering. It was not feasible, however, to train the ANFIS system with grid partitioning unless variables were removed, and this impacts on the standardisation of the comparison with the other systems.

8 Conclusion

Missing data causes significant information loss in studies as information is wasted and insight cannot be gained into the underlying causes of the absence. Through the use of data resulting from an HIV seroprevalence survey, this paper investigates and compares five machine learning paradigms in order to impute missing data: RFs, AANN-GA, AANF-GA, RF-AANN-GA and AANN-GA-RF. It is evident from the presented results that the RF algorithm as a regression system to impute the missing data outperforms the other paradigms investigated for the studied data set. This is true for both computation time and computation accuracy, with RFs outperforming the other paradigms by up to 32 % on average for some categories.

References

1. Ssali, G., Marwala, T.: Computational Intelligence and Decision Trees for Missing Data Estimation. In: Proceedings of the International Joint Conference on Neural Networks, part of the IEEE World Congress on Computational Intelligence, WCCI 2008, IJCNN, pp. 201–207. IEEE, Los Alamitos (2008)
2. Horton, N.J., Kleinman, K.P.: Much Ado About Nothing: A Comparison of Missing Data Methods and Software to Fit Incomplete Data Regression Models. *The American Statistician* 61(1), 79–90 (2007)
3. Markey, M.K., Tourassi, G.D., Margolis, M., DeLong, D.M.: Impact of Missing Data in Evaluating Artificial Neural Networks Trained on Complete Data. In: *Computers in Biology and Medicine*, vol. 36, pp. 517–525. Elsevier, Amsterdam (2006)
4. Little, R.J., Rubin, D.B.: *Statistical Analysis with Missing Data*. John Wiley & Sons, Chichester (2002)
5. Ziegler, M.L.: Variable selection when confronted with missing data. PhD thesis, University of Pittsburgh (2006)

6. Fogarty, D.J.: Multiple Imputation as a Missing Data Approach to Reject Inference on Consumer Credit Scoring. *Intersat* 41(9) (2006)
7. Yuan, K.H., Bentler, P.M.: Three likelihood-based methods for mean and covariance structure analysis with non-normal missing data. *Sociological Methodology*, 165–200 (2000)
8. Nelwamondo, F.V., Mohamed, S., Marwala, T.: Missing data: a of neural network and expectation maximisation techniques. *Current Science* 93(11), 1514–1521 (2007)
9. Betechuoh, B.L., Marwala, T., Tettey, T.: Autoencoder Networks for HIV Classification. *Current Science* 91(11), 1467–1473 (2006)
10. Biau, G., Devroye, L., Lugosi, G.: Consistency of Random Forests and Other Averaging Classifiers. *Journal of Machine Learning Research* 9, 2015–2033 (2008)
11. Masisi, L., Nelwamondo, F.V., Marwala, T.: The Effect of Structural Diversity of an Ensemble of Classifiers on Classification Accuracy. In: *IASTED International Conference on Modelling and Simulation (Africa-MS)* (2008)
12. Qi, Y., Klein-Seetharaman, J., Bar-Joseph, Z.: Random Forest Similarity for Protein-Protein Interaction Prediction from Multiple Sources. In: *Pacific Symposium on Biocomputing*, vol. 10, pp. 531–542 (2005)
13. Ho, T.K.: Random Decision Forests. In: *ICDAR 1995: Proceedings of the Third International Conference on Document Analysis and Recognition*, vol. 1 (1995)
14. Breiman, L., Cutler, A.: Random Forests. Department of Statistics, University of California Berkeley (2004)
15. Brencce, J.R., Brown, D.E.: Improving the Robust Random Forest Regression Algorithm (2006)
16. Engelbrecht, A.P.: *Computation Intelligence, an Introduction*. John Wiley & Sons, Ltd., Chichester (2002)
17. Haykin, S.: *Neural Networks: A Comprehensive Foundation*. Macmillan, New York (1994)
18. Jang, J.S.R., Gulley, N.: *Fuzzy Logic Toolbox*. The MathWorks Inc. (1997)
19. Abraham, A.: Neuro fuzzy systems: Sate-of-the-art modeling techniques. In: Mira, J., Prieto, A.G. (eds.) *IWANN 2001*. LNCS, vol. 2084, pp. 269–276. Springer, Heidelberg (2001)
20. Jang, J.S.R.: Neurofuzzy Modelling and Control. *Proceedings of the IEEE* 83 (1995)
21. Jang, J.S.R.: Input Selection for ANFIS Learning. In: *Proceedings of IEEE International Conference on Fuzzy Systems* (1998)
22. Wong, H.: Genetic Algorithms. *Surprise 96 Journal*. Imperial College of Science Technology and Medicine (1996)
23. Zalzal, A.M.S., Fleming, P.J.: Genetic Algorithms in Engineering Systems. *IET* (1997)
24. Breiman, L.: Random Forests. *Machine Learning* 45, 5–32 (2001)
25. Yuan, Y., Lorenzo, R., Andrea, C.: On Early Stopping in Gradient Descent Learning. *Constructive Approximation* 26(2), 289–315 (2007)
26. Ntsaluba, A.: Summary Report: National HIV and Syphilis Sero-prevalence Survey of Women Attending Public Antenatal Clinics in South Africa, 2001 Department of Health, South African Government (2001)
27. Mistry, J., Nelwamondo, F.V., Marwala, T.: Investigation of Autoencoder Neural Network Accuracy for Computational Intelligence Methods to Estimate Missing Data. In: *IASTED International Conference on Modelling and Simulation* (2008)
28. Pantanowitz, A., Marwala, T.: Evaluating the Impact of Missing Data Imputation. LNCS (LNAI), vol. 5678. Springer, Heidelberg (to appear, 2009)

Ubiquitous Middleware Using Mobility Prediction Based on Neuro-Association Mining for Adaptive Distributed Object System*

Romeo Mark A. Mateo¹, Malrey Lee², and Jaewan Lee¹

¹ School of Electronic and Information Engineering, Kunsan National University,
68 Miryong-dong, Kunsan, Chonbuk 573-701, South Korea
{rmmateo, jwlee}@kunsan.ac.kr

² School of Electronics and Information Engineering, Chonbuk National University, 664-14,
DeokJin-dong, Jeonju, Chonbuk 561-756, South Korea
mrlee@chonbuk.ac.kr

Abstract. This paper proposes a ubiquitous middleware for the adaptive distributed object system to consider the mobility support of application services in distributed environment. To ensure the seamless connectivity of a moving client, a prediction based on association mining of mobility patterns is used to perform the replication of services in the next base station. The proposed neuro-association mining is based on Apriori to generate rules for prediction and these rules become nodes for the structure of multilayer perceptron (MLP) to classify the next location and replicate the resource currently serving the mobile client. We present our simulation environment for the ubiquitous middleware and a classification performance evaluation where the proposed algorithm shows more accurate and provide more comprehensive structure compared to neural network based classifiers.

1 Introduction

The concept of efficient communication and coordination of components within the computer networks in distributed systems are still reflected to current technologies like ubiquitous, mobile and peer-to-peer systems. Classical distributed system primary goal is to share effectively data or resources within the network. The common challenges of distributed system designers to effectively distribute information are heterogeneity of system components, scalability of resources, failure handling, concurrency of components, and transparency of services [1]. The current researches in distributed system focus on efficient search of objects [2,3], object group modeling [4], dynamic replications [5] and load distribution of objects [6,7]. The effective management of resources is critical for distributed system components. The access of each resources and services in distributed systems are transparent to client where it is not aware of the physical location of the resources because of the fully connected

* This paper is supported by Industry and University Research Consortium of Small and Medium Business Administration (SMBA).

networks and the delay time on accessing resources are minimal. This contradicts to the concept of wireless environment. Because of limited connectivity in wireless transmission, the system should provide a simple request and reply procedure. Also, location-aware is provided to mobile users where context of location and available services on that base station are exposed to user for quality of service (QoS) and because of this, services from other base station cannot be provided. At this reason, we present a design to allow the functions of ubiquitous technology in a distributed object environment in terms of accessing resources transparently. Moreover, the quality of service is obtained by providing fast responses on requests and this can be achieved by replicating the services.

In replication schemes, the client requests are distributed to other replicated services which minimize the queuing of requests providing a faster response. However, there is a delay on performing the replication copying and initializing the object. Predicting the next replication can minimized the delay time of object replication. Data mining is a method of extracting rules from huge data and can be used in prediction [8]. Association rule mining method determines frequent patterns from the data transaction where these frequent patterns analyzed as the trends of events or transactions. The issue of replication are when does a replication of object is needed and where should the replication takes place. Association rule mining can be used to predict the needed object to be replicated in the system. This method is used in our proposed system to extract rules from the data and use these rules to adapt the new constraints of a system.

In this paper, we present a design of a ubiquitous middleware for our previous work in adaptive distributed object system [9] named as intelligent distributed framework (IDF) to support ubiquitous technology. Autonomous agents are included in the framework which represents application components. The IDF considers services for distributed programs and hardware, and services for the limited resources by means of the proposed ubiquitous middleware. The adaptive module of the IDF includes our proposed prediction technique to predict the next location of a mobile user currently accessing resource and replicate that resource on the location. The prediction is done by using our proposed Neuro-Apriori algorithm which provides comprehensive rules and accurately predicts the next location to provide the object replication.

2 Related Works

2.1 Ubiquitous Middleware

The current developments in ubiquitous computing focus on smart information acquisition which uses the wireless environment as medium of communication. Ubiquitous computing concepts have emerged not only providing information wirelessly but also automates services based on the context information from user profiles and the environment. Large amount of information transfer and adding additional features are research challenges of ubiquitous computing. Commonly, the designs of interactions of users to the system are simple to prevent unnecessary overheads and delays. In contrast, a fully connected distributed system provides a

technology that hides complex interactions. Because of limited connectivity in wireless environments, it is hard to realize the usual techniques of distributed system in ubiquitous networks. Current middleware technologies for mobile and ubiquitous support are limited in providing transparency of services to mobile users which are tackled by some researches [10,11]. The middleware for mobile computing based on mobile agents (MA) is proposed by Belevista et. al [10]. With the use of autonomous mobile agent, users can access services even if the terminal disconnects because these agents deliver the results upon reconnection. The HOMEROS is proposed [11] which allow high flexibility by adopting a hybrid-network model and dynamically configurable reflective ORB to provide flexibility to applications. The hybrid-network model is designed to efficiently manage enormous resources, context, location, and various services. Our proposed middleware support the mobility of programs and scalable to additional components. It supports agent protocols for the basic communication and service protocols which adjust and serve the agents through their task in a transparent manner.

2.2 Mobility Prediction

Mobility prediction is commonly used to optimize the location management in cellular environments. An adaptive location prediction is presented in [12] where it uses a hierarchy of location area and estimates the location probabilities of each mobile user. In [13], data mining approach for the location prediction used to allocate resources in a PCS network where it uses sequential mining. There are three phases presented. First is mining the user mobility pattern, second is extracting the mobility rules and the last is the mobility prediction. The predicted movement can then be used to increase the efficiency of location management. Also, using this technique can effectively allocate resources to the most probable-to-move cells instead of blindly allocating excessive resources in the cell-neighborhood of a mobile-user. Association mining is also used for mobility prediction. Association mining method determines frequent patterns from the data transaction where these frequent patterns analyzed as the trend of the events or transactions. In our previous research, we used the association mining rules and basing on these rules, agent predicts the future mobile location [14]. However, the accuracy of the association rules is limited on its procedure of support counting of patterns. Classification method is a form of data analysis that can be used to extract models describing important data classes or predict future data trends. The neural network is one of the most commonly used classification techniques. However, it is difficult to interpret the structure on how the process provides the accurate result. But most researches already accepted the black box characteristic of neural-networks for their applications. Our proposed algorithm for the mobility prediction is based on the association mining and to have an accurate reading, the rules are pruned on the network structure of a multilayer perceptron.

3 Ubiquitous Middleware for Adaptive Distributed Object System

Intelligent models are mostly topics of research where recent approaches are solutions to the previous problems experienced in classical systems and tackle the issue of

solving future constraints. This study presents the intelligent distributed framework (IDF) which implements scalable system and addresses the use of intelligent approaches for management of adaptive distributed object in ubiquitous environment illustrated in Figure 1. The proposed framework which is based on our previous work [9] consists of three layers of software and hardware components: ubiquitous layer, intelligent middleware layer and infrastructure layer.

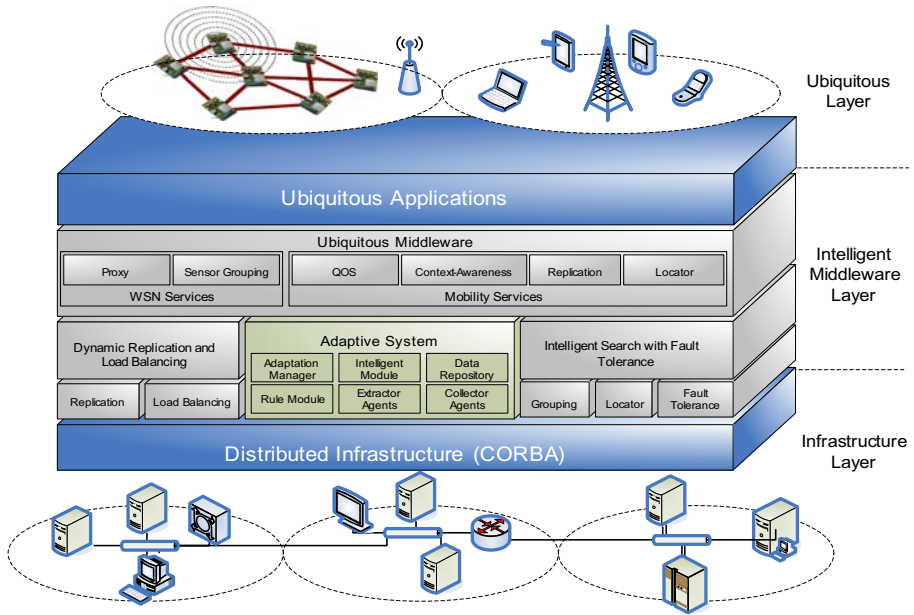


Fig. 1. Intelligent Distributed Framework: an adaptive distributed object system

An integration of ubiquitous technology is the improvement from the previous work [9]. The IDF considers the services for distributed programs and hardware, and services for the limited resources by means of ubiquitous middleware. Mobile devices are equipped with context-aware services. Moreover, location transparency is implemented in an ORB-like interaction where it can request for services to other location. We based the design of IDF on CORBA technology where the communication of clients and services are using the ORB technology. On the other point of view, while ORB technology is desirable to implement the transparent services, it is limited to process in wireless sensors and embedded devices. We try to tackle this issue in the ubiquitous layer. In this layer, hardware components are mobile devices, wireless sensors and embedded devices while software components are agents and primitive commands to communicate with the services. The mobile and wireless devices are classified into two; 1) object-oriented and 2) proxy controlled devices. The object oriented devices have enough processing power to host objects in able to interact with the distributed object system seamlessly while the proxy controlled has no capability to host an object but only provide a proxy within the system to access its primitive controls. We present the ubiquitous middleware

which is another sub layer between the intelligent middleware layer and ubiquitous layer that provide services to support mobility. The intelligent middleware layer acts as the primary middleware where services are transparently operating on serving the clients. Interaction of clients from ubiquitous layer and services are handled by this layer. The service functions are influenced by the adaptive module in able to interact correctly with services in goal of optimizing the performance of tasks. In our proposal, we used the term “influence” as a method of knowledge transferring to another service to change its requirements based on analysis of the adaptive system. The infrastructure layer is networks of different computers that are powerful of storing data and processing transactions within the system, like PCs and servers. These computers are communicating in wired network system to provide massive information exchange between the systems. The intelligent middleware layer presents an overlay communication of software components that are hosted by hardware devices in the ubiquitous and infrastructure layers. The proposed system consists of components that perform the information processing and used this relevant information for optimal configuration. Figure 2 shows the layered interaction of the proposed model to the Intelligent Distributed Framework and the ubiquitous middleware.

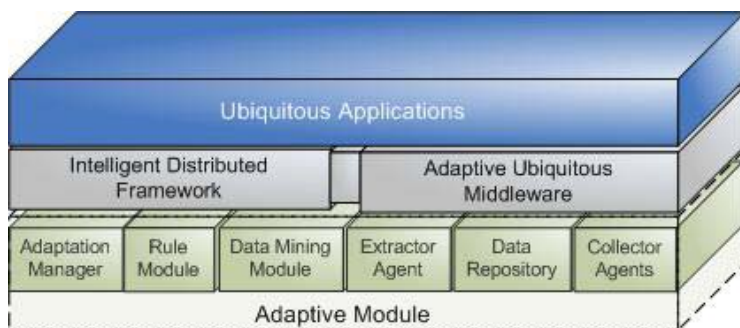


Fig. 2. Adaptive Module providing integration between the IDF and ubiquitous middleware

The adaptive module interacts with the services of IDF and ubiquitous middleware by adaptation manager to inform the services with the new information. This adaptive scheme optimizes the performance of the task by inter-relating the services task and analyzing the result. The following are the components of the adaptive system:

- **Adaptation manager** – main component of the adaptive model that influences the functions of services. The rule information that is needed by the services is stored in the rule module. These rules are generated by the extractor agent using the data mining module. Based on its, the adaptive manager informs the services to change its requirements. Simultaneous sending of messages is done to each service that is affected by the system changes.
- **Rule module** – this module organizes and stores rules extracted by the extractor agents. Rule information is classified based data mining algorithm.

- **Data mining module** – provides various data mining algorithms that are used by the extractor agent to extract information from data repository. This module is configured by algorithm experts to implement the appropriate data mining procedure on a specific adaptation task.
- **Extractor agent** – this agent is used to extract the rules from the data repository. The information from the services like the number of current replicas and the number of clients is some variables to process the rule generation. Extractor agent chooses the data mining algorithm from data mining module to suite the rule generation and provides needed results.
- **Data repository** – data storage for collected data from different transactions and events of services. The collected data is used by the extractor agent to process the rule extraction.
- **Collector agent** – a mobile agent circulates through the system to gather information and store to a data repository. Transaction results and data events are data that this agent collects and stores the data in the data repository.

4 Dynamic Replication Based on Neuro-Apriori Algorithm

The dynamic replication scheme uses association rule mining to construct a pattern from transactions of object request to predict the next location where an object is needed for replication. In this paper, we refer objects as resources. Apriori algorithm was used to perform association rule extraction. Classification association rule mining (CARM) [14] is used which provide a class outcome to a rule. The data is provided with input attributes (x_i) and a class (c_j) or outcome. There are two steps of Apriori algorithm which use to generate the candidate pattern and choose the frequent pattern. First is the join step which finds P_k , a set of candidate k -itemsets by joining P_{k-1} with itself. Second is the prune step where C_k is generated as superset of P_k , and all of the frequent k -itemsets are included in C_k . Choosing the frequent pattern is based on confidence value. The support count determines each pattern frequency shown in Equation 1. After determining the support count, the confidence is determined by getting the ratio of support count and total number of data in Equation 2. After generating all patterns with its respective confidence, the patterns are used in comparing the current object replication patterns. These patterns R_k are shown in Equation 3.

$$s_count (A \rightarrow B) = P(A \cup B) \quad (1)$$

$$confidence (A \rightarrow B) = P(B | A) = \frac{s_count (A \cup B)}{s_count (A)} \quad (2)$$

$$R_k = \text{If } L_1 \text{ and } \dots, \text{ and } L_k \text{ then } L_j \quad (3)$$

After the execution of the Apriori algorithm, all rules are used for the rule nodes (r_k) which are hidden layers and class nodes (z_k) of the network structure of proposed algorithm. The multilayer perceptron (MLP) is used in data mining for classification techniques. In our proposed model, the MLP is consisted of three layers of processing

nodes an input layer which accepts the input variables used in the classification procedure, rule layers, and an output layer with one node per class. Each r_k are connected to x_i by a weight (w_{ji}) for each antecedent a rule consists. The classes of the MLP are labeled by the outcome from R_k . Each r_k connects to its corresponding outcome by a weight (w_{kj}). All weights are initially set with random values. Each data pattern is trained using the current configuration by the back propagation algorithm. The calculation of inputs and weights of the MLP is shown in Equation 4.

$$z_k = f\left(\sum_{j=1}^{nH} w_{kj} f\left(\sum_{i=1}^d w_{ji} x_i + w_{j0}\right) + w_{k0}\right) \quad (4)$$

$$J(w) = \frac{1}{2} \sum_{k=1}^c (t_k - z_k)^2 = \frac{1}{2} \|t - z\|^2 \quad (5)$$

The back-propagation algorithm is a gradient descent optimization procedure which minimizes the mean square error between the network's output and the desired output for all input patterns P . The outputs are compared to the target values and determine the differences. The error is minimized by a function when the network output match the desired outputs where the weights are changed. The least mean square (LMS) for two layer nets is shown in Equation 5, where t and z are the target and the network output vectors of length c and w represents all the weights in the network. After the training, stopping at a certain criteria like epoch counts, the structure of the MLP with the labeled rule patterns from the association mining is used for the classification of data.

5 Experimental Evaluation

The intelligent distributed framework uses Visibroker 7.0 to implement the CORBA-based object grouping and JADE to implement the agent interaction. The Neuro-Apriori algorithm is coded in Java and integrated to the adaptive module. The operating system (OS) platforms used for the experiment were Windows OS, Red Hat Linux and Sun Solaris 8 to simulate the heterogeneity of system. The proposed method was evaluated by determining the client delay time and classification accuracy.

5.1 Generation of the Rule Nodes

We execute the proposed Neuro-Apriori algorithm to process the user mobility pattern and produce the mobility rules to a specific user. The user has a 322 attributes in the database and we select the first 6 hours of its movement. Time period indicated by 1 to 6 values and locations are labeled NX where X are 10 different locations. The preprocess method only select the latest 100 tuples of data and ignoring the data with missing data values. The selection of many tuples will increase the time processing so that is why we chose only limited records. After selecting, the configuration of the proposed algorithm is set with the minimum support (S) to 50 out of 100, which means that the possibilities of the pattern will most likely 50 percent and high of the pattern can be found in the data. The result produces 541 rules. These rules are used in the structure of Neuro-Apriori. Table 1 presents the result of the rules generated with its corresponding confidence (C).

Table 1. Association rules showing only 5

541 rules of user A mobility, showing only 5	S	C
1=N10 2=N1 4=N9=> 6=N2	0.69	0.99
1=N10 2=N1 4=N9 5=N2=> 6=N5	0.685	0.99
1=N10 2=N1 3=N6=> 4=N9 6=N5	0.655	0.98
1=N10 2=N1 => 6=N2	0.70	0.97
1=N10 => 6=N9	0.76	0.95

5.2 Performance of the Proposed Algorithm

After generating rules, these become rule nodes of the proposed classifier and trained by the back propagation. Each rule has weight connection to input layers by its antecedents and also to the output layer by its class or rule outcome. The trained structure of proposed algorithm is used to support the intelligent search mechanism of the IDF by providing a seamless connection to a service via replication. The replication and cloning service implements the location prediction based on the proposed Neuro-Apriori algorithm.

We compare the accuracy of the proposed algorithm in MLP and RBF classifiers shown in Figure 3. Epoch is the training count of neural network-based classifiers that determines the speed of the algorithm to classify. The accuracy rate of the proposed algorithm is comparable to the MLP illustrated in Figure 3. However, because of the association rules in the structure of the Neuro-Apriori, it is easier to understand with its comprehensive rules. The structure of MLP and RBF are more sophisticated using several weight connection and nodes which has no meaning compared to Neuro-Apriori algorithm.

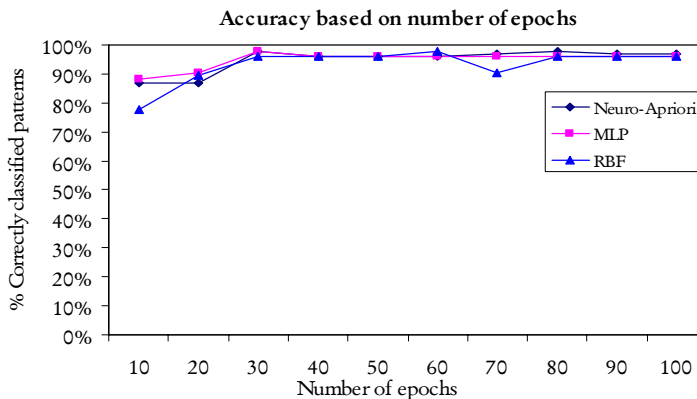


Fig. 3. Classification accuracy of proposed neuro-Apriori and other neural network-based classifiers based on number of epochs

6 Conclusions and Future Work

The integration of intelligent system contributes to the efficiency of the system management and effectiveness of providing quality of service to clients. Moreover, robust implementation of system considering the limited resources in mobile environments is a timely issue. In this paper, a ubiquitous middleware for the intelligent distributed framework, our previous work, is proposed to consider ubiquitous and limited resource devices in the adaptive distributed object system. The adaptive module of IDF is designed to utilize several data mining methods for object grouping, intelligent searching and dynamic replication schemes. The adaptive module uses a prediction technique of the next location of the user to dynamically replicates the resources that a users currently accessing. The prediction is done by using our proposed Neuro-Apriori algorithm, a combination of Apriori and MLP, which provides comprehensive rules and accurately predicts the next location to provide the object replication.

The contribution of this paper from the previous research [9] is the efficiency of seamless connection of clients to resources and the future works includes more implementation on the components from the mobility middleware.

References

1. Coulouris, G., Dollimore, J., Kinderberg, T.: *Distributed Systems Concepts and Design*, 4th edn., pp. 1–25. Addison-Wesley, Reading (2005)
2. Badidi, E., Keller, R.K., Kropf, P.G., Van Dongen, V.: The Design of a Trader-based CORBA Load Sharing Service. In: Proc. of the 12th International Conference on Parallel and Distributed Computing Systems, pp. 75–80 (1999)
3. Van Steen, M., Ballintijn, G.: Achieving Scalability in Hierarchical Location Services. In: Proc. of the 26th International Computer Software and Applications Conference (2002)
4. Felber, P., Guerraoui, R.: Programming with Object Groups in CORBA. *IEEE Concurrency* 8(1), 48–58 (2000)
5. Felber, P., Guerraoui, R., Schiper, A.: Replication of CORBA objects. In: Krakowiak, S., Shrivastava, S.K. (eds.) *BROADCAST 1999*. LNCS, vol. 1752, p. 254. Springer, Heidelberg (2000)
6. Othman, O., O’Ryan, C., Schmidt, D.C.: The Design of an Adaptive CORBA Load Balancing Service. *IEEE Distributed Systems Online* 2(4) (2001)
7. Kwok, Y.K., Cheung, L.S.: A New Fuzzy-decision based Load Balancing System for Distributed Object Computing. *Journal of Parallel and Distributed Computing* 2(64), 238–253 (2004)
8. Han, J., Kamber, M.: *Data Mining Concepts and Techniques*, 2nd edn., pp. 1–38. Morgan Kaufman, San Francisco (2006)
9. Mateo, R.M.A., Yoon, I., Lee, J.: Data-Mining Model Based on Multi-agent for the Intelligent Distributed Framework. In: Nguyen, N.T., Jo, G.-S., Howlett, R.J., Jain, L.C. (eds.) *KES-AMSTA 2008*. LNCS (LNAI), vol. 4953, pp. 753–762. Springer, Heidelberg (2008)
10. Bellavista, P., Corradi, A., Stefanelli, C.: Mobile Agent Middleware for Mobile Computing. *Computer* 34(3), 73S–81S (2001)

11. Han, W., Yoon, Y.B., Youn, H.Y., Cho, W.D.: A New Middleware Architecture for Ubiquitous Computing Environment. In: Proc. of STFEUS, pp. 117–121 (2004)
12. Das, S., Sen, S.: Adaptive Location Prediction Strategies Based on a Hierarchical Network Model in a Cellular Mobile Environment. *The Computer Journal* 42(6) (1999)
13. Yavas, G., Katsaros, D., Ulusoy, O., Manolopoulos, Y.: A Data Mining Approach for Location Prediction in Mobile Environments. *Data & Knowledge Engineering* 54, 121–146 (2005)
14. Ali, K., Manganaris, S., Srikant, R.: Partial Classification using Association Rules. In: Proceedings of KDD 1997, pp. 115–118. AAAI Press, Menlo Park (1997)

A Growing Algorithm for RBF Neural Network*

Han Honggui and Qiao Junfei

College of Electronic and Control Engineering, Beijing University of Technology, Beijing,
China

Rechard112@emails.bjut.edu.cn

Abstract. This paper presents a growing algorithm to design the architecture of RBF neural network called growing RBF neural network algorithm (GRBF). The GRBF starts from a single prototype randomly initialized in the feature space; the whole algorithm consists of two major parts: the structure learning phase and parameter adjusting phase. In the structure algorithm, the growing strategy is used to judge when and where the RBF neural network should be grown in the hidden layer based on the sensitivity analysis of the network output. In the parameter adjusting strategy, the whole weights of the RBF should be adjusted for improving the whole capabilities of the GRBF. In the end, the proposed GRBF network is employed to track non-linear functions. The computational complexity analysis and the results of the simulations confirm the efficiency of the proposed algorithm.

Keywords: Growing algorithm, RBF neural network (RBFNN), Structure design, Output model.

1 Introduction

Because of the input layer and output layer of the RBFNN are decided in advance, the real dynamic part of the RBF structure is the hidden layer. In recent years, there are two main methods for the dynamic structure of the RBF neural network: One is growing, and another is pruning.

In the growing methods, A. Esposito *et al.* [1] proposed a growing RBF based on the evolutionary optimization strategy, however, it was computationally very expensive to implement and it is also well known that this algorithm suffered the slow and premature convergence problems. To save computational time of the RBF neural network learning, a prior method called clustering techniques has been proposed [2] to the center location. But the clustering method is an un-resolved problem. This algorithm was limited by the intrinsic flaws. Orr [3] proposed the regularized forward selection (RFS) algorithm for RBF networks, which combines forward subset

* This work was supported by the National 863 Scheme Foundation of China under Grant 2007AA04Z160; National Science Foundation of China under Grant 60674066, 60873043; Ph.D. Programs Foundation from Ministry of Chinese Education under Grant 200800050004; Beijing Municipal Natural Science Foundation under Grant 4092010.

selection and zero-order regularization and achieves better generalization. Generally speaking, this subset selection method had several major disadvantages. The mostly disadvantage is that in order to increase the chance of obtaining a satisfactory RBF network, Orr's method [3] has to use a very large set of candidate RBF nodes of different centers and widths. Sometimes, this is computationally too expensive to implement, since all the candidate RBF nodes have to be stored for batch operations and the number of all candidates will increase exponentially as the search space dimension increases.

On the other side, Yingwei *et al.* [4] introduced a pruning strategy based on the relative contribution of each hidden neuron to the overall network output to reduce the complex of the RBF neural network. The resulting neural network was minimal in theory by the presented method. Other methods for pruning in RBF networks have been proposed by Salmerón *et al.* [5], and Hao *et al.* [6]. Comparatively, the growing methods often require much less computation than the pruning methods. A major problem with pruning methods is the possible existence of under-utilization problem. In other words, these methods require additional computational efforts. Quite a large number of parameters or variables need to be preset and training data need to be stored and reused for pruning purposes.

Different from existing methods in RBF neural network construction, this paper proposes a growing RBF neural network algorithm (GRBF), which performs simultaneous network growing and parameter optimization within an integrated analytic framework. The rest of this paper is organized as follows. In Section 2, the sensitivity analysis (SA) of model output for the RBF neural network is introduced. Section 3 gives brief analyses of the growing method. And then this new algorithm is used to decide hidden nodes for the RBF neural network. Meanwhile, the parameter adjusting algorithm is described. It is shown that our proposed method is benchmarked against well-known dynamic RBF algorithms in Section 4. To demonstrate the superior performance of our GRBF algorithm, this GRBF algorithm is used to track the non-linear functions. The conclusions are given in Section 5.

2 The Sensitivity Analysis (SA) of Model Output for RBFNN

A thorough description of sensitivity analysis methods can be found in [7]. The most common SA is sampling-based. There are several steps to conduct SA. The following steps can be identified as (the details can be found in [7]):

- Step 1) Define the model, its input factors and output variable.
- Step 2) Assign probability density functions or ranges of the variation to each input factor.
- Step 3) Generate an input matrix through sampling design.
- Step 4) Evaluate the output.
- Step 5) Assess the influences or relative importance of each input factor on the output variable.

Sensitivity analysis (SA) is an available tool [8]-[9] which may be used to study the behavior of a system, or a model, and to ascertain the contribution ratio of the outputs depending on each or some of the input parameters. Among the SA methods, quite

often they are identified almost as a mathematical definition, with a differentiation of the output respecting to the input. For this reason, quantitative measure of sensitivity, such as the EFAST method [10] is described as follows:

$$S_h = \frac{\text{Var}_h[E(Y|Z_h = \alpha_h)]}{\text{Var}(Y)} \quad (1)$$

where, Z_h denotes an input factor, $h=1,2,\dots,p$, and Z_h represents the output value of the hidden nodes in the RBF neural network in this paper. Y is the model response, $E(Y|Z_h = \alpha_h)$ is the expectation of Y conditional on a fixed value α_h of Z_h and the variance Var_h is taken over all the possible value of Z_h . The ratio S_h represents the main effect. It is called the first-order index in the SA terminology. Thus, the main effect of a factor represents the average effect of that factor on the response or conversely these methods allow the computation of that fraction of the variance of a given model output which is due to each input factor.

The model is additive when the response is nonlinear but interactions are negligible. In that case, the main effects are the suitable indexes for the sensitivity analysis of model output [11].

3 The Growing Method for Selecting Hidden Nodes of RBF NN

Considering the intrinsic structure of RBF neural network, if the RBF neural network consists of assured inputs, certain hidden nodes and outputs, it can only adjust the position of the center ν , the width of the centers δ and the weights w which are in the hidden layer or connecting the hidden nodes and the output nodes. In this paper, the single response relationship between the hidden neurons and the output of the RBF is discussed. We state that the relevance of a hidden node is related to its influence on the RBF response. This is the key idea of the method proposed to determine the optimal architecture for the RBF. The parameters w of the hidden nodes are the input values of the Growing algorithm based on the sensitivity analysis (SA). And this SA is based on the Fourier decomposition of the variance in the frequency domain.

3.1 Selecting Hidden Nodes

In the following, a generic model is assumed to describe a neural network system. The model is represented by a mapping f (a deterministic or stochastic function) which relates the inputs domain to the output space:

$$Y = f(Z_1, Z_2, \dots, Z_p) = \sum_{i=1}^p \beta_i Z_i \quad (2)$$

The input factors (Z_1, Z_2, \dots, Z_p) are supposed to be the variables described by the parameters w of the RBF hidden nodes, $Z_1 = w_{1\bullet}, Z_2 = w_{2\bullet}, \dots, Z_p = w_{p\bullet}$, p is the number of the nodes in the output layer; $w_{i\bullet} = [w_{i1}, w_{i2}, \dots, w_{im}]^T$, m is the number of the nodes in the output layer. Y is taken to be a scalar even in the application we

shall consider each output variable in turn. Based on the EFAST method, the polynomial expansion can be described again. The range of the factor Z_h is $[a_q, b_q]$, so the Z_h performances as follows:

$$Z_h^{(q)} = (b_q + a_q / 2) + (b_q - a_q / 2) \sin(\omega_h s^{(q)}) \tag{3}$$

where, $s^{(q)} = 2\pi n / N$, ω_h is the frequency, q is the simulation number, N is the total simulation number. If $Z_h^{(q)}$ is straightforward to note $z_h^{(q)} = \sin(\omega_h s^{(q)})$ and that the formula (2) should be as:

$$Y^{(n)} = Y_0 + \sum_{i=1}^p \beta_i \sin(\omega_i s^{(q)}) + \sum_{i=1}^p \sum_{j=1}^p \beta_{ij} \sin(\omega_i s^{(q)}) \sin(\omega_j s^{(q)}) + \dots \tag{4}$$

Based on this formula, the linear effect of Z_h corresponds to the Fourier amplitude at the fundamental frequency ω_h . As s varies, all the factors oscillate at the corresponding driving frequency ω_h and their range is systematically explored. For the EFAST method [11], a parametric representation of the form is often used.

$$Z_h(s) = \frac{1}{2} + \frac{1}{\pi} \arcsin(\sin(\omega_h s)), \quad h = 1, 2, \dots, p \tag{5}$$

This transformation allows a better coverage of the factors' apace since it generates samples that are uniformly distributed in the range $[0, 1]$.

Just for Fourier amplitudes, the p -factor model can be described as follows:

$$f(s) = f(Z_1(s), Z_2(s), \dots, Z_p(s)) \tag{6}$$

where, the range of s is $[-\pi, \pi]$, so the expanded in a Fourier series of the form:

$$f(s) = \sum_{j=-\infty}^{\infty} (A_j \cos(\omega_j s) + B_j \sin(\omega_j s)) \tag{7}$$

The Fourier coefficients are defined as:

$$A_j = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(s) \cos(\omega_j s) ds, \quad B_j = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(s) \sin(\omega_j s) ds; \text{ the range of } s \text{ is } [-\pi, \pi].$$

Therefore, N equally spaced sample points are required to perform the Fourier analysis. N represents the sample size and coincides with the number of model evaluations. Based on the Fourier translation, the variance D_y can be computed as:

$$D_y = Var(Y) = 2 \sum_{k=1}^{+\infty} (A_k^2 + B_k^2) \tag{8}$$

The portion of the variance of Y explained by Z_h alone is:

$$D_y = \text{Var}_{Z_h} [E(Y|Z_h)] = 2 \sum_{k=1}^{+\infty} (A_{k\omega_h}^2 + B_{k\omega_h}^2) \quad (9)$$

where, $A_{k\omega_h}$ and $B_{k\omega_h}$ denote the Fourier coefficients for the fundamental frequency and its higher harmonics $k\omega_h$.

In the EFAST approach, the number of simulation runs represents the sampling frequency and, to meet the Nyquist criterion, it must equal to:

$$N = 2M\omega_{\max} + 1, (\omega_{\max} = \max(\omega_i)) \quad (10)$$

So the variance $\text{Var}(Y)$ can be evaluated in the frequency domain through the following relationship:

$$\text{Var}(Y) = 2 \sum_{\omega=1}^{\frac{(N-1)}{2}} (A_{\omega}^2 + B_{\omega}^2) \quad (11)$$

A_{ω} and B_{ω} denote the Fourier coefficients at frequency ω .

In fact, based on this analysis the total sensitivity index ST_h can be shown as:

$$ST_h = \frac{(A_{\omega}^2 + B_{\omega}^2)}{\sum_{\omega=1}^{\frac{(N-1)}{2}} (A_{\omega}^2 + B_{\omega}^2)} \quad (12)$$

The advantages of the ST_h make the judgments in this paper much better than the S_h . Based on the former analysis of ST_h , the main steps of the proposed approach for selecting hidden nodes in the RBF are as table 1.

Table 1. Procedure of selecting hidden nodes

- ① For each factor Z_h , finding its minimal and maximal values a_h and b_h .
- ② Setting the interference factor to M and choose the number of simulation runs N .
- ③ Computing the frequency $\omega_h = (N-1)/2M$ to be assigned to the factor Z_h .
- ④ Just considering the output of the hidden nodes in the RBF neural network, the factors are varied according to the curve definition; computing the total sensitivity index ST_h of the factor Z_h based on formula (12).
- ⑤ Computing the percentage contribution $ST_h / \sum_{i=1}^p ST_i$ of the output of each hidden node.
- ⑥ If the percentage contribution $ST_h / \sum_{i=1}^p ST_i$ larger than ε_1 , the node h is the active node and a new node should be inserted with relation to the active node. The initial parameters of the new node are given as:

Table 1. (continued)

- 1). $v_{new} = \frac{1}{2}(v_h + v_{nearest})$. v_{new} is the position of the center of the new node, v_h is the position of the center of the active node, $v_{nearest}$ is the position of the center of the node which is the nearest node of the active node.
- 2). $\delta_{new} = \min(\delta_i)$. δ_{new} is the width of the new node, δ_i are the width of the existing nodes.
- 3). $w_{new} = 0.5 \times w_{act}$. w_{new} is the weights of the new node, w_{act} is the weights of the active node.
- ⑦ Repeating ①-⑥ until all of the existing hidden nodes are considering.

3.2 Parameters Adjusting

After adding new nodes to the RBF neural network, the number of the centers is confirmed. Then the whole parameters of the RBF neural networks will be adjusted. In fact, these parameters adjusting relate to the final capabilities of the RBF neural networks directly.

Considering the training process of the RBF, researchers have put forward many methods to adjust the parameters of the RBF neural network. The parameter adjusting algorithm is based on the mean squared error (MSE) in this paper:

$$E(t) = \frac{1}{T} \sum_{t=1}^T (y_d(t) - y(t))^2 \quad (13)$$

where, T is the total number of the samples, $y_d(t)$ is the expected output of the t step, $y(t)$ is the practical output of the t step. The goal of this method is to reach $E(t) < \varepsilon$ by learning. ε is the expected stable error. The details of the adjusting process are:

1). The weights w ;

$$w_{ij}(t+1) = w_{ij}(t) - \eta_1 \frac{\partial E(t)}{\partial w_{ij}(t)} ; i = 1, 2, \dots, p, j = 1, 2, \dots, m \quad (14)$$

η_1 is a plus constant, and it is less than 1.

2). The width δ of the centers;

$$\delta_i(t+1) = \delta_i(t) - \eta_2 \frac{\partial E(t)}{\partial \delta_i(t)} ; i = 1, 2, \dots, N \quad (15)$$

η_2 is a plus constant, and it is less than 1.

3). The position v of the centers;

In section 3.2, the position v of the new inserting nodes has been discussed, and the position v of the other centers will be discussed here.

$$v_i(t+1) = \begin{cases} v_i(t) - \eta_3 (P(k) - v_i(t)) & \text{if } v_i \text{ is active} \\ v_i(t) - \eta_4 \frac{\partial E(t)}{\partial v_i(t)} & \text{others} \end{cases} ; i = 1, 2, \dots, p \quad (16)$$

η_3, η_4 are the plus constants, which are less than 1; p is the whole number of the hidden nodes after adding new nodes. This new algorithm will continue to circulate until reach the stable error.

3.3 Growing RBF Neural Network

Choose the parameters M , N , ε_1 , η_1 , η_2 , η_3, η_4 ; initialize the centers v , The width δ , The weights w and the number of the hidden nodes p . In each sampling period, the main steps of the growing RBF neural network algorithm are shown as table 2.

In this new growing RBF algorithm, two points need to be highlighted. First, usually, the former dynamic RBF neural network used the clustering methods or based on the information matrix, these methods required heavy computation. The SA method used in this paper is based on the Fourier translation, and then the percentage contribution of the hidden neurons is computed in a quantitative way. This growing does not have to judge for the structure every step, therefore gives a completely satisfactory method for growing the hidden nodes. Second, the common SA method is based on the quantitative and qualitative methods and between local and global techniques. The SA method used in this paper is global, and the percentage contribution of the hidden neurons is direct related to the RBF output.

According to the above analysis, this proposed GRBF neural network algorithm can obtain economical structure size; so that some computing time and memory space are saved.

Table 2. Procedure of GRBF neural network

<p>① Training a given RBF for some epochs.</p> <p>② Finding out the active nodes in the hidden layer and go to step③; if there is no active node, go to step④.</p> <p>③ Adding new nodes to the RBF neural network.</p> <p>④ Adjusting the parameters of the RBF neural network and updating the whole value of the parameters.</p> <p>⑤ Repeating the step②-④, Stopping computing until the GRBF catches the expected stable MSE.</p>
--

4 Simulations

To demonstrate the effectiveness of the proposed algorithm, the nonlinear function approximation sample is discussed in this paper. The results are compared with other algorithms such as SGP-RBF [12], and GAP-RBF [13]. Consider a common

non-linear function which was also used in [14] to demonstrate the effect of the algorithms:

$$y = 1.1 \times (1 - x_1 + 2x_2^2) \times e^{(-x_1^2 / 2)} \tag{17}$$

where there are 2 continuous attributes x_i ($i=1,2$). And the data set $\{x_i; y\}$ is generated by the equation (17), and x_i satisfies the uniform distribution $U [0, 10]$. For each trial, the size of training samples is 200, the size of testing samples is 200.

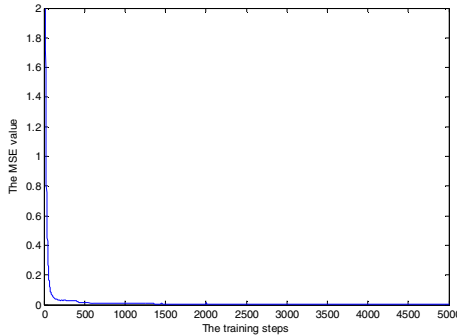


Fig. 1. The error results in the tracking process

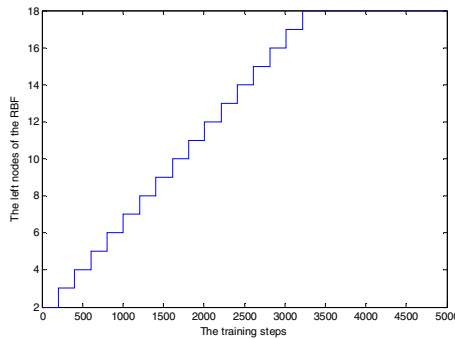


Fig. 2. The number of the nodes in the tracking process

The real output at time k is $y(k)$, the required value at time k is $y_d(k)$. The inputs of GRBF are given as: $P(k) = (x_1(k), x_2(k))$. The training MSE for tracking is 0.001, the initial radius of every hidden node is 0.1; the initial weight of every hidden node is randomly given in the interval $[0, 1]$. The initial value of M and N are $M = 4$ and $N = 5000$. There are two initial nodes in the hidden layer. Fig 1 shows 5000 steps of the error values in the training process, the error values show that when the new node is inserted to the hidden layer, the error values will shake. However, the error values can be convergence quick after adding new nodes; Fig 2 shows the dynamic number

of the nodes in the tracking process; Fig 3 shows the width δ of the centers after training; Fig 4 shows the RBF position of the centers ν after training. The final results of all the algorithms are shown in Table 3. The compared values are: training time, test error, left nodes after training.

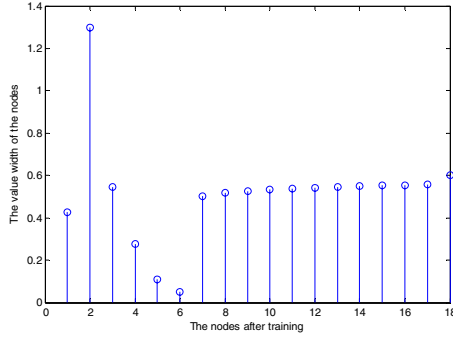


Fig. 3. The δ values of the left nodes after training

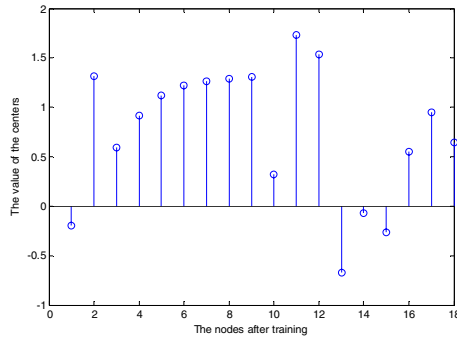


Fig. 4. The ν values of the left nodes after training

Table 3. The performance comparison of different algorithms

Algorithm	CPU Time(s)	Training Error	Testing Error	No. of Neurons
SGP-RBF	42.32	0.001	0.0086	19
GAP-RBF	26.86	0.001	0.0031	16
GRBF	22.67	0.001	0.0025	18

Based on table 3, the GRBF is faster and more accurate than SGP-RBF and GAP-RBF. The structure of this GRBF is simpler than the SGP-RBF; the memory space is fewer owing to the simple structure. The nodes in the GRBF are more than the GAP-RBF, but the algorithm is faster and more accurate than GAP-RBF. The results prove that this GRBF performs better than the former two algorithms.

5 Conclusion

This paper proposed a growing algorithm for the RBF structure design; which can obtain an economical RBF neural network and is insensitive to the initial conditions. The simulation results show that this proposed algorithm holds good performances for tracking the non-linear functions, identifying the non-linear dynamic systems and modelling the complexity systems. It can achieve better generalization abilities for the applications. Finally, this type of GRBF based approach has the potential to be used in estimating a range of variables that are typically troublesome to measure using hardware. This suggests that such new algorithm can be a relatively cost-effective approach for measuring and other useful applications.

References

1. Esposito, A., Marinaro, M., Oricchio, D., Scarpetta, S.: Approximation of continuous and discontinuous mappings by a growing neural RBF- based algorithm. *Neural Networks* 13(3), 651–665 (2000)
2. Wang, X.X., Chen, S., Brown, D.J.: An approach for constructing parsimonious generalized Gaussian kernel regression models. *Neurocomputing* 62, 441–457 (2004)
3. Orr, M.J.L.: Regularization on the selection of radial basis function centers. *Neural Computation* 7(3), 606–623 (1995)
4. Yingwei, L., Sundararajan, N., Saratchandran, P.: Performance evaluation of a sequential minimal radial basis function (RBF) neural network learning algorithm. *IEEE Trans. Neural Networks* 9(2), 308–318 (1998)
5. Salmerón, M., Ortega, J., Puntonet, C.G., Prieto, A.: Improved RAN sequential prediction using orthogonal techniques. *Neurocomputing* 41, 153–172 (2001)
6. Hao, P., Chiang, J.: Pruning and model-selecting algorithms in the RBF frameworks constructed by support vector learning. *International Journal of Neural Systems* 6(4), 283–293 (2006)
7. Saltelli, A., Ratto, M., Tarantola, S., Campolongo, F.: Sensitivity analysis practices: Strategies for model-based inference. *Reliability Engineering and System Safety* 91(10–11), 1109–1125 (2006)
8. Cariboni, J., Gatelli, D., Liska, R., Saltelli, A.: The role of sensitivity analysis in ecological modeling. *Ecological Modeling* 203(1–2), 167–182 (2007)
9. Saltelli, A., Chan, K.-S., Scott, E.M.: *Sensitivity Analysis*. Wiley, New York (2000)
10. Saltelli, A., Tarantola, S., Chan, K.-S.: A quantitative model independent method for global sensitivity analysis of model output. *Technometrics* 41(1), 39–56 (1999)
11. Lauret, P., Fock, E., Mara, T.A.: A Node Pruning Algorithm Based on a Fourier Amplitude Sensitivity Test Method. *IEEE Trans. Neural Networks* 17(2), 273–283 (2006)
12. Er, M.J., Wu, S.: A fast learning algorithm for parsimonious fuzzy neural systems. *Fuzzy Sets and Systems* 126(33), 337–351 (2002)
13. Hao, P., Chiang, J.: Pruning and model-selecting algorithms in the RBF frameworks constructed by support vector learning. *International Journal of Neural Systems* 6(4), 283–293 (2006)
14. Leng, G., Prasad, G., McGinnity, T.M.: An on-line algorithm for creating self-organizing fuzzy neural networks. *Neural Networks* 17(10), 1477–1493 (2004)

Fault Tolerance Based on Neural Networks for the Intelligent Distributed Framework*

Michael Angelo G. Salvo, Jaewan Lee, and Jung-sik Lee

School of Electronic and Information Engineering, Kunsan National University 69121 68
Miryong-dong, Kunsan, Chonbuk 573-701, South Korea
{masalvo, jwlee, jslee}@kunsan.ac.kr

Abstract. Current studies on the intelligent distributed framework in distributed systems use multi-agents which include replication agents, grouping agents, locator agents, and load balancing agents among others which work systematically to provide quality of service. This research aims to improve that quality of service by implementing a neural network in the fault tolerant scheme. In case of an object failure or disconnection, the properties of that object will be used as input data in the multilayer perceptron (MLP) to select an alternate object. The fault tolerant scheme then chooses a new object by training these properties using the backpropagation algorithm. Results show that the proposed algorithm recorded the highest accuracy rate compared to the ZeroR, Simple Logistic, and J48 algorithms.

Keywords: Fault Tolerance, Neural Networks, Intelligent Distributed Framework.

1 Introduction

Current researches for the intelligent distributed framework use mobile agents which addresses the use of intelligence in the distributed environment. Agents are provided with intelligence based on acquired information. The newly discovered rules from the environment are used by the agent to change or evolve its task to solve problems. Multi-agent and mobile agent technologies provide developers a new paradigm of designing and implementing software applications.

In the distributed environment, object group models are designed to manage the system by grouping appropriate objects. The object groups perform cooperation for efficient service. Communications of object groups reflect the inter-dependence and take place from one group to another. An object group is a set of objects related logically. A group acts as a logical addressable entity where an entity that requests a service from the group is a client of the group [1].

For clients, quality of service (QoS) is important where object replication is implemented to handle the large number of client requests. In contrast of QoS, the management of the replicated objects is necessary and each load from servers must be

* This paper is supported by the Industry and University Research Consortium of Small and Medium Business Administration (SMBA).

balanced in order for the system to provide reliable services and have an optimal performance. As discussed in [2], any applications require the middleware on which they are built be predictable and reliable.

Quality of service includes reliability and performance which this paper aims to improve on. In this paper, objects in the distributed environment are grouped based on their similarity in terms of each object's properties. But replication schemes do not guarantee that the primary replica would be readily available. Therefore, in the event of object disconnection, another object with similar or nearest properties to the one which was disconnected shall process the client's request. For this, we use the multilayer perceptron (MLP) architecture to train the disconnected objects properties as input data using the backpropagation algorithm. The MLP is compared to the ZeroR classifier, Simple Logistic, and J48 tree. The result was favorable for the MLP architecture.

2 Related Works

2.1 Fault Tolerance in Distributed Objects

In [3], a fault-tolerant object using replication of objects in network-wide distributed object-oriented communications systems, and a mechanism for managing multiple objects that execute the target functions in the system were discussed. The replication management mechanism combines fault detection using the local timer of each node and a mechanism for maintaining internal state consistency. This mechanism reduces the overhead of replication and at the same time distributes the load. In [4], a new algorithm is presented to support fault tolerant objects in distributed object oriented systems. The algorithm uses checkpointing and message logging scheme. However the novelty of this scheme is in identifying the checkpointing instances such that the checkpointing time will not affect the regular response time for the object requests.

As discussed in [2], FT-CORBA specification defines the architecture, a set of services, and associated fault tolerance mechanisms that constitute a framework for resilient, highly available, distributed software systems. Through application of this framework, applications can attain a degree of reliability beyond what can be achieved through traditional server redundancy.

Recently, the distributed real-time services are developing in distributed object computing environments in a way that can support a new programming paradigm of the distributed platform that requires interoperability among heterogeneous systems. These services are based on distributed middleware and object-oriented technologies. But there have been difficulties in the managing of distributed objects and providing real-time objects with the timing constraints. In [5], discusses a real-time object group (RTOG) platform that can manage and group the distributed objects for reducing their own complicated managements and interfaces, and add distributed real-time requirements to real-time objects without modifying the ORB itself. While in [6] tackles the Object Group Service (OGS) programming model of CORBA and discussed various levels of group transparency.

A grouping service which implements classification method for efficient search of objects was also discussed in [1]. The intelligent model uses extracted rules for

dynamic replication scheme of objects. The replication agent (RA) creates and manages the replica objects in the server. After initializing a new object replica it will send the object information to the grouping agent. The grouping agent is the one who manages the object groups. It classifies the object according to its properties and then sends the new structure to the locator agent (LA). The LA in turn adjusts its structure for efficient search of objects. To ensure that the reproduction of your illustrations is of a reasonable quality, we advise against the use of shading. The contrast should be as pronounced as possible.

2.2 Implementing Failure Detectors

A failure ‘detector’ is not necessarily accurate. Most fall into the category of *unreliable failure detectors* according to [7]. An unreliable failure detector may produce one of two values when given the identity of a process: *Unsuspected* or *Suspected*. Both of these results are hints, which may or may not accurately reflect whether the process has actually failed. A result of *Unsuspected* signifies that the detector has recently received evidence suggesting that the process has not failed; for example, a message was recently received from it. But of course the process can have failed since then. A result of *Suspected* signifies that the failure detector has some indication that the process may have failed.

A *reliable failure detector* is one that is always accurate in detecting a process’s failure. It answers processes queries with either a response of *Unsuspected* – which, as before, can only be a hint – or *Failed*. A result of *Failed* means that the detector has determined that the process has crashed. [8, 9] discussed the implementation of failure detectors in Distributed Systems, while [10] tested the quality of failure detectors. In this study, we shall use the most common of failure detection methods, the sending of *is_alive* or heartbeat messages as discussed in [2].

3 Structure of the Intelligent Distributed Framework

The ubiquitous layer consists of software and hardware components utilizing the resources on the ubiquitous environment. In this layer, hardware components are mobile devices, wireless sensors and embedded devices while software components are agents and primitive commands to communicate with the services. The mobile and wireless devices are classified into two; 1) object-oriented and 2) proxy controlled devices. The object oriented devices have enough processing power to host objects in able to interact with the distributed object system seamlessly while the proxy controlled has no capability to host an object but only provide a proxy within the system to access its primitive controls. An adaptive ubiquitous middleware is presented which is another sub layer between the intelligent middleware layer and ubiquitous layer that provide services to support mobility. The intelligent middleware layer acts as the primary middleware where services are transparently operating on serving the clients. Interaction of clients from ubiquitous layer and services are handled by this layer. Users and administrators do not need to know the configuration on how to find, where to find and how to manage the resources but transparently executes services. The service functions are influenced by the adaptive module in able

to interact correctly with services in goal of optimizing the performance of tasks. The infrastructure layer is a network of different computers that are powerful of storing data and processing transactions within the system, like PCs and servers. These computers are communicating in wired network system to provide massive information exchange between the systems. The intelligent middleware layer presents an overlay communication of software components that are hosted by hardware devices in the ubiquitous and infrastructure layers.

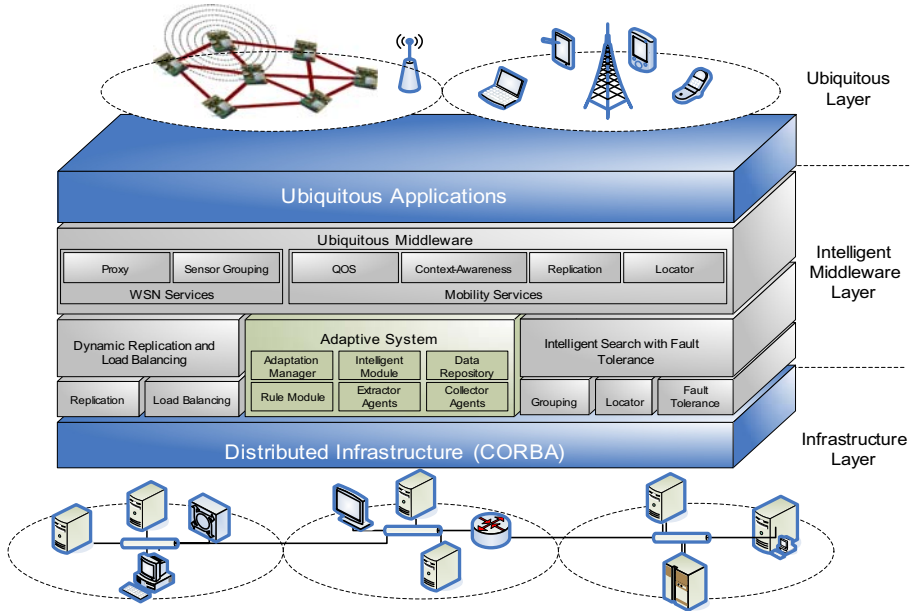


Fig. 1. Intelligent Distributed Framework consists of three (3) layers supporting ubiquitous technology and distributed object system

3.1 Components of the Fault Tolerant Scheme

Replication Agent. The replication agent (RA) creates and manages object replicas in the server, as discussed in [11]. RAs within the server are coordinating to other RA managing the replicated objects. If an object has changed its values then RA of that object communicated through other RA to inform the changes. Monitoring and analyzing the access of the clients to the objects are necessary in adapting the system requirements where the result of the analysis is the creation or termination of an object replica.

Load Balancing Agent. The load balancing agent is responsible for load distribution. A load is defined as a single access of a client to an object. In accessing objects, the loads are distributed to the object replicas and follow the threshold values from [12]. An adaptive scheme from [1] is used to distribute loads in order to overcome the problem from two common algorithms of load balancing. The load balancing agent coordinates with the sub-components of the group which determines the object replicas in distributing the loads.

Grouping Agent. The grouping agent manages the object groups. A dynamic classification is done to determining the membership of the object based on its properties. The properties are configured by the application users. Object groups are bound to other object groups to make the service more efficient on searching the objects. Dynamic group binding is discussed in [1].

Locator Agent. Client requests are queried through the object groups of the system. However, the bound object groups contain large number of object information where it is difficult to find the most appropriate object for the request. This is discussed in [13]. Locator agent classifies the request content to choose the appropriate object. The locator agent processes the client's request by sending the processed input to mobile agents and calculates the membership function. Each mobile agent which is initialized by the grouping agent represents object groups assigned by the group service. A mobile agent uses its fuzzy rule and processes the classification.

4 Selection of Alternative Object Based on Neural Networks

In many applications, a "correct" answer if delivered too late is considered a failure. Therefore, such systems must continue to function even in the presence of faults. Though having replication schemes provides quality of service, it does not guarantee that the replicated object will continue to function. This study uses the multilayer perceptron to improve the quality of service offered by the intelligent distributed framework so that incase of object disconnection, an alternate object will be chosen to process the client's request. Consider objects from different servers. These are objects with varying property values. But some objects are similar to objects in other servers by having similar properties. These objects are grouped virtually based on the k -means algorithm.

$$J = \sum_{n=1}^N \sum_{m=1}^M \|x_{nm} - c_m\| \quad (1)$$

Based on the number of k clusters specified by an administrator, the objects are processed to the unsupervised algorithm of k -means shown in Equation 1 where the vector values of each objects calculates the distance from group center value. The process continues to adjust the groupings until a certain threshold is achieved. We get the minimum value from the objective function to stop the process. After the process in Equation 1, the system produces a virtual group shown in Figure 2 where each object on a server joins the virtual group.

In this study we use that advantage to select an alternate object to process a client request in the event of object disconnection. This paper's main contribution and focus is on this distinct event. In order to detect failures we assume the simplest form of fault detection which is the sending of heartbeat messages. An accessed object o sends an '*o is here*' to the fault detector and does this every T seconds. The fault detector estimates a maximum transmission time of D seconds. If the fault detector does not receive the message then it will be assumed that object o has failed. The property values of the disconnected object will then be taken as input data in training the MLP using the backpropagation algorithm. Once a new object is found, that object will resume operation of the client's request.

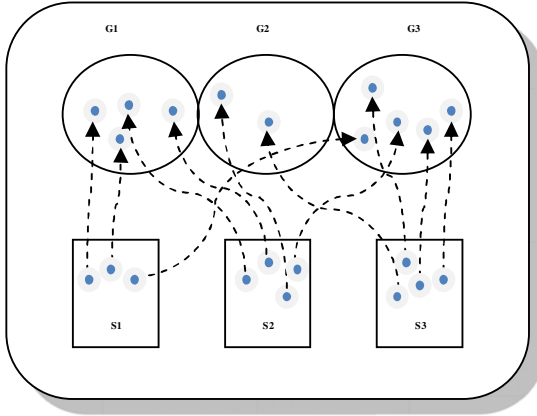


Fig. 2. The grouping based on k -means clustering algorithm. This is used to provide compactness in each cluster and group objects with similar properties together.

Backpropagation is a supervised learning algorithm where the desired output is known. Input patterns are trained and the error is propagated backwards from the output to the input layer.

As an example, assume a job promotion in a company where a managerial position is vacant and eligible employees will be evaluated based on performance, resourcefulness, and skills. By definition, a candidate must have high marks on each of the criteria. But under some circumstance, the top candidate was unable to be considered for promotion. Thus, the candidate next in line must be the one to take his place and receive the promotion.

The marks received by each employee according to the criteria are comparable to the property values of each object, be it according to services, etc. Each property value acts as a neuron. The action of a potential neuron is determined by the weight associated with the neuron's inputs in Equation 2. A threshold modulates the response of a single neuron to a particular stimulus confining such response to a predetermined range of values.

$$z = \sum_{i=1}^n x_i w_i \quad (2)$$

Equation 3 defines the output y of a neuron as an activation function f of the weighted sum of $n+1$ inputs.

$$y = f\left(\sum_{i=0}^n x_i w_i\right) \quad (3)$$

The threshold is incorporated into the equation as the extra input in Equation 4.

$$f(x) = \begin{cases} 1 & \text{if } \sum_{i=0}^n x_i w_i > 0 \\ 0 & \text{if } \sum_{i=0}^n x_i w_i \leq 0 \end{cases} \quad (4)$$

The output produced by a neuron is determined by the activation function. This function should ideally be continuous, monotonic, and differentiable. With these features in mind, the most commonly chosen function is the sigmoid which is shown in Equation 5.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (5)$$

The accuracy of the response is measured in terms of an error E defined as the difference between the current and desired output in Equation 6.

$$E = \frac{1}{2} \sum_j (t_{pj} - o_{pj})^2 \quad (6)$$

The error E is propagated backwards from the output to the input layer. Appropriate adjustments are made by slightly changing the weights in the network by a proportion of the overall error E .

After weights are adjusted, the examples are presented again. The error is again calculated and weights are adjusted and the process is repeated until the current output is satisfactory or the network cannot improve further.

We present the input-output pair p and produce the current output o_p . We then calculate the output of the network and calculate the error for each output unit for a particular pair using Equation 7.

$$\delta_{pj} = (t_{pj} - o_{pj}) f'(net_{pj}) \quad (7)$$

In Equation 8, we calculate the error by the recursive computation of δ for each of the hidden units j in the current layer. Where w_{kj} are the weights in the k output connections of the hidden unit j , δ_{pk} are the error signals from the k units in the next layer and $f'(net_{pj})$ is the derivative of the activation function. Then we propagate backwards the error signal through all the hidden layers until the input layer is reached.

$$\delta_{pj} = \sum_k \delta_{pk} w_{kj} f'(net_{pj}) \quad (8)$$

Finally, we repeat the steps for Equations 7 and 8 until the error is acceptably low. After the structure of the neural network is trained, the structure is used to compare the property of the disconnected object to other objects in the group.

Table 1. An example of collected properties from objects which will be used as input patterns to select a new object in case of disconnection

	$P(1)$	$P(2)$	$P(3)$	$P(4)$	$P(5)$	$P(6)$	$P(7)$	$P(8)$	$P(9)$	$P(10)$
$O(1)$	2	5	1	0	4	4	6	5	7	3
$O(2)$	6	1	9	5	5	5	3	4	3	2
$O(3)$	3	2	8	3	1	1	4	2	0	4
$O(4)$	0	9	5	1	2	3	0	3	7	5
$O(5)$	4	4	8	2	1	5	9	5	0	0
$O(6)$	2	1	0	5	3	4	5	6	9	8

Table 1 shows an example of objects O and their property values P . Each value is taken from a certain trait of an object. As another example, in the healthcare environment, the objects (that offer their services to clients) are physicians and the properties are the numerical representations of the general knowledge of the physician in each specialization; $P(1)$ = Cardiology, $P(2)$ = Endocrinology, $P(3)$ = Pediatrics, $P(4)$ = Radiology, etc. In the event of an object disconnection, e.g. the physician is not available due to some reasons, these properties are taken as input patterns to train the neural network so system can select an alternate object to process the client's request.

5 Simulation Results

The proposed intelligent distributed framework used Visibroker 7.0 to implement the CORBA-based object grouping. We programmed the ANN in Java and put the functionality in a JADE agent. The operating system used for the experiment was Windows XP. Our simulation focused on testing the neural network's ability to select an object which is the most similar to the disconnected object using one, two, and three hidden layers in the MLP structure. We then train these input patterns using the back propagation algorithm. The proposed method was evaluated by determining the generalization accuracy of the multilayer perceptron and was later compared with the accuracy of other algorithms.

We used 25000 epochs for our simulation. First, we tested the algorithm using 5 neurons in the input pattern. The number of neurons represent the number of properties taken into consideration in selecting a new object. After recording the general accuracy, we increased the hidden patterns to two, and finally to three. Table 2 shows the performance of the network.

Table 2. Generalization accuracy of the algorithm trained with the respective number of neurons and hidden layers. It can be inferred that there is no significant increase in generalization accuracy when increasing the hidden layers to three from two layers.

Number of Neurons	One Hidden Layer	Two Hidden Layers	Three Hidden Layers
5	89.8%	92.1%	92.3%
10	91.1%	92.5%	92.2%
15	90.4%	92.7%	93.1%
20	90.6%	92.3%	92.1%

After determining that 2 hidden layers provide higher generalization accuracy, and that three hidden layers only increases the computational time, the MLP with two hidden layers was used to compare to other algorithms. We used a data from the UCI data repository, a popular repository of different data for data mining experiments. Figure 3 shows the accuracy of each algorithm in classifying 303 patterns using 10 folds of cross-validation. The MLP provided the highest accuracy rate with 84.49% followed closely by the Simple Logistic with 83.19%. J48 gave 77.56% while ZeroR was only 54.46% accurate.

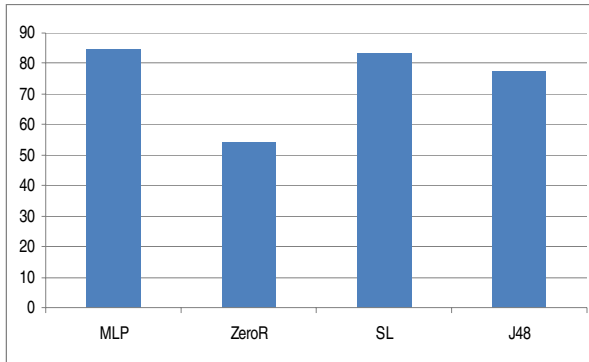


Fig. 3. Bar graphs showing the accuracy of each algorithm. The MLP provided the highest accuracy rate with 84.49% followed closely by the Simple Logistic with 83.19%. J48 gave 77.56% while ZeroR was only 54.46% accurate.

6 Conclusion

This paper presented a fault tolerant scheme for the intelligent distributed framework by using an artificial neural network (ANN). Properties of a disconnected object were used as input patterns in training the multilayer perceptron (MLP), which is an ANN. The properties were trained using the back propagation algorithm to select a new object to process the client's request. Simulation results show that two hidden layers provide higher generalization accuracy than only one hidden layer. Furthermore, the MLP yielded the highest accuracy rate compared to ZeroR, Simple Logistic, and J48.

References

1. Mateo, R., Lee, J.: Intelligent Distributed Platform using Mobile Agent based on Dynamic Group Binding. *Journal of Korean Society for Internet Information* 8(3), 131–143 (2007)
2. Martin, R., Totten, S.: Introduction to Fault Tolerant CORBA, CORBA News Brief (2003), <http://www.ociweb.com>
3. Takemoto, M.: Fault-tolerant Object on Network-wide Distributed Object-oriented Systems for Future Telecommunications Applications. In: *Pacific Rim International Symposium on Fault-Tolerant Systems*, pp. 139–146 (1997)

4. Beedubail, G., Karmarkar, A., Gurijala, A., Marti, W., Pooch, U.: An Algorithm for Supporting Fault Tolerant Objects in Distributed Object Oriented Operating Systems. In: Proceedings of the 4th International Workshop on Object Orientation in Operating Systems, pp. 142–148 (1995)
5. Joo, S., Oh, S., Shin, C., Hwang, J.: CORBA Based Real-Time Object-Group Platform in Distributed Computing Environment. In: Sloot, P.M.A., Abramson, D., Bogdanov, A.V., Gorbachev, Y.E., Dongarra, J., Zomaya, A.Y. (eds.) ICCS 2003. LNCS, vol. 2659, p. 703. Springer, Heidelberg (2003)
6. Felber, P., Guerraoui, R.: Programming with Object Groups in CORBA. *IEEE Concurrency* 8(1), 48–58 (2000)
7. Coulouris, G., Dollimore, J., Kindberg, T.: *Distributed Systems Concepts and Design*, 4th edn. Pearson/Addison Wesley (2005)
8. Chandra, T., Toueg, S.: Unreliable Failure Detectors for Reliable Distributed Systems. *Journal of the ACM* 43(2), 225–267 (1996)
9. Hayashibara, N., Cherif, A., Katayama, T.: Failure Detectors for Large-Scale Distributed Systems. In: 21st IEEE Symposium on Reliable Distributed Systems, p. 404 (2002)
10. Xiong, N., Yang, Y., Chen, J., He, Y.: On the Quality of Service of Failure Detectors Based on Control Theory. In: Proceedings of the 20th International Conference on Advanced Information Networking and Applications, pp. 75–80 (2006)
11. Mateo, R., Lee, M., Lee, J.: Load balancing using dynamic replication scheme for the distributed object group. In: Szczuka, M.S., Howard, D., Ślęzak, D., Kim, H.-k., Kim, T.-h., Ko, I.-s., Lee, G., Sloot, P.M.A. (eds.) ICHIT 2006. LNCS, vol. 4413, pp. 458–468. Springer, Heidelberg (2007)
12. Mateo, R., Yoon, I., Lee, J.: Cooperation Model for Object Group Using Load Balancing. *International Journal of Computer Science and Network Security* 6(12), 138–147 (2006)
13. Mateo, R., Lee, J.: Intelligent Search Mechanism based on Neuro-fuzzy System for the Distributed Object Groups. *International Journal of Security and Its Applications* 2(1), 19–28 (2008)
14. Chen, C., Tsai, K.: The Server Reassignment Problem for Load Balancing in Structured P2P Systems. *IEEE Transactions on Parallel and Distributed Systems Peer-to-peer Computing* 19(2), 234–246 (2008)

Learning RNN-Based Gene Regulatory Networks for Robot Control

Wei-Po Lee and Tsung-Hsien Yang

Department of Information Management,
National Sun Yat-sen University
Kaohsiung, Taiwan
wplee@mail.nsysu.edu.tw

Abstract. With the unique characteristic of orchestrating gene expression level in cellular metabolism during the development of living organisms, gene regulatory networks can be modeled as reliable and robust control mechanisms for robots. In this work we devise a recurrent neural network-based GRN model to control robots. To simulate the regulatory effects and make our model inferable from time-series data, we develop an enhanced learning algorithm, coupled with some heuristic techniques of data processing for performance improvement. We also establish a method of programming by demonstration to collect behavior sequence data of the robot as the expression profiles, and then employ our framework to infer controllers automatically. To verify the proposed approach, experiments have been conducted and the results show that our regulatory model can be inferred for robot control successfully.

Keywords: recurrent neural network, gene regulation, time series data, bio-inspired robot control, learning by demonstration.

1 Introduction

Gene regulatory networks (GRNs) are complex biological systems. They dynamically orchestrate the level of expression for each gene in the genome by controlling whether and how the gene will be transcribed into RNA [1]. In a GRN, the network structure is an abstraction of the chemical dynamics of a system, and the network nodes are genes that can be regarded as functions obtained by combining basic functions upon the inputs. These functions have been interpreted as performing a kind of information processing within the cell, which determines cellular behaviors. Such systems often include dynamic and interlock feedback loops for further regulation of network architecture and outputs. With these unique characteristics, GRNs can be modeled as reliable and robust control mechanisms for robots.

The first important step in applying GRNs to controlling robots is to develop a framework for GRN modeling. In the work of GRN modeling, many regulation models have been proposed [2][3]; they can range from very abstract models (involving Boolean values only) to very concrete ones (including fully biochemical interactions with stochastic kinetics), depending on the biological levels to be studied. To

construct a network from experimental data, an automated procedure (i.e., reverse engineering) is advocated: the GRN model is firstly determined, and then different computational methods are developed for the chosen model to reconstruct networks from the time-series data [2][3].

As can be observed from the literature, works in modeling GRNs shared similar ideas in principle. However, depending on the research motivations behind such works, different researchers have explored the same topic from different points of view; thus the implementation details of individual work are also different. Works with abstract models involve less biological detail and display only qualitative dynamic behavior; they can implement large size networks. On the other hand, works with concrete models describe network dynamics in detail and is closer to biological reality; they can only implement small size networks. The effect of model selection is especially apparent in the application of robot control. For example, in the work by Quick et al. [4] and Stewart et al. [5], the authors have emphasized on describing the operational details of their artificial genes, enzymes, and proteins to show how their GRN-based control systems are close to the biological process, rather than how their models can be used to control robot practically. In addition, the work by Bentley has described a GRN model involving interactions of genes and proteins, and implemented a simulated evolution mechanism to derive simple controllers [6]. However, its main focus is on the developmental process of individuals during the evolution, but not the control task. Different from the above works, we will take the principles of gene regulation to develop a biologically plausible model, investigate whether the presented model is inferable from data, and how to exploit the GRN modeling approach to construct robot controllers in particular.

As recurrent neural networks (RNNs) consider the feedback loops and take internal states into account, they are able to show the system dynamics of the network over time. With these characteristics, this kind of network that offers an ideal model to act as GRNs, is especially suitable for the control systems. Therefore, in this work we develop a RNN-based regulatory model for robot control. To simulate the regulatory effects and make our model inferable from time-series expression data, we also implement an enhanced RNN learning algorithm, coupled with some heuristic techniques of data processing to improve the learning performance. After establishing a framework for GRN modeling, we develop a method of programming by demonstration to collect behavior sequence data of a robot as the time series profiles, and then employ our framework to infer behavior controllers automatically. To verify the presented approach, two series of experiments have been conducted to demonstrate how it operates and how it can be used to construct controllers for a mobile robot and a walking robot successfully.

2 Modeling GRNs for Robot Control

The most important issues in using GRNs for robot control are the development of GRN model and the computational method for constructing the model from available time series data. To address the relevant problems, this section describes how we take biological, computational, and engineering points of view to develop an inferable control model and how to use this model to learn robot controllers.

2.1 RNN-Based Regulatory Model

There are several recurrent neural network architectures, ranging from restricted classes of feedback to full interconnection between nodes. Vohradsky and his colleagues have proposed the use of fully recurrent neural network architecture in studying regulatory genetic networks such as those involved in the transcriptional and translational control of gene expression [7]. In this work, we also take a similar architecture to model GRN, but unlike their work that mainly simulates regulatory effects, our goal is to infer regulatory networks from measured expression data.

In a fully recurrent net, each node has a link to any node of the net, including itself. Using such a model to represent a GRN is based on the assumption that the regulatory effect on the expression of a particular gene can be expressed as a neural network in which each node represents a particular gene and the wiring between the nodes define regulatory interactions. Similarly, in a GRN, the level of expression of genes at time t can be measured from a gene node, and the output of a node at time $t+\Delta t$ can be derived from the expression levels and connection weights of all genes connected to the given gene at the time t . That is, the regulatory effect to a certain gene can be regarded as a weighted sum of all other genes that regulate this gene. Then the regulatory effect is transformed by a sigmoidal transfer function into a value between 0 and 1 for normalization.

The same set of the above transformation rules is applied to the system output in a cyclic fashion until the input does not change any further. As in [7], here we use the basic ingredient to increase the power of empirical correlations in signaling constitutive regulatory circuits. It is to generate a network with nodes and edges corresponding to the level of gene expression measured in microarray experiments, and to derive correlation coefficients between genes. To calculate the expression rate of a gene, the following transformation rules are used:

$$\frac{dy_i}{dt} = k_{1,i}G_i - k_{2,i}y_i$$

$$G_i = \{1 + e^{-(\sum_j w_{i,j}y_j + b_i)}\}^{-1}$$

where y_i is the actual concentration of the i -th gene product; $k_{1,i}$ and $k_{2,i}$ are the accumulation and degradation rate constants of gene product, respectively; G_i is the regulatory effect on each gene that is defined by a set of weights (i.e., $w_{i,j}$) estimating the regulatory influence of gene j on gene i , and an external input b_i representing the reaction delay parameter.

When the above GRN model is used to control a robot, each gene node now corresponds to an actuator of the robot in principle. Two extra nodes are added to serve as inter-genes and their roles are not specified in advance. The redundancy makes the controllers easier to be inferred from data. Fig. 1 illustrates the architecture of our GRN controller. In this architecture, the sensor information received from the environment is continuously sent to all nodes of the fully interconnected network, and the outputs of the actuator nodes (i.e., a_i) are interpreted as motor commands to control the robot. For control tasks in which the sensor information is not required, for example the locomotion task, the perception part in the figure is simply disabled.

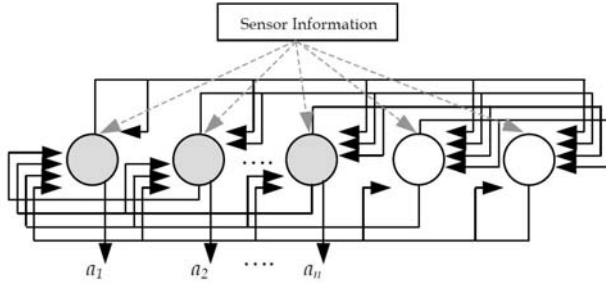


Fig. 1. The GRN control architecture

2.2 Learning Algorithm for Constructing GRN Controllers

After the network model for robot control is decided, the next phase is to find settings of the thresholds and time constants for each neuron, and the weights of the connections between the neurons so that the network can produce the most approximate system behavior. By introducing a scoring function for network performance evaluation, the above task can be regarded as a parameter estimation problem with the goal of maximizing the network performance (or minimizing an equivalent error measure). To achieve this goal, here we use the backpropagation through time (BPTT, [8]) learning algorithm to update the relevant parameters of recurrent networks in discrete-time steps.

Instead of mapping a static input to a static output as in a feedforward network, BPTT maps a series of inputs to a series of outputs. The central idea is “unfolding” the discrete-time recurrent neural network (DTRNN) into a multilayer feedforward neural network when a sequence is processed. Once a DTRNN has been transformed into an equivalent feedforward network, the resulting feedforward network can then be trained using the standard backpropagation algorithm.

The goal of BPTT is to compute the gradient over the trajectory and update network weights accordingly. As mentioned above, the gradient decomposes over time. It can be obtained by calculating the instantaneous gradients and accumulating the effect over time. In BPTT, weights can only be updated after a complete forward step during which the activation is sent through the network and each processing element stores its activation locally for the entire length of the trajectory. That is, as in the traditional backpropagation algorithm, the weights are updated in the backward step according to the following rules [8]:

$$\Delta w_{i,j} = -\eta \frac{\partial E(t_0, t_1)}{\partial w_{i,j}} = \eta \sum_{\tau=t_0}^{t_1} \delta_j(\tau) x_i(\tau-1)$$

$$\delta_j(\tau) = \begin{cases} f'(net_j(\tau)) e_j(\tau) & \text{if } \hat{\delta} = t_1 \\ f'(net_j(\tau)) \left[e_j(\tau) + \sum_{\ell \in U} \delta_\ell(\tau+1) w_{i,\ell} \right] & \text{if } t_0 \leq \hat{\delta} < t_1 \end{cases}$$

where δ_j is the derivative of the error surface, η is the learning rate, e_j means the corresponding error, and U is the set of all output units.

In the above learning procedure, learning rate is an important parameter. Yet, it is difficult to choose an appropriate value to achieve an efficient training because the cost surface for multi-layer networks can be complicated and what works in one location of the cost surface may not work well in another location. Delta-bar-delta is a heuristic algorithm for modifying the learning rate in the training procedure [9]. It is inspired by the observation that the error surface may have a different gradient along each weight direction so that each weight should have its own learning rate. In our modeling work, to save the effort in choosing appropriate learning rate, this algorithm is implemented for automatic parameter adjustment.

2.3 Robot Programming by Demonstration

After establishing a framework for GRN modeling, we then use it to construct behavior controllers for robots. In this work, we develop a method of programming-by-demonstration (sometimes called imitation) to collect data of behavior sequence as gene expression profiles, and then employ the GRN modeling framework to infer behavior controllers automatically.

Imitation has been studied in ethology and experimental psychology for many years. It is a powerful mechanism in social animals for learning and delivering new knowledge [10]. In fact, some experiments have discovered that there exists a mechanism in the primate brain to drive the imitation behavior. For example, macaque inferior premotor cortex has been identified as being involved in the movements of reaching and grasping, and the mirror neurons are active when a monkey performs certain kinds of actions, such as grasping, manipulating, and placing [10].

The imitation mechanism has two parts: one is an active process that is for acquiring new behaviors; the other is a passive process for imitating known behaviors. For a robot, the former is to try to employ a certain learning strategy to produce the behavior that is currently shown but not known previously. And the latter is to recognize what kind of behavior a demonstrator is performing and to retrieve the same kind of behavior that has been previously developed and recorded in the memory. As the passive process can be achieved by a straightforward way (i.e., building a mapping table to link the extracted behavior trajectory to the most similar one recorded previously), in this work we concentrate on the active process (i.e., learning new behaviors). For active imitation, we take an engineering point of view and consider imitation as a vehicle for learning new behaviors. It can be considered as a method of programming by demonstration [11][12]. Fig. 2 illustrates the procedure. In this method, the robot is firstly shown how to perform the desired behavior: it is driven manually to achieve the target task. In this stage, the robot can be regarded as a teacher showing the correct behavior. During the period of human-driven demonstration, at each time step the relevant information received from the robot's sensors and actuators are recorded to form a behavior data set for later training. In other words, it is to derive the time-series expression profiles of sensors and actuators from the qualitative behavior demonstrated by the robot.

After the behavior data is obtained, in the second stage the robot plays the role of a learner that is trained to achieve the target task. As described in the above sections, a RNN-based GRN model is adopted as the behavior controller here for the learner, and the corresponding learning algorithm is used to train the controller. To cope with

different environment situations, the robot is operated to achieve the target behavior a few times so that a reliable and robust controller can be obtained. All expression data from different behavior trials are collected and arranged in a single training set. By minimizing the accumulated action error over the entire training data set, the robot can improve its behavior and finally achieve the task. If the robot cannot produce a similar behavior as in the demonstration, the user can modify the training set by driving the robot to repeat the sub-behaviors that it failed to achieve, and then adding the newly obtained patterns to the data set to start the re-learning procedure.

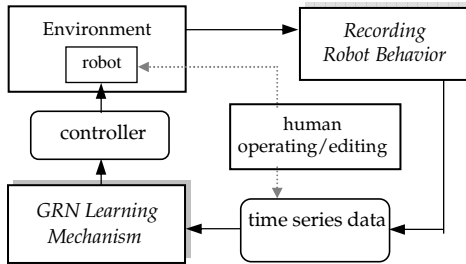


Fig. 2. Robot programming by demonstration

3 Experiments and Results

Following the proposed GRN model and the corresponding learning procedure, we conduct two series of experiments for evaluation. The first series is to examine whether the proposed model can be inferred from a given set of time series data. In this phase, data sets are obtained from GRN simulation software. The second series of experiments is to investigate whether the proposed approach can infer GRNs for robot control. The data sets in the series of experiments are behavior sequences of the robots and they are collected from the demonstration procedure.

3.1 Modeling GRNs

In the experiments of GRN modeling, we firstly used a GRN simulation software Genexp ([7]) to produce expression data. Different gene networks were defined in which the accumulation and degradation rate constants of gene product were chosen from preliminary test. Because all sets of experiments have similar results, only one set of the experiments is reported as a representative and explanatory example here.

In this example, a four genes network was defined and the simulation was run for 30 time steps for data collection. Then the proposed approach was employed to infer the above network. The result is shown in Fig. 3. It compares the behaviors of each pair of genes in the original and reconstructed networks, in which the x-axis represents time step and y-axis, the concentrations of different gene components. As can be observed, the behaviors of the two systems are nearly identical and the accumulated error for the five nodes is very small. It shows that the network can be reconstructed from the expression data by the learning mechanism presented.

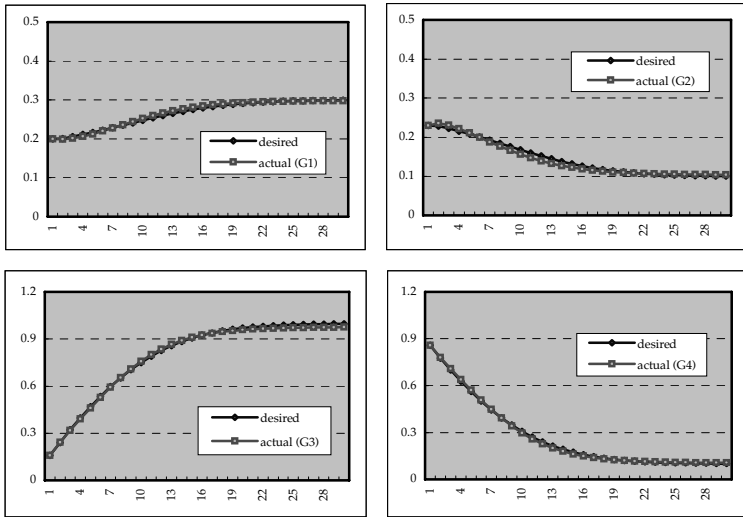


Fig. 3. Behaviors of the target (desired) and inferred (actual) systems

3.2 Learning GRNs for Robot Control

In the second series of experiments, we examined whether the GRN controllers can also be inferred by the learning mechanism from the time series data collected in the procedure of human-driven demonstration. Experiments have been conducted for two kinds of robots: one mobile robot and one walking robot. To save evaluation time in the learning procedure, the experiments were performed in simulation. In fact, many studies have shown that the controllers developed in a realistic simulator can be transferred successfully to a real robot [13]. In addition, to obtain robust controllers, random noise has been injected to the sensor/actuator dataset as in the experiments of GRN modeling described in the above section.

The first experiment is to develop a controller for the robot to achieve a box-pushing task. The popular mobile robot *Khepera* (described in [13]) has been used in this experiment. As described in section 2.3, in the experiment the robot was driven manually to perform the target task, and the relevant information of sensors and actuators was recorded as time series data to derive the controller. In our experiments, a fine time-slice technique was used in simulation and each time step lasted for 100 ms. Once the data set of behavior sequence has been collected, it was used as the expression profiles to infer the controller. Here a four nodes GRN controller was defined, including two nodes serving as the left and right wheel motors and the other two nodes as inter-genes. In the learning procedure, at each time step the perception information from infrared sensors was sent to each node and the controller was executed once. The outputs obtained from the specific nodes of the controller were then recorded and compared to the desired values. By minimizing the accumulated motor

error the target controller can be inferred successfully. Fig. 4 shows the typical box-pushing behavior generated by the inferred controller.

The above learning procedure is an efficient and convenient method to obtain behavior controllers without explicit programming. But it should be noted that to ensure the success of this approach the problem of class imbalance has to be overcome [14][15]. The problem means that in a classification task, when the numbers of examples within each class are quite different, the classification performance of the standard classifier may be damaged. This is a crucial problem to be encountered in the machine learning community since the data collected is often not distributed normally in the real world applications. In the robot learning application, to achieve a goal behavior means to accomplish a sequence of some sub-goal behaviors. Corresponding to a classification task, here the pair of input and output of the controller at a certain time step represents a data point, and each sub-task can be regarded as a specific class that considers the similar data points. As the robot may spend different numbers of time steps to deal with different sub-tasks, the training data set is thus imbalanced. When the robot cannot learn the target behavior properly, it will be more efficient to iteratively demonstrate and modify the parts the robot wrongly behaved to collect data to further balance the training set.

Different from the above reactive control task, the second type of experiments involves internal states: to develop a locomotion controller for a walking robot. It has been shown that animals walk in a stereotyped way by adopting rhythmic patterns of movement, and the neural control of these stereotyped movements is hierarchically organized. It has also become clear that these kinds of motion patterns are controlled by the rhythm generator mechanism called central pattern generator (CPG) that provides feedforward signals needed for locomotion even in the absence of sensory feedback and high-level control [16]. In this experiment, the proposed GRN model works as the role of CPG to control the motors of the robot legs.

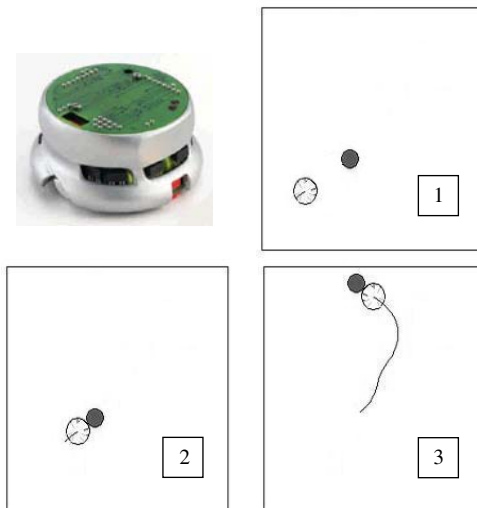


Fig. 4. The robot and its typical behavior

As the main goal here is to examine whether the locomotion task can be achieved by the presented approach, we did not build a walking robot but simply took the activation data of the successful controller developed in [17] as the time-series data for the inference of a GRN model. The robot used in their work has six legs that may be either up or down, and the robot can swing the leg forward or backward when it is up. Each leg has a limited angular motion that can be used as sensory information of the leg controller. To simplify the control architecture for a six legs insect, they assumed that the locomotion controller exhibits left-right and front-back symmetries. Therefore, only one set of leg controller parameters needs to be determined, and then these parameters can be copied to each of the six legs. More details of the robot can be found in [17].

To obtain a controller with locomotion behavior, in our experiments a GRN model with five gene nodes was used to control a leg: one node for foot control, two nodes for forward and backward swing control, and two nodes as inter-genes. The proposed approach was employed again to derive controllers. Fig. 5 shows the results. The left part of Fig. 5 is the activation data produced by a typical leg controller reported in [17], in which the x-axis represents the time steps and y-axis, the normalized activations of the actuators (from top to bottom: foot, swing forward, and swing backward). The inferred results are presented in the right part of Fig. 5, that show the controllers can be inferred successfully.

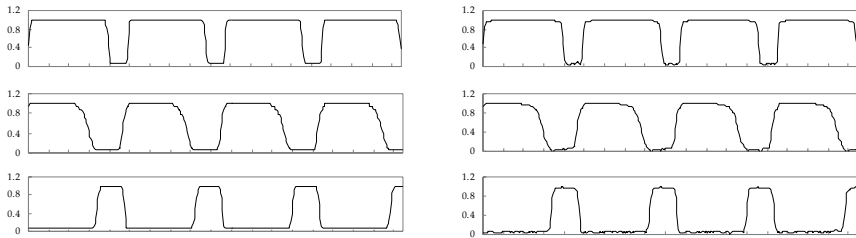


Fig. 5. The activation data of the desired (left) and inferred (right) leg controllers

4 Conclusions and Future Work

GRNs have been shown to play important roles in the development of living organism and they can be modeled to work as robot controllers. Depending on the biological levels to be studied, for example the qualitative network behavior or system dynamics, different GRN models have been proposed. Taking biological, computational and engineering perspectives, here we have derived a recurrent neural network-based regulatory model for robot control. We have also implemented a learning mechanism and designed a procedure of programming by demonstration to infer robot controllers from time series data. To verify the proposed methodology, different sets of experiments have been conducted for GRN modeling and robot control. The results have shown that our approach can be successfully used to infer GRN controllers from time series data obtained from a demonstration procedure to control a mobile robot and a walking robot.

Currently we are extending our work toward several directions. The first is to use the proposed approach to construct controllers for tasks involving more internal states. In those cases, the controllers can take the advantage of their feedback links to operate in the environment with and without sensory information. Meanwhile, it is worthwhile to investigate how to employ other types of learning algorithms to improve the modeling performance. Another direction is to deal with the scalability problem in which more network nodes and connections will be needed eventually for more complicated control tasks. We plan to develop and integrate an efficient clustering scheme into our framework to construct networks in a hierarchical way.

References

1. Davison, E.H., Rast, J.P., Oliveri, P., et al.: A genomic regulatory network for development. *Science* 295, 1669–1678 (2002)
2. de Jong, H.: Modeling and simulation of genetic regulatory systems: a literature review. *Journal of Computational Biology* 9, 67–103 (2002)
3. Karlebach, G., Shamir, R.: Modelling and analysis of gene regulatory networks. *Nature Review Molecular Cell Biology* 9, 770–780 (2008)
4. Quick, T., Nehaniv, C.L., Dautenhahn, K., et al.: Evolving embodied genetic regulatory network-driven control systems. In: *Proceeding of the Seventh European Conference on Artificial Life*, pp. 266–277 (2003)
5. Stewart, F., Taylor, T., Konidaris, G.: METAMorph: Experimenting with genetic regulatory networks for artificial development. In: *Proceeding of the Eighth European Conference on Artificial Life*, pp. 108–117 (2005)
6. Bentley, P.J.: Controlling robots with fractal gene regulatory networks. In: de Castro, L., von Zuben, F. (eds.) *Recent Developments in Biologically Inspired Computing*, pp. 320–339 (2005)
7. Vohradsky, J.: Neural network model of gene expression. *The FASEB Journal* 15, 846–854 (2001)
8. Werbos, P.J.: Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE* 78, 1550–1560 (1990)
9. Jacobs, R.A.: Increased rates of convergence through learning rate adaptation. *Neural Networks* 1, 295–307 (1988)
10. Rizzolatti, G., Craighero, L.: The mirror-neuron system. *Annual Review of Neuroscience* 27, 169–192 (2004)
11. Dillmann, R.: Teaching and learning of robot tasks via observation of human performance. *Robotics and Autonomous Systems* 47, 109–116 (2004)
12. Nakaoka, S., Nakazawa, A., Kanehiro, F., et al.: Learning from observation paradigm: leg task models for enabling a biped humanoid robot to imitate human dance. *International Journal of Robotics Research* 26, 829–844 (2007)
13. Nolfi, S., Floreano, D.: *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machine*. MIT Press, Cambridge (2000)
14. Japkowicz, N., Stephen, S.: The class imbalance problem: a systematic study. *Intelligent Data Analysis* 6, 429–450 (2002)
15. Sun, Y., Kamel, M., Wong, A., et al.: Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition* 40, 3358–3378 (2007)
16. Ijspeert, A.J.: Central pattern generators for locomotion control in animals and robots: a review. *Neural Networks* 21, 642–653 (2008)
17. Beer, R.D., Gallagher, J.C.: Evolving dynamical neural networks for adaptive behavior. *Adaptive Behavior* 1, 91–122 (1992)

Fault Detection for Networked Control Systems via Minimum Error Entropy Observer

Jianhua Zhang², Lei Cai², and Hong Wang¹

¹ The University of Manchester, PO Box 88, Manchester, M60, 1QD, UK

² Beijing Key Laboratory of Industrial Process Measurement & Control New Technology and System, North China Electric Power University, Beijing 102206, China

Abstract. In this paper, a novel fault detection scheme is presented for networked control systems with random delays and noises. Since the random noises and delays existed in networked control system are probably non-Gaussian, the extended Luenberger observer under minimum error entropy frame is utilized to generate residual so as to detect faults in networked control systems. Finally, an illustrative example is given to demonstrate the effectiveness of the proposed method.

Keywords: Networked control systems, fault detection, neural networks.

1 Introduction

Networked control systems (NCSs) are spatially distributed closed-loop feedback control systems in which the communication between the nodes occurs through a shared band-limited digital communication network [1]. The primary advantages of NCSs are reduced system wiring, ease of installation and maintenance, and increased system flexibility. However, the insertion of communication network in the feedback control loop makes the analysis and design of an NCS become more complex.

Recently, more and more attention has been paid on studying fault detection of the NCSs, and some approaches to fault diagnosis of NCSs were presented [2-15]. The method based on structure matrix of network-induced delays was presented in [3-5], the existed fault detection theory of time-delay systems was extended to NCSs [6-9], the approach based on quasi Takagi-Sugeno fuzzy model was proposed in [10] and observer-based fault detection was employed in [11]. In addition, considering the randomness of NCSs, robust fault detection filter was designed based on discrete Markovian jump systems in [12], the approach based on low-pass post-filtering was also presented in [13-15].

The delay characteristics on NCSs are stochastic in most cases, even the induced delays may not obey Gaussian distribution. The influence of network-induced delay on conventional observer based fault detection was evaluated in [16], the problem of observer based fault detection for NCSs has not fully investigated yet. It is necessary to study the approach to fault detection of NCSs with random delays.

Following the recent developments on the information theoretic learning in the signal processing community, adaptive observer under generalized entropy criterion

framework was proposed [19]. In this paper, we employ a minimum error entropy (MEE) observer to generate residuals in order to detect faults of NCSs, since there are uncertain components (e.g. uncertain time-delay, disturbance or bandwidth). Compared with the conventional observer, the MEE observer captures all the information in the error sequences in nonlinear dynamic systems with non-Gaussian noises and delays besides first and second order statistical information.

The rest of this paper is organized as follows: Section 2 gives the system description and formulates the problem of fault detection for NCSs. Section 3 utilizes MEE Luenberger observer to generate residual. Section 4 gives the detection threshold and the rule of evaluating residual. Section 5 verifies the efficiency of the method by an illustrative example. And the last section concludes the paper.

2 System Description and Problem Formulation

Consider a NCS whose dynamics can be characterized by following linear time invariant (LTI) discrete model

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + B_{f_a} f_a(k) + B_\omega \omega(k) \\ z(k) &= Cx(k) + D_{f_s} f_s(k) + D_v v(k) \\ y(k) &= z(k - \tau_{sc}(k)) \end{aligned} \quad (1)$$

where $x \in R^n$ denotes the state vector, $u \in R^m$ the control input vector and $y \in R^l$ the measurable output vector. $\omega \in R^p$ and $v \in R^p$ are system noise and measurement noise respectively. $f_a \in R^q$ and $f_s \in R^q$ are unknown actuator fault and sensor fault respectively. $\tau_{sc} \in \mathbf{Z}^+$ is the network-induced delay from sensor to controller respectively, $A, B, C, B_{f_a}, B_{f_s}, B_\omega, B_v$ are known real matrices of appropriate dimensions.

For the NCS, we introduce the following assumptions:

- 1) The actuators and sensors are time-driven, while the controller is event-driven.
- 2) The data packets are transmitted in right order without packet dropout.
- 3) A is nonsingular and (A, C) is observable.

Both network-induced delays and the noise in NCSs are probably non-Gaussian, the conventional observer cannot be employed to generate residual for detecting faults [3]. An alternative criterion is needed to extend observer so as to achieve optimality. It is natural to replace mean-square-error (MSE) with MEE, since entropy can capture all the information in the error sequences. The objective here is to investigate a fault detection method for NCSs within entropy criterion framework. The fault detection scheme in the NCS context is shown as Fig.1, in which τ_{ca} is the network-induced delay from controller to actuator. The procedure of fault detection consists of residual generation and residual evaluation.

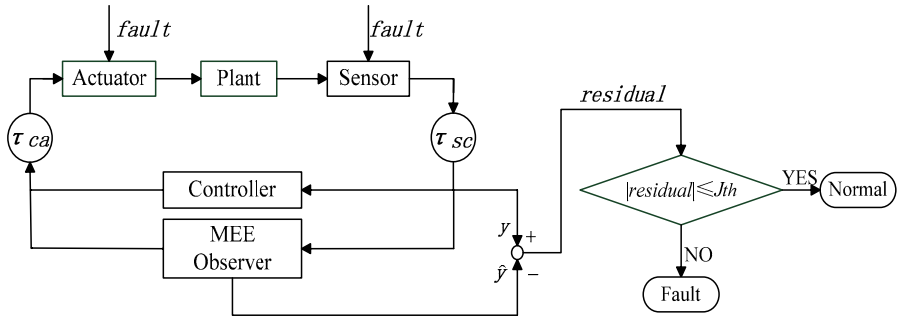


Fig. 1. Scheme of fault detection for NCSs

3 Design of Residual Generator

Luenberger proposed state observer to deal with state estimation for deterministic multivariable dynamic systems. For stochastic NCSs, the objective here is to design a residual generator by utilizing the following extended Luenberger observer which is specified by

$$\begin{aligned}
 \hat{x}(k+1) &= A\hat{x}(k) + Bu(k) + L(y(k) - \hat{y}(k)) \\
 \hat{z}(k) &= C\hat{x}(k) \\
 \hat{y}(k) &= \hat{z}(k - \tau_{sc}(k))
 \end{aligned} \tag{2}$$

where $\hat{x}(k) \in R^n$ and $\hat{y}(k) \in R^l$ are state and output estimation respectively. The observer gain matrix $L \in R^{n \times l}$ is updated using information theoretic learning technique during estimating state such that the entropy of the error $e(k) = y(k) - \hat{y}(k)$ between the measured and estimated output is minimized at each instant [18].

Minimum Renyi entropy of a random variable means minimizing its uncertainties, and all the moments of probability density function of the random variable are constrained [16]. In this paper, the quadratic Renyi entropy $\hat{H}_2(e)$ of error $e(k)$ is employed to be the performance index for design adaptive gain of the observer, and it can be obtained directly from data samples $\{e_1(k), e_2(k), \dots, e_{\bar{N}}(k)\}$ by utilizing Parzen windowing with kernel function. $\kappa_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$

$$\hat{H}_2(e) = -\log \left[\frac{1}{\bar{N}^2} \sum_{i=1}^{\bar{N}} \sum_{j=1}^{\bar{N}} \kappa_\sigma(e(i) - e(j)) \right] \tag{3}$$

Although the above calculation could be done through the Monte Carlo method, it is preferred to find a more practical solution to handle online, instantaneous computation for state estimation. The on-line estimation of quadratic Renyi entropy at instant k in a non-stationary environment can be obtained using a sliding window of samples [21].

$$\hat{H}_2(e) = -\log \left[\frac{1}{W} \sum_{i=k-W}^{k-1} \kappa_\sigma(e(i) - e(k)) \right] \quad (4)$$

$\hat{V}(e) = \frac{1}{W} \sum_{i=k-W}^{k-1} \kappa_\sigma(e(i) - e(k))$ is defined as information potential of error in [18], W the width of sliding window. Minimizing the entropy of error $\hat{H}_2(e)$ is equivalent to maximizing the information potential $\hat{V}(e)$ due to monotonically increasing “log” function. The performance index used to update the extended Luenberger observer gain L would be the stochastic quadratic information potential of error, i.e.

$$J = \frac{1}{W} \sum_{i=k-W}^{k-1} \kappa_\sigma(e(i) - e(k)) \quad (5)$$

the updated rule for the observer gain L can be obtained by stochastic gradient algorithm. Denote $L = [L_{:,1} \quad L_{:,2} \quad \dots \quad L_{:,t}] \in R^{n \times t}$, where $L_{:,j} \in R^{n \times 1}$, $j = 1, 2, \dots, t$. Applying the chain rule and taking the derivates of performance criteria J with respect to the observer gain L ,

$$\frac{\partial J}{\partial L(k)_{:,j}} = \frac{1}{W} \sum_{i=k-W}^{k-1} \kappa'_\sigma(e(i) - e(k)) \cdot C \cdot \left(\frac{\partial \hat{x}(k - \tau_{sc}(k))}{\partial L(k)_{:,j}} - \frac{\partial \hat{x}(i - \tau_{sc}(i))}{\partial L(i)_{:,j}} \right) \quad (6)$$

where $\kappa'_\sigma(e(i) - e(k)) = \frac{-(e(i) - e(k))}{\sigma^2} \cdot \kappa_\sigma(e(i) - e(k))$,

$$\begin{aligned} & \frac{\partial \hat{x}(k - \tau_{sc}(k))}{\partial L(k)_{:,j}} \\ &= \frac{\partial \hat{x}(k - \tau_{sc}(k))}{\partial \hat{x}(k)} \cdot \frac{\partial \hat{x}(k)}{\partial L(k)_{:,j}} \\ &= \frac{\partial \hat{x}(k - \tau_{sc}(k))}{\partial \hat{x}(k - \tau_{sc}(k) + 1)} \cdot \frac{\partial \hat{x}(k - \tau_{sc}(k) + 1)}{\partial \hat{x}(k - \tau_{sc}(k) + 2)} \dots \frac{\partial \hat{x}(k - 1)}{\partial \hat{x}(k)} \cdot \frac{\partial \hat{x}(k)}{\partial L(k)_{:,j}} \\ &\approx \frac{\partial \hat{x}(k)}{\partial L(k)_{:,j}} = (A - L \cdot C) \frac{\partial \hat{x}(k - 1)}{\partial L(k - 1)_{:,j}} + I_{n \times n} \cdot (y(k - 1) - \hat{y}(k - 1)) \end{aligned}$$

The observer gain L can be then updated as follows

$$L_{:,j}(k+1) = L_{:,j}(k) + \eta \cdot \frac{\partial J(k)}{\partial L_{:,j}(k)} \quad (7)$$

where η is step size for adaptation. It should be pointed out that the observer can be implemented by an ADALINE (Adaptive Linear Neuron) without a bias weight, whose weights form the observer gain.

In this paper, the residual signal generator can be constructed as

$$r(k) = y(k) - \hat{y}(k) \tag{8}$$

where the error $r(k)$ between the real output and observer output in the NCS is utilized to detect faults in the NCS.

4 Fault Detection

In order to trigger the fault alarm, we have to select a threshold which is the maximum of the residual in normal NCSs. In this paper, the following threshold [20] is employed

$$J_{th} = E\left\{\sum_{k=0}^N r^T(k)r(k)\right\}^{\frac{1}{2}} \tag{9}$$

where N is the length pre-described for the expectation operator.

Based on the threshold selection, the occurrence of faults can be detected by comparing $r(k)$ and J_{th} according to the following principle:

$$\begin{cases} |r(k)| \leq J_{th}, & \text{no fault occurs} \\ |r(k)| > J_{th} & \text{fault has occurred} \end{cases}$$

5 Illustrative Example

A numerical example is given to show the performance of the proposed method. Considering the discrete linear time-invariable NCS as follows

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 0 & 1 \\ -1 & -0.25 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0.01 \end{bmatrix} u(k) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} f_a(k) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \omega(k) \\ z(k) &= [1 \quad 0.01] x(k) + f_s(k) + v(k) \\ y(k) &= z(k - \tau_{sc}(k)) \end{aligned} \tag{10}$$

In this simulation, it is assumed that the control input is $u(k) = [-0.1 \quad -0.25]x(k)$, the probability density function of $\tau_{sc}(k)$ obeys the following β -distribution

$$\gamma_{\tau_{sc}}(x) = \begin{cases} [5^9 \beta(3,7)]^{-1} x^2 (5-x)^6, & x \in [0,5] \\ 0, & \text{otherwise} \end{cases}$$

where $\beta(a,b) = \int_0^1 x^{a-1} (1-x)^{b-1} dx$, both white noises $\omega(k), v(k)$ belong to $[-0.005, 0.005]$. The kernel size used to estimate the entropy is experimentally set to

$\sigma = 0.1$, the initial observer gain $L = [-0.05 \quad -0.15]^T$. The fault is imposed as follows

$$f_s(k) = \begin{cases} 0, & t < 150s \\ 1, & 150s < t < 200s \end{cases}$$

The simulation results are shown in Figs.2-5. Fig.2 shows the increasing information potential of the MEE Luenberger observer, in which the entropy of the observer output error decreases with time. Figs.3 and 4 show two state errors respectively. The convergent state errors indicate better performance of the observer. The residual $r(k)$ generated by MEE observer is shown in Fig.5, which illustrates effective ness of the proposed method despite the disturbances $\omega(k)$ and $\nu(k)$ in the NCS.

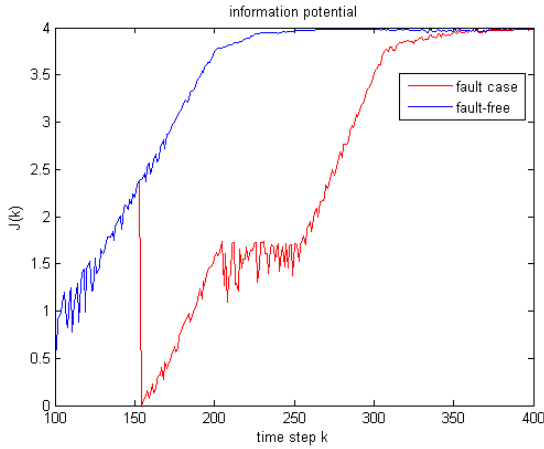


Fig. 2. Information potential of observer error

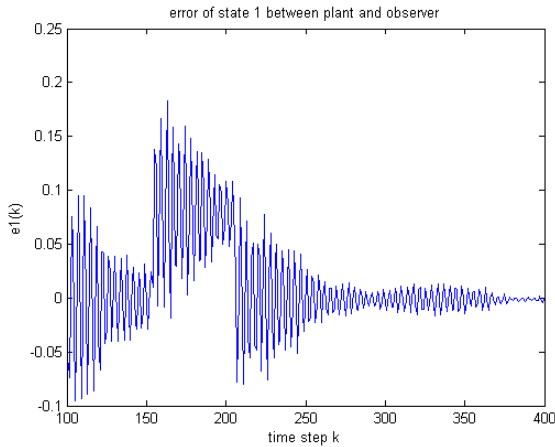


Fig. 3. Estimation error of state 1

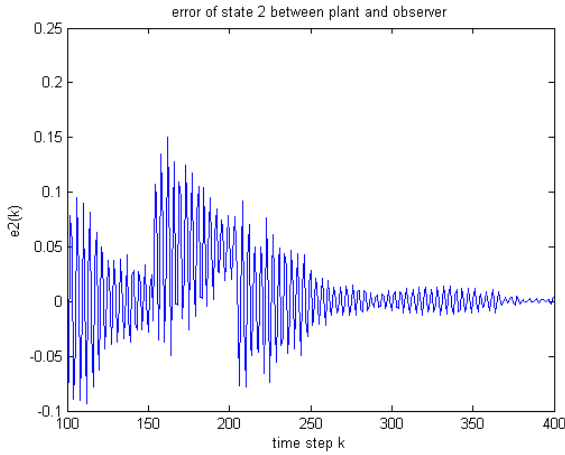


Fig. 4. Estimation error of state 2

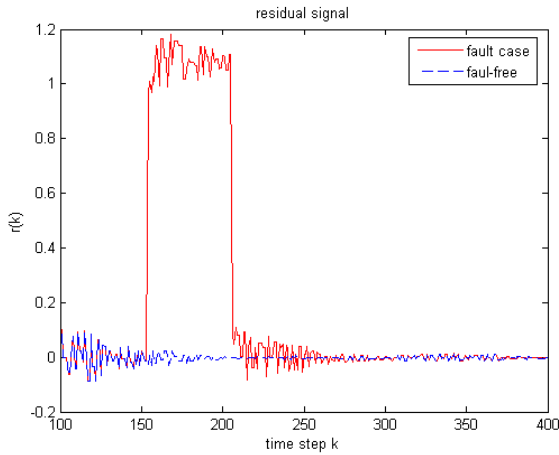


Fig. 5. Residual signal

6 Conclusions

This paper investigates the fault detection problem of NCSs with random network-induced delays and noises. Since the random delays and noises may not obey Gaussian distribution, the extended Luenberger observer is utilized to generate residual for detecting faults in NCSs. A time-variant threshold is employed to improve the sensitivity to the faults. The proposed method may be generalized to nonlinear NCSs in further research.

Acknowledgments. This work is supported by National Natural Science Foundation of China under grants (No. 60674051) and Beijing Natural Science Foundation under grant (No. 4072022). These are gratefully acknowledged.

References

1. Hespanha, J.P., Naghshtabrizi, P., Xu, Y.: A Survey of Recent Results in Networked Control Systems. *Proceedings of the IEEE* 95, 138–162 (2007)
2. Fang, H., Ye, H., Zhong, M.: Fault Diagnosis of Networked Control Systems. *Annual Reviews in Control* 31, 55–68 (2007)
3. Ye, H., Ding, S.X.: Fault Detection of Networked Control Systems with Network-Induced Delay. In: *Proceedings of the 8th International Conference on Control, Automation, Robotics and Vision (ICARCV 2004)*, Kunming, China, pp. 294–297 (2004)
4. Wang, Y., Ye, H., Cheng, Y., Wang, G.: Fault Detection of NCS Based on Eigendecomposition and Pade Approximation. In: *Preprints of 6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, Beijing, China, pp. 937–941 (2006)
5. Ye, H., He, R., Liu, H., Wang, G.: A New Approach for Fault Detection of Networked Control Systems. In: *Proceedings of IFAC 14th Symposium on System Identification*, Newcastle, Australia, pp. 654–659 (2006)
6. Yang, H.L., Saif, M.: Observer Design and Fault Diagnosis for State Retarded Dynamical Systems. *Automatica* 34, 217–227 (1998)
7. Zhong, M., Ye, H., Sun, T., Wang, G.: An iterative LMI Approach to Robust Fault Detection Filter for Linear System with Time-varying Delays. *Asian Journal of Control* 8, 86–90 (2006)
8. Ding, S.X., Jeansch, T., Frank, P.M., Ding, E.L.: A Unified Approach to the Optimization of Fault Detection Systems. *International Journal of Adaptive Contribution and Signal Processing* 14, 725–745 (2000)
9. Chen, W., Saif, M.: An Iterative Learning Observer for Fault Detection and Accommodation in Nonlinear Time-delay Systems. *International Journal of Robust and Nonlinear Control* 16, 1–19 (2006)
10. Zheng, Y.: Fault Diagnosis and Fault Tolerant Control of Networked Control Systems. Ph.D Dissertation, Huazhong University of Science and Technology, China (2003)
11. Zhong, M., Liu, Y., Ma, C.: Observer-Based Fault Detection for Networked Control Systems with Random Time-delays. In: *Proceedings of International Conference on Intelligent Computing (ICIC)*, Kunming, China, pp. 528–531 (2006)
12. Mao, Z., Jiang, B., Shi, P.: H_∞ Fault Detection Filter Design for Networked Control Systems Modelled by Discrete Markovian Jump Systems. *IET control theory & applications* 1, 1336–1343 (2007)
13. Ye, H., Zhang, P., Ding, S.X., Wang, G.: A Time–frequency Domain Fault Detection Approach Based on Parity Relation and Wavelet Transform. In: *Proceedings of the 39th IEEE Conference on Decision and Control (IEEE CDC 2000)*, Sydney, Australia, pp. 4156–4161 (2000)
14. Zhang, P., Ye, H., Ding, X., Wang, G., Zhou, D.: On the Relationship between Parity Space and H_2 Approaches to Fault Detection. *Systems & Control Letters* 55, 94–100 (2006)
15. Ye, H., Wang, Y.: Application of Parity Relation and Stationary Wavelet Transform to Fault Detection of Networked Control Systems. In: *Proceedings of the 1st IEEE Conference on Industrial Electronics and Applications (ICIEA 2006)*, Singapore, pp. 1–6 (2006)

16. Erdogmus, D., Principe, J.C.: An Error-entropy Minimization Algorithm for Supervised Training of Nonlinear Adaptive Systems. *IEEE Transactions on Signal Processing* 50, 1780–1786 (2002)
17. Parzen, E.: On Estimation of a Probability Density Function and Mode. *Time Series Analysis Papers*, Holden-Day, CA (1967)
18. Principe, J.C., Xu, D., Fisher, J.: Unsupervised Adaptive Filtering: Information Theoretic Learning. In: Haykin, S. (ed.), vol. 1, pp. 265–319. Wiley, New York (2000)
19. Xu, J., Erdogmus, D., Principe, J.: Minimum Error Entropy Luenberger Observer. In: *Proceedings of American Control Conference*, Portland, USA, pp. 1923–1928 (2005)
20. Chen, Y., Duan, Z., Liu, Y., Wang, R.: Fault Detection for Networked Control System with Random Delays. In: *Proceedings of IEEE Conference on Robotics, Automation and Mechatronics*, Chengdu, China, pp. 95–100 (2008)
21. Erdogmus, D., Principe, J.C., Kim, S.P., Sanchez, J.C.: A Recursive Renyi's Entropy Estimator. In: *Proceedings of IEEE Neural Networks for Signal Processing*, Martigny, Valais, Switzerland, pp. 209–217 (2002)

Discrete-Time Reduced Order Neural Observers

Alma Y. Alanis¹ and Edgar Nelson Sanchez²

¹ Departamento de Ciencias Computacionales, CUCEI, Universidad de Guadalajara, Apartado Postal 51-71, Col. Las Aguilas, Zapopan, Jalisco, C.P. 45080, Mexico

almayalanis@gmail.com

² CINVESTAV, Unidad Guadalajara, Apartado Postal 31-438, Plaza La Luna, Guadalajara, Jalisco, C.P. 45091, Mexico

Abstract. A nonlinear discrete-time reduced order neural observer for the state estimation of a discrete-time unknown nonlinear system, in presence of external and internal uncertainties is presented. The observer is based on a discrete-time recurrent high order neural network (RHONN) trained with an extended Kalman filter (EKF)-based algorithm. This observer estimates the state of the unknown discrete-time nonlinear system, using a parallel configuration. To illustrate the applicability simulation results are included.

Keywords: Reduced order neural observers, Recurrent high order neural networks, Kalman filtering learning, Discrete-time nonlinear systems, Van der Pol oscillator.

1 Introduction

Modern control systems usually require a very structured knowledge about the system to be controlled; such knowledge should be represented in terms of differential or difference equations. This mathematical description of the dynamic system is named as the model. There can be several motives for establishing mathematical descriptions of dynamic systems, such as: simulation, prediction, fault detection, and control system design.

Basically there are two ways to obtain a model; it can be derived in a deductive manner using physics laws, or it can be inferred from a set of data collected during a practical experiment. The first method can be simple, but in many cases is excessively time-consuming; it would be unrealistic or impossible to obtain an accurate model in this way. The second method, which is commonly referred as system identification [26], could be a useful short cut for deriving mathematical models. Although system identification not always results in an accurate model, a satisfactory model can be often obtained with reasonable efforts. The main drawback is the requirement to conduct a practical experiment, which brings the system through its range of operation [4, 16].

Many of the nonlinear control publications assume complete accessibility for the system state; this is not always possible. For this reason, nonlinear state estimation is a very important topic for nonlinear control [17]. State estimation has

been studied by many authors, who have obtained interesting results in different directions. Most of those results need the use of a special nonlinear transformation [15] or a linearization technique [5], [9]. Such approaches can be considered as a relatively simple method to construct nonlinear observers; however, they do not consider uncertainties [11], [12], [25]. In practice, there exist external and internal uncertainties. Observers which have a good performance even in presence of model and disturbance uncertainties, are called robust; their design process is too complex [2], [3], [7], [24]. All the approaches mentioned above need the previous knowledge of the plant model, at least partially. Recently, other kind of observers has emerged: neural observers [1], [8], [10], [14], [17], [19], for unknown plant dynamics. However, all the approaches mentioned above estimates all the state, including the measurable variables, in order to reduce the computational complexity, in this paper we propose the use of a reduced order neural observer, which uses the available measurement and only estimates the unmeasurable variables, besides the proposed observer provides a mathematical model for unknown nonlinear systems.

Neural networks have grown to be a well-established methodology, which allows for solving very difficult problems in engineering, as exemplified by their applications to identification and control of general nonlinear and complex systems. In particular, the use of recurrent neural networks for modelling and learning has rapidly increased in recent years ([19] and references there in).

There exist different training algorithms for neural networks, which, however, normally encounter some technical problems such as local minima, slow learning, and high sensitivity to initial conditions, among others. As a viable alternative, new training algorithms, e.g., those based on Kalman filtering, have been proposed [5], [6], [21], [26]. Due to the fact that training a neural network typically results in a nonlinear problem, the Extended Kalman Filter (EKF) is a common tool to use, instead of a linear Kalman filter [6].

As is well known [19], recurrent high order neural networks (RHONN) offer many advantages for modelling of complex nonlinear systems. On the other hand EKF training for neural networks allows to reduce the epoch size and the number of required neurons [6]. Considering these two facts, we propose the use of the EKF training for RHONN in order to model complex nonlinear systems.

The best well-known training approach for recurrent neural networks (RNN) is the back propagation through time learning [21]. However, it is a first order gradient descent method and hence its learning speed could be very slow [21]. Recently the Extended Kalman Filter (EKF) based algorithms has been introduced to train neural networks, in order to improve the learning convergence [21]. The EKF training of neural networks, both feedforward and recurrent ones, has proven to be reliable and practical for many applications over the past ten years [21].

In this paper, a recurrent high order neural network (RHONN) is used to design of an adaptive recurrent neural observer for nonlinear systems, whose mathematical model is assumed to be unknown. The learning algorithm for the RHONN is implemented using an Extended Kalman Filter (EKF). The applicability of these schemes is illustrated via simulation for a van der Pol oscillator.

In this paper, we consider a class of MIMO discrete-time nonlinear system, for which we develop a reduced order Luenberger-like observer [17]; then this observer is applied to a discrete-time unknown nonlinear system [13]. This observer is based on a recurrent high order neural network (RHONN) [20], which estimates the state vector of the unknown plant dynamics. The learning algorithm for the RHONN is based on an extended Kalman filter (EKF) ([1], [18] and references therein).

2 Preliminaries

2.1 Nonlinear Reduced Order Observers

We now know how to design a full state observer for an observable system [23]. The full state observer adds n first order equations to the n existing first order equations and thus doubles the complexity of the system. If each component of the measurement $y(t)$ is independent of the others (i.e., if C has full rank m), then we really only need to estimate $r = n - m$ components of the system state. We start by considering the special case where the components of y are simply components of x . (The more general case, in which the components of y are nonlinear combinations of the components of x , turns out to be a simple extension of this special case.) Without loss of generality, we may order the states such that

$$y(k) = Cx(k) = \begin{pmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_p(k) \end{pmatrix}$$

with $C = [\mathbf{I}_p 0_{p \times r}]$, $p + r = n$, $x \in \mathbb{R}^n$, $y \in \mathbb{R}^p$. It is convenient to define

$$x_a(k) = \begin{pmatrix} x_1(k) \\ \vdots \\ x_p(k) \end{pmatrix}$$

the measurable variables ($x_a \in \mathbb{R}^p$), and ($x_b \in \mathbb{R}^r$) the unmeasurable variables

$$x_b(k) = \begin{pmatrix} x_{p+1}(k) \\ \vdots \\ x_n(k) \end{pmatrix}$$

Then the complete equations may be written

$$\begin{pmatrix} x_a(k+1) \\ x_b(k+1) \end{pmatrix} = \begin{pmatrix} f_a(x_a(k), x_b(k), u(k)) \\ f_b(x_a(k), x_b(k), u(k)) \end{pmatrix}$$

$$y(k) = [\mathbf{I}_p, 0_{p \times r}] \begin{pmatrix} x_a(k) \\ x_b(k) \end{pmatrix}$$

Written individually, the state equations are

$$\begin{aligned} x_a(k+1) &= f_a(x_a(k), x_b(k), u(k)) \\ x_b(k+1) &= f_b(x_a(k), x_b(k), u(k)) \\ y(k) &= x_a(k) \end{aligned} \tag{1}$$

where $u(k) \in \mathfrak{R}^m$ is the input vector, $f_a(\bullet)$ and $f_b(\bullet)$ are unknown nonlinear functions. Then, as x_a are the measurable variables, only it is necessary design an observer for x_b , notice that the dimension of the observer dynamic equations is $r = n - p$. For this reason, this is referred to as a *reduced order observer* [23].

2.2 Discrete-Time Recurrent High Order Neural Networks

Consider the following discrete-time recurrent high order neural network (RHONN):

$$x_i(k+1) = w_i^\top z_i(x(k), u(k)), \quad i = 1, \dots, n \tag{2}$$

where x_i ($i = 1, 2, \dots, n$) is the state of the i th neuron, L_i is the respective number of higher-order connections, $\{I_1, I_2, \dots, I_{L_i}\}$ is a collection of non-ordered subsets of $\{1, 2, \dots, n\}$, n is the state dimension, w_i ($i = 1, 2, \dots, n$) is the respective on-line adapted weight vector, and $z_i(x(k), u(k))$ is given by

$$z_i(x(k), u(k)) = \begin{bmatrix} z_{i1} \\ z_{i2} \\ \vdots \\ z_{iL_i} \end{bmatrix} = \begin{bmatrix} \prod_{j \in I_1} y_{ij}^{d_{ij}(1)} \\ \prod_{j \in I_2} y_{ij}^{d_{ij}(2)} \\ \vdots \\ \prod_{j \in I_{L_i}} y_{ij}^{d_{ij}(L_i)} \end{bmatrix} \tag{3}$$

with $d_{j_i}(k)$ being a nonnegative integers, and y_i is defined as follows:

$$y_i = \begin{bmatrix} y_{i1} \\ \vdots \\ y_{i1} \\ y_{in+1} \\ \vdots \\ y_{in+m} \end{bmatrix} = \begin{bmatrix} S(x_1) \\ \vdots \\ S(x_n) \\ u_1 \\ \vdots \\ u_m \end{bmatrix} \tag{4}$$

In (4), $u = [u_1, u_2, \dots, u_m]^\top$ is the input vector to the neural network, and $S(\bullet)$ is defined by

$$S(x) = \frac{1}{1 + \exp(-\beta x)} \tag{5}$$

Consider the problem to approximate the general discrete-time nonlinear system (1), by the following discrete-time RHONN [20]:

$$\chi_i(k+1) = w_i^{*\top} z_i(x(k), u(k)) + \epsilon_{z_i}, \quad i = 1, \dots, n \tag{6}$$

where χ_i is the i th plant state, ϵ_{z_i} is a bounded approximation error, which can be reduced by increasing the number of the adjustable weights [20]. Assume that there exists ideal weights vector w_i^* such that $\|\epsilon_{z_i}\|$ can be minimized on a compact set $\Omega_{z_i} \subset \mathfrak{R}^{L_i}$. The ideal weight vector w_i^* is an artificial quantity required for analytical purpose [20]. In general, it is assumed that this vector exists and is constant but unknown. Let us define its estimate as w_i and the estimation error as

$$\tilde{w}_i(k) = w_i^* - w_i(k) \quad (7)$$

The estimate w_i is used for stability analysis which will be discussed later. Since w_i^* is constant, then $\tilde{w}_i(k+1) - \tilde{w}_i(k) = w_i(k+1) - w_i(k)$, $\forall k \in 0 \cup \mathbb{Z}^+$.

2.3 The EKF Training Algorithm

It is known, that Kalman filtering (KF) estimates the state of a linear system with additive state and output white noises [5], [22]. For KF-based neural network training, the network weights become the states to be estimated. In this case the error between the neural network output and the measured plant output can be considered as additive white noise. Due to the fact that the neural network mapping is nonlinear, an EKF-type is required (see [18] and references therein). The training goal is to find the optimal weight values which minimize the prediction error. In this paper, we use a EKF-based training algorithm described by

$$w_i(k+1) = w_i(k) + \eta_i K_i(k) e_i(k) \quad (8)$$

$$K_i(k) = P_i(k) H_i(k) M_i(k)$$

$$P_i(k+1) = P_i(k) - K_i(k) H_i^T(k) P_i(k) + Q_i(k), \quad i = 1, \dots, n$$

with

$$M_i(k) = [R_i(k) + H_i^T(k) P_i(k) H_i(k)]^{-1} \quad (9)$$

$$e_i(k) = y(k) - \hat{y}(k) \quad (10)$$

where $e(k) \in \mathfrak{R}^p$ is the observation error and $P_i(k) \in \mathfrak{R}^{L_i \times L_i}$ is the weight estimation error covariance matrix at step k , $w_i \in \mathfrak{R}^{L_i}$ is the weight (state) vector, L_i is the respective number of neural network weights, $y \in \mathfrak{R}^p$ is the plant output, $\hat{y} \in \mathfrak{R}^p$ is the NN output, n is the number of states, $K_i \in \mathfrak{R}^{L_i \times p}$ is the Kalman gain matrix, $Q_i \in \mathfrak{R}^{L_i \times L_i}$ is the NN weight estimation noise covariance matrix, $R_i \in \mathfrak{R}^{p \times p}$ is the error noise covariance, and $H_i \in \mathfrak{R}^{L_i \times p}$ is a matrix, in which each entry (H_{ij}) is the derivative of the i th neural output with respect to ij th neural network weight, (w_{ij}), given as follows:

$$H_{ij}(k) = \left[\frac{\partial x_i(k)}{\partial w_{ij}(k)} \right]^T \quad (11)$$

where $i = 1, \dots, n$ and $j = 1, \dots, L_i$. Usually P_i and Q_i are initialized as diagonal matrices, with entries $P_i(0)$ and $Q_i(0)$, respectively. It is important to remark that $H_i(k)$, $K_i(k)$ and $P_i(k)$ for the EKF are bounded; for a detailed explanation of this fact see [22].

3 Discrete-Time Reduced Order Neural Observers

As explained above, $f_a(\bullet)$ and $f_b(\bullet)$ are unknown nonlinear functions, however x_a are the measurable variables, and x_b are the unmeasurable variables, then , in this section, we consider a neural identifier for x_a and a neural observer to estimate x_b , for system (11), which is assumed to be observable.

For system (11), we propose a reduced order neural observer (RONO) with the following structure:

$$\begin{aligned} \hat{x}(k) &= [\hat{x}_1(k) \dots \hat{x}_i(k) \dots \hat{x}_j(k) \dots \hat{x}_p(k)]^\top \\ \hat{x}_i(k+1) &= w_i^\top z_i(x_a(k), \hat{x}_b(k)u(k)) + g_i e(k) \\ \hat{x}_j(k+1) &= w_j^\top z_j(x_a(k), u(k)) + g_j e(k) \\ \hat{y}(k) &= C\hat{x}(k) \end{aligned} \tag{12}$$

with $i = 1, \dots, p, j = p + 1, \dots, n, L_i, L_j \in \mathbb{R}^p, w_i, w_j, z_i$ and z_j as in (2); the weight vectors are updated on-line with a decoupled EKF (8) – (11) and the output error is defined by

$$e(k) = y(k) - \hat{y}(k) \tag{13}$$

and the state estimation error as:

$$\tilde{x}(k) = x(k) - \hat{x}(k) \tag{14}$$

Considering (13) and (14)

$$e(k) = C\tilde{x}(k) \tag{15}$$

The proposed neural observer scheme is shown in Figure 1.

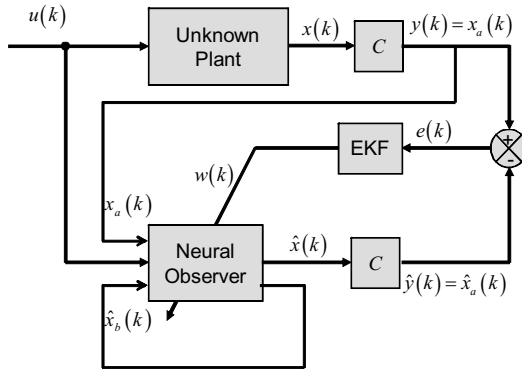


Fig. 1. Reduced order neural observer scheme

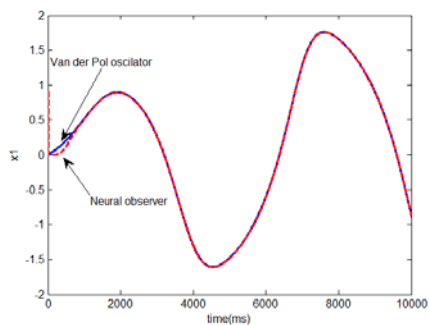


Fig. 2. Time evolution of the state $x_1(k)$ (solid line) and its estimated $\hat{x}_1(k)$ (dashed line)

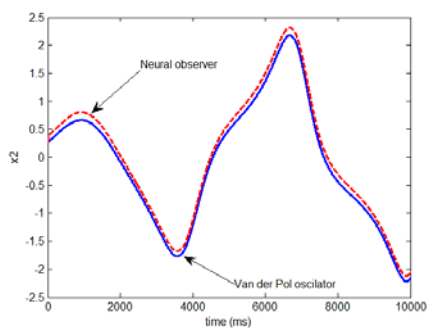


Fig. 3. Time evolution of the state $x_2(k)$ (solid line) and its estimated $\hat{x}_2(k)$ (dashed line)

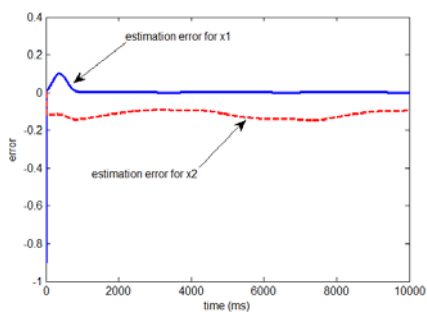


Fig. 4. Estimation errors $\tilde{x}_1(k)$ (solid line) and $\tilde{x}_2(k)$ (dashed line)

3.1 RONO for the Van Der Pol Oscillator

In this section, the neural observer is applied to a modified Van der Pol oscillator, whose nonlinear dynamics is represented by the following equation [27]:

$$\begin{aligned}
 x_1(k+1) &= x_1(k) + Tx_2(k) \\
 x_2(k+1) &= x_2(k) + T((0.5 - x_1^2(k))x_2(k) + T(-x_1(k) + u(k))) \\
 y(k) &= x_1(k) \\
 u(k) &= 0.5 \cos(1.1kT)
 \end{aligned} \tag{16}$$

where variables $x \in \mathfrak{R}^2$, $u \in \mathfrak{R}$, and $y \in \mathfrak{R}$ are the state, input, and output of the system, respectively and T is the sampling period, which is fixed at 1×10^{-3} s.

Simulation Results. To estimate the state x_2 , we use the RONO [12] with $n = 2$ trained with the EKF [8].

$$\begin{aligned}
 \hat{x}_1(k+1) &= w_{11}(k)S(\hat{x}_2(k))S(u(k)) + w_{12}(k)S^2(\hat{x}_2(k))S(x_1(k))S^2(u(k)) \\
 &\quad + w_{13}(k)S(x_1(k)) + g_1e(k) \\
 \hat{x}_2(k+1) &= w_{21}(k)S(x_1(k))S(\hat{x}_2(k))S(u(k)) + w_{22}(k)S(u(k))S(\hat{x}_2(k)) \\
 &\quad + w_{23}(k)S(u(k))S^2(\hat{x}_2(k)) + g_2e(k) \\
 \hat{y}(k) &= \hat{x}_1(k)
 \end{aligned} \tag{17}$$

The training is performed on-line, using a parallel configuration as displayed in Figure 1. All the NN states are initialized in a random way. The associated covariances matrices are initialized as diagonals, and the nonzero elements are: $P_1(0) = P_2(0) = 100000$; $Q_1(0) = Q_2(0) = 1000$ and $R_1(0) = R_2(0) = 10000$, respectively. The simulation results are presented in Figures 2 and 3. They display the time evolution of the estimated states $x_1(k)$ and $x_2(k)$, respectively. Figure 4. shows the estimation errors.

Remark 1. The purpose of this paper is to develop a reduced order neural observer for a class of MIMO nonlinear systems in discrete-time, by means of the use of the EKF as the neural network learning algorithm without the knowledge of a nominal plant model; this approach is validated by the simulation results presented above.

Remark 2. Even if the EKF is not an easy learning algorithm, it presents an excellent performance and has proven to be reliable and practical for many applications over the past ten years ([18] and references therein).

4 Conclusions

A RHONN is used to design a reduced order Luenberger-like observer for a class of MIMO discrete-time nonlinear system; this observer is trained with an EKF-based algorithm, which is implemented on-line as a parallel configuration.

Simulation results shows the effectiveness of the proposed RONO, as applied to a discrete-time Van der Pol oscillator. Currently, the authors are working on the stability analysis of the proposed scheme.

Acknowledgement. The authors thank the support of CONACYT Mexico, through Project 57801Y.

References

1. Alanis, A.Y., Sanchez, E.N., Loukianov, A.G.: Discrete time adaptive backstepping nonlinear control via high order neural networks. *IEEE Transactions on Neural Networks* 18(4), 1185–1195 (2007)
2. Chen, F., Dunnigan, M.W.: Comparative study of a sliding-mode observer and Kalman filters for full state estimation in an induction machine. In: *IEE Proceedings Electric Power Applications*, January 2002, vol. 149(1), pp. 53–64 (2002)
3. Coutinho, D.F., Pereira, L.P.F.A.: A robust Luenberger-like observer for induction machines. In: *Proceedings IEEE IECON 2005* (November 2005)
4. Farrell, J.A., Polycarpou, M.M.: *Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches*. John Wiley and Sons, New York (2006)
5. Grover, R., Hwang, P.Y.C.: *Introduction to Random Signals and Applied Kalman Filtering*, 2nd edn. John Wiley and Sons, New York (1992)
6. Haykin, S.: *Kalman Filtering and Neural Networks*. John Wiley and Sons, New York (2001)
7. Huang, H., Feng, G., Cao, J.: Robust state estimation for uncertain neural networks with time-varying delay. *IEEE Transactions on Neural Networks* 19(8), 1329–1339 (2008)
8. Kim, Y.H., Lewis, F.L.: *High-Level Feedback Control with Neural Networks*. World Scientific, Singapore (1998)
9. Krener, A.J., Isidori, A.: Linearization by output injection and nonlinear observers. *System and Control Letters* 3, 47–52 (1983)
10. Levin, A.U., Narendra, K.S.: Control of nonlinear dynamical systems using neural networks - part II: observability, identification and control. *IEEE transactions on neural networks* 7(1), 30–42 (1996)
11. Li, J., Zhong, Y.: Comparison of three Kalman filters for speed estimation of induction machines. In: *Proceedings Industry Applications Conference 2005* (October 2005)
12. Liu, Y., Wang, Z., Liu, X.: Design of exponential state estimators for neural networks with mixed time delays. *Physics Letters A* 364(5), 401–412 (2007)
13. Loukianov, A.G., Rivera, J., Cañedo, J.M.: Discrete-time sliding mode control of an induction motor. In: *Proceedings IFAC 2002*, Barcelona, Spain (July 2002)
14. Marino, R.: Observers for single output nonlinear systems. *IEEE Transactions on Automatic Control* 35, 1054–1058 (1990)
15. Nicosia, S., Tornambe, A.: High-gain observers in the state and parameter estimation of robots having elastic joints. *System and Control Letters* 13, 331–337 (1989)
16. Norgaard, M., Ravn, O., Poulsen, N.K., Hansen, L.K.: *Neural Networks for Modelling and Control of Dynamic Systems*. Springer, New York (2000)
17. Poznyak, A.S., Sanchez, E.N., Yu, W.: *Differential Neural Networks for Robust Nonlinear Control*. World Scientific, Singapore (2001)

18. Sanchez, E.N., Alanis, A.Y., Chen, G.: Recurrent neural networks trained with Kalman filtering for discrete chaos reconstruction. In: Proceedings of Asian-Pacific Workshop on Chaos Control and Synchronization 2004, Melbourne, Australia (July 2004)
19. Ricalde, L.J., Sanchez, E.N.: Inverse optimal nonlinear high order recurrent neural observer. In: International Joint Conference on Neural Networks, IJCNN 2005, Montreal, Canada (August 2005)
20. Rovithakis, G.A., Chistodoulou, M.A.: Adaptive Control with Recurrent High - Order Neural Networks. Springer, New York (2000)
21. Singhal, S., Wu, L.: Training multilayer perceptrons with the extended Kalman algorithm. In: Touretzky, D.S. (ed.) Advances in Neural Information Processing Systems, vol. 1, pp. 133–140. Morgan Kaufmann, San Mateo (1989)
22. Song, Y., Grizzle, J.W.: The extended Kalman Filter as Local Asymptotic Observer for Discrete-Time Nonlinear Systems. *Journal of Mathematical systems, Estimation and Control* 5(1), 59–78 (1995)
23. Utkin, V., Guldner, J., Shi, J.: Sliding Mode Control in Electromechanical Systems. Taylor and Francis, Philadelphia (1999)
24. Walcott, B.L., Zak, S.H.: State observation of nonlinear uncertain dynamical system. *IEEE Transactions on Automatic Control* 32, 166–170 (1987)
25. Wang, Z., Ho, D.W.C., Liu, X.: State estimation for delayed neural networks. *IEEE Transactions on Neural Networks* 16(1), 279–284 (2005)
26. Yu, W., Li, X.: Nonlinear system identification using discrete-time recurrent neural networks with stable learning algorithms. *Information Sciences* 158, 131–147 (2004)
27. Zhu, Q., Guo, L.: Stable adaptive neurocontrol for nonlinear discrete-time systems. *IEEE Transactions on Neural Networks* 15(3), 653–662 (2004)

A New Neural Observer for an Anaerobic Wastewater Treatment Process

Rubén Belmonte-Izquierdo¹, Salvador Carlos-Hernández²,
and Edgar Nelson Sanchez¹

¹ Department of Automatic Control, CINVESTAV del IPN, Unidad Gdl.,
Av. Científica 1145, Col. El Bajío, CP. 45015,
Zapopan, Jalisco, Mexico

{rbelmont, sanchez}@gdl.cinvestav.mx

² Department of Energy and Natural Resources, CINVESTAV del IPN, Unidad Saltillo,
Carr. Saltillo-Mty Km. 13, CP. 25900,
Ramos Arizpe, Coahuila, Mexico
salvador.carlos@cinvestav.edu.mx

Abstract. In this paper, a recurrent high order neural observer (RHONO) for anaerobic processes is proposed. The main objective is to estimate biomass and substrate in a completely stirred tank reactor. The recurrent high order neural network (RHONN) structure is based on hyperbolic tangent as activation function. The learning algorithm is based on an extended Kalman filter. The applicability of the proposed scheme is illustrated via simulation. Thus, this observer can be successfully implemented for control purposes.

Keywords: Recurrent high order neural observer, anaerobic digestion, extended Kalman filter.

1 Introduction

The rapid growth in the size of raw wastewater because of domestic, industrial and agricultural wastes requires careful consideration of all society sectors. One of the more encouraging methods for wastewater treatment is anaerobic digestion. The products of anaerobic digestion have value and can be used to offset treatment costs. However, this bioprocess is sensitive to variations on the operating conditions, such as pH, temperature, overloads, etc. In addition, some variables and parameters are hard to measure due to economical or technical constraints. Then, estimation and control strategies are required in order to guarantee adequate performance.

1.1 Brief Review of the State of the Art

In biological processes there exist hardly measurable or immeasurable variables which are necessary for process control [1]. Furthermore, the last two decades have seen an increasing interest to improve the operation of bioprocesses by applying advanced control schemes [2], [3], [4]. Hence, observer design is a prioritized

problem to be solved in addition to select adequate sensors. Last years, fuzzy algorithms have been used to design observers and controllers for bioprocesses [5], [6], [7], also Takagi Sugeno fuzzy observers [1] and nonlinear observers, specifically asymptotic and interval observers are dealt in [4].

Complete knowledge of the system model is usually assumed in order to design nonlinear state estimators; nonetheless this is not always possible. Moreover, in some cases especial nonlinear transformations are required; nevertheless those transformations are not often robust in presence of uncertainties. An interesting approach for avoiding the associated problem of model-based state observers is the neural network observer. Neural observers require feasible outputs and inputs measures and a training algorithm in order to learn the process dynamics; in this case, the model knowledge is not strictly necessary [8], [9], [10].

In this paper, a new neural observer is proposed in order to estimate mainly biomass and substrate in an anaerobic process for paper mills effluents treatment. This process is developed in a completely stirred tank reactor with biomass filter. The observer structure is based on hyperbolic tangent as activation function and is trained using an extended Kalman filter algorithm. The main advantage of this observer is a high performance and a low design and tune complexity.

2 Anaerobic Digestion Process

2.1 Process Description

Anaerobic digestion (AD) is a biological process in which organic matter (substrate) is degraded by micro-organisms (biomass), in absence of oxygen. Such degradation produces biogas (methane and carbon dioxide), and stable organic residues. The process is developed in four successive stages: hydrolysis, acidogenesis, acetogenesis and methanogenesis.

A functional diagram proposed in [11] is shown in Fig. 1. Biomass is classified as: X_1 , corresponding to hydrolytic, acidogenic and acetogenic bacteria and X_2 , corresponding to methanogenic bacteria. On the other hand, the organic load is classified in S_1 , the components *equivalent glucose*, which model complex molecules and S_2 , the components *equivalent acetic acid*, which represent the molecules directly transformed in acetic acid.

Thus, a mathematical model of the process is deduced from this functional diagram. On one side, the physical-chemical phenomena (acid-base equilibria and material conservation) are modeled by algebraic equations (1). On the other side, the biological phenomena are modeled by ordinary differential equations (2), which represent the dynamical part of the process. Finally the gaseous phase (CH_4 and CO_2) is considered as the process outputs (3). More details can be found in [12].

$$0 = g(x_a, x_d), \quad (1)$$

$$\dot{x}_d = f(x_a, x_d, u), \quad (2)$$

$$y = h(x_a, x_d), \quad (3)$$

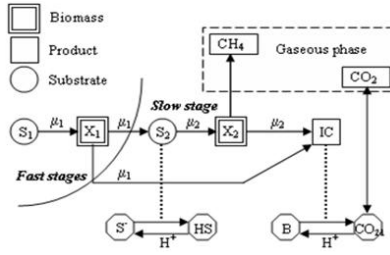


Fig. 1. Functional diagram of the anaerobic digestion

with:

$$\begin{aligned}
 x_a &= [HS \quad H^+ \quad S^- \quad CO_{2d} \quad B]^T, \\
 x_d &= [X_1 \quad S_1 \quad X_2 \quad S_2 \quad IC \quad Z]^T, \\
 u &= [S_{1in} \quad S_{2in} \quad IC_{in} \quad Z_{in} \quad D_{in}]^T,
 \end{aligned} \tag{4}$$

where HS is non ionized acetic acid (mol/l), H^+ ionized hydrogen (mol/l), S^- ionized acetic acid (mol/l), CO_{2d} dissolved carbon dioxide (mol/l), B measured bicarbonate (mol/l), IC inorganic carbon (mol/l), Z the total of cations (mol/l), S_{1in} the fast degradable substrate input (mol/l), S_{2in} the slow degradable substrate input (mol/l), IC_{in} the inorganic carbon input (mol/l), Z_{in} the input cations (mol/l) and D_{in} the dilution rate (h^{-1}).

2.2 Problem Statement

The last stage of the anaerobic process is considered the limiting stage on account of it is the slowest and the most important stage for process stability. Methanogenesis is very sensitive to variations on substrate concentration and the increase of biomass can be stopped by an excessive substrate production in the previous stages [1]. Depending on the amplitude and duration of these variations, the environment can be acidified so much so that biomass growth is inhibited, hence, the substrate degradation and the methane production can be blocked. Methane production, biomass growth and substrate degradation are good indicators of the biological activity inside the reactor. These variables can be used for monitoring the process and to design control strategies. Some commercial biogas sensors have been developed in order to measure methane production in bioprocesses [13]. However, substrate and biomass measures are more restrictive. The existing biomass sensors are quite expensive, are designed from biological viewpoint and then, they are not reliable for control purposes. Furthermore, substrate measure is done off-line by chemical analysis, which requires at least two hours. Then, state observers are an interesting alternative in order to deal with this situation.

3 Neural Networks

An artificial neural network (NN) is a massively parallel distributed processor, inspired from biological neural networks, which can store experimental knowledge and makes it available for use [14]. An artificial NN consists of a finite number of neurons (structural element), which are interconnected to each other.

One very useful NN structure is the recurrent NN. This recurrent structure has a profound impact on the learning capability of the NN and on its performance [14]. This structure also offers a better suited tool to model and control nonlinear systems [8]. Using neural networks, control algorithms can be developed to be robust to uncertainties and modeling errors. Recurrent high order neural networks (RHONN) are a generalization of the first-order Hopfield networks; they are proposed in [15].

3.1 Discrete-Time Recurrent High Order Neural Network

Let consider a MIMO nonlinear system:

$$x_i(k+1) = F(x(k), u(k)), \quad (5)$$

where $x \in \mathbb{R}^n$, represents state variables, $u \in \mathbb{R}^m$ is input vector, and $F \in \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is a nonlinear function.

Let consider also the following discrete-time RHONN:

$$\hat{x}_i(k+1) = w_i^T z_i(\hat{x}(k), u(k)), \quad i = 1, \dots, n, \quad (6)$$

where \hat{x}_i ($i = 1, 2, \dots, n$) is the state of the i -th neuron, n the state dimension; w_i is the respective on-line adapted weight vector, $u = [u_1, \dots, u_m]^T$ is the external input vector to the neural network and $z_i(x(k), u(k))$ is given by:

$$z_i = [z_{i_1} \quad z_{i_2} \quad \dots \quad z_{i_{L_i}}]^T = \left[\prod_{j \in I_1} y_{ij}^{d_{ij}(1)} \quad \prod_{j \in I_2} y_{ij}^{d_{ij}(2)} \quad \dots \quad \prod_{j \in I_{L_i}} y_{ij}^{d_{ij}(L_i)} \right]^T, \quad (7)$$

with L_i the respective number of higher-order connections, $\{I_1, I_2, \dots, I_{L_i}\}$ a collection of non-ordered subsets of $\{1, 2, \dots, n+m\}$, d_{ij} non-negative integers, and y_i defined as follows:

$$y_i = [y_{i_1} \quad \dots \quad y_{i_n} \quad y_{i_{n+1}} \quad \dots \quad y_{i_{n+m}}]^T = [S(\hat{x}_1) \quad \dots \quad S(\hat{x}_n) \quad u_1 \quad \dots \quad u_m]^T. \quad (8)$$

The nonlinear system (5) can be approximated by the following discrete-time RHONN parallel representation [9]:

$$\mathcal{X}_i(k+1) = w_i^{*T} z_i(\mathcal{X}(k), u(k)) + \epsilon_{z_i}, \quad i = 1, \dots, n, \quad (9)$$

where \mathcal{X}_i is the NN state vector, ϵ_{z_i} is a bounded approximation error, which can be reduced by increasing the number of adjustable weights [15]. Assume that there exists ideal weight vector w_i^* such that $\|\epsilon_{z_i}\|$ can be minimized on a compact set $\Omega_{z_i} \subset \mathbb{R}^{L_i}$.

The ideal vector w_i^* is an artificial quantity required for analytical purpose [15]. In general, it is assumed that this vector exists and is constant but unknown.

3.2 The Extended Kalman Filter as Training Algorithm

The Kalman filter (KF) is a set of mathematical equations which provides an efficient computational solution to estimates the state of a linear dynamic system with additive state and output white noises [16]. For KF-based neural network training, the network weights become the states to be estimated. The error between the neural network output and the measured plant output can be considered as additive white noise. If, however, the model is nonlinear, the use of KF can be extended through a linearization procedure, the resulting filter is the well-known extended Kalman filter (EKF). Since the neural network mapping is nonlinear, an EKF-type is required. The training goal is to find the optimal weight values, which minimize the predictions error. In this work, we use, an EKF-based training algorithm described by:

$$\begin{aligned} w_i(k+1) &= w_i(k) + \eta_i K_i(k) e_i(k) \\ K_i(k) &= P_i(k) H_i(k) M_i(k) \\ P_i(k+1) &= P_i(k) - K_i(k) H_i^T(k) P_i(k) + Q_i(k) \\ i &= 1, \dots, n, \end{aligned} \quad (10)$$

with:

$$\begin{aligned} M_i(k) &= [R_i(k) + H_i^T(k) P_i(k) H_i(k)]^{-1} \\ e_i(k) &= y(k) - \hat{y}(k), \end{aligned} \quad (11)$$

where $e_i(k) \in \mathbb{R}^p$ is the observation error, $P_i(k) \in \mathbb{R}^{L_i \times L_i}$ is the prediction error covariance matrix at step k , $w_i(k) \in \mathbb{R}^{L_i}$ is the weight (state) vector, L_i is the respective number of neural network weights, $y \in \mathbb{R}^p$, is the plant output, $\hat{y} \in \mathbb{R}^p$ is the NN output, η_i is the learning rate, $K_i(k) \in \mathbb{R}^{L_i \times p}$ is the Kalman gain matrix, $Q_i(k) \in \mathbb{R}^{L_i \times L_i}$ is the NN weight estimation noise covariance matrix, $R_i(k) \in \mathbb{R}^{p \times p}$ is the error noise covariance, and; $H_i(k) \in \mathbb{R}^{L_i \times p}$ is the matrix for which each entry (H_{ij}) is the derivative of the i -th neural output with respect to ij -th NN weight, (w_{ij}), given as follows:

$$H_{ij}(k) = \left[\frac{\partial \hat{y}(k)}{\partial w_{ij}(k)} \right]^T, \quad (12)$$

where $i=1, \dots, n$ and $j=1, \dots, L_i$. Usually P_i , Q_i and R_i are initialized as diagonal matrices, with entries $P_i(0)$, $Q_i(0)$ and $R_i(0)$ respectively. It is important to remark that $H_i(k)$, $K_i(k)$ and $P_i(k)$ for the EKF are bounded [16].

4 RHONO for Biomass and Substrate Estimation

4.1 Observer Design

First, system (1)-(3) is transformed to discrete time obtaining a form similar to:

$$\begin{aligned} x(k+1) &= F(x(k), u(k)) + d(k) \\ y(k) &= h(x(k)), \end{aligned} \quad (13)$$

where $x \in \mathbb{R}^n$ is the state vector of anaerobic digestion, $u \in \mathbb{R}^m$ is the input vector, $y \in \mathbb{R}^p$ is the output vector, $h(x(k))$ is a nonlinear function of the process states, $d(k) \in \mathbb{R}^n$ is a disturbance vector and $F(\bullet)$ is a smooth vector field and $F_i(\bullet)$ its entries; hence (13) can be rewritten as:

$$\begin{aligned} x(k) &= [x_1(k) \dots x_i(k) \dots x_n(k)]^T \\ d(k) &= [d_1(k) \dots d_i(k) \dots d_n(k)]^T \\ x_i(k+1) &= F_i(x(k), u(k)) + d_i(k), \quad i = 1, \dots, n \\ y(k) &= h(x(k)). \end{aligned} \quad (14)$$

After, a Luenberger neural observer (RHONO) is proposed for X_2 and S_2 with the following structure:

$$\begin{aligned} \hat{X}_2(k+1) &= w_{11}S(\hat{X}_2(k)) + w_{12}S^2(\hat{X}_2(k)) + w_{13}S(IC(k)) \\ &\quad + w_{14}S^2(\hat{X}_2(k))D_{in}(k) + g_1e(k), \\ \hat{S}_2(k+1) &= w_{21}S(\hat{S}_2(k)) + w_{22}S^2(\hat{S}_2(k)) + w_{23}S(IC(k)) \\ &\quad + w_{24}S^2(\hat{S}_2(k))D_{in}(k) + w_{25}S^2(\hat{S}_2(k))S_{2in}(k) + g_2e(k), \end{aligned} \quad (15)$$

with the output given as:

$$\hat{Y}_{CH_4} = R_1 R_2 \hat{\mu}_2 \hat{X}_2, \quad (16)$$

$$\hat{Y}_{CO_2} = \hat{\lambda} R_2 R_3 \hat{\mu}_2 \hat{X}_2. \quad (17)$$

As shown in (15), the proposed observer has a parallel configuration. Besides, the weight vectors are updated on-line with an EKF (10). The hyperbolic tangent:

$$S(x) = \alpha \tanh(\beta x), \quad (18)$$

with $\alpha, \beta > 0$ is used as activation function. This choice is done because the antisymmetric functions allow the NN to learn the respective dynamics in a faster way in comparison with other functions [17]. In addition, the hyperbolic tangent derivative is easily obtained. The RHONO structure is displayed on Fig. 2. For more details, the reader is referred to [18].

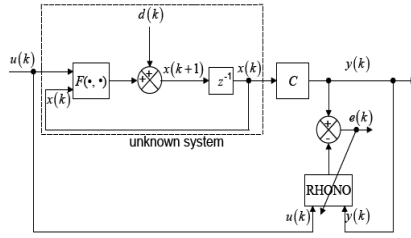


Fig. 2. Observer scheme

4.2 Tuning Guidelines

The covariance matrices are initialized as diagonals and verifying:

$$P_i(0) > R_i(0) > Q_i(0) . \tag{19}$$

This condition implies that *a priori* knowledge is not required to initialize the vector weights [19]. In fact, higher entries in $P_i(0)$ correspond to a higher uncertainty in the *a priori* knowledge. It is advisable to set $P_i(0)$ between 100-1000 and so on for the other covariance matrices observing (19). In this way, the covariance matrices for the Kalman filter are initialized as diagonals, with nonzero elements:

$$\begin{aligned} P_1(0) &= P_2(0) = P_3(0) = 1000, \\ R_1(0) &= R_2(0) = 10, R_3(0) = 1, \\ Q_1(0) &= Q_2(0) = 1, Q_3(0) = 0.1. \end{aligned} \tag{20}$$

An arbitrary scaling can be applied to $P_i(0)$, $R_i(0)$ and $Q_i(0)$ without altering the evolution of the weight vector. Since the NN outputs do not depend directly on the weight vector, the matrix H is initialized as $H(0) = 0$.

It is assumed that weights values are initialized to small random values drawn from a zero mean and normal distribution. The learning rate (η) determines the magnitude of the correction term applied to each neuron weight; it is usually bounded to $0 < \eta < 1$; moreover, η is far reaching on the convergence. Thus, if η is small then the transient estimated state is over-damped; if η is large then the transient estimated state is under-damped; finally if η is larger than a critical value then the estimated state is unstable. Therefore, it is better to set η to a small value and edge it upward if necessary. More details are discussed in [19].

The Luenberger-like observer gain (g) is set by trial and error; unfortunately there is a shortage of clear scientific rationale to define it. However, it is bounded to $0 < g < 0.1$ for a good performance on the basis of training experience.

5 Results and Discussion

The process model and the observer are implemented using Matlab/Simulink™. The observer is initialized at random values to verify the estimation convergence. In order

to test the observer sensitivity to change on inputs, a disturbance on the input substrate (30% S_{2in} increase) is incepted at $t = 200$ hours and eliminated at $t = 500$ hours. The on-line measurement of pH, Y_{CH_4}, Y_{CO_2} are supposed, as well as the system inputs. The scheme proposed in this paper is compared with that one described in [10] using the same operating conditions, as shown in Fig. 3. First, the main advantage of the proposed scheme in this paper is its simpler structure; also, the activation function is defined as a hyperbolic tangent, whereas in [10] is defined as a logistic function; hyperbolic tangent derivative is easier to obtain. Fig. 3 illustrates the improved performance of the scheme proposed in this paper and hence, its ease of use in control schemes and implementation in real time. Even though the observer is initialized at a random value, the convergence in both schemes is evident in the beginning of the simulation. Thus, it can be noticed that the neural observer is a good alternative to estimate those important states of the considered anaerobic process.

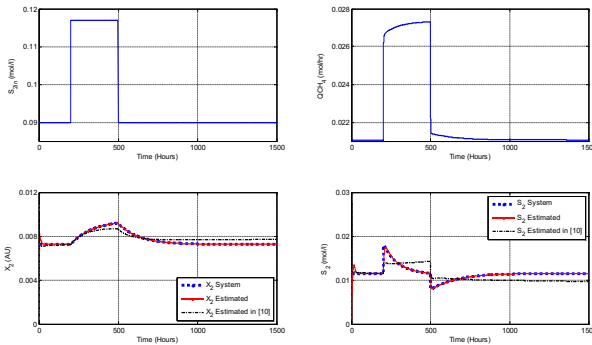


Fig. 3. Neural observer comparison considering a disturbance in S_{2in}

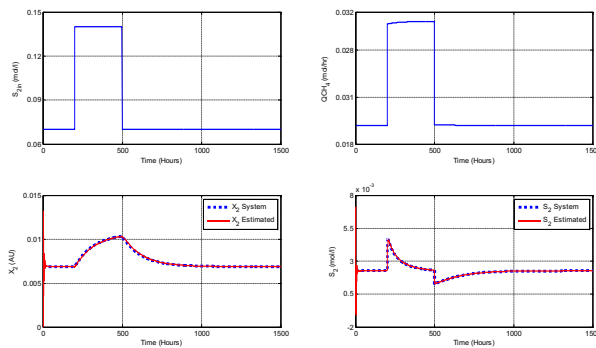


Fig. 4. Performance considering variation in μ_{1max}, μ_{2max} and a disturbance in S_{2in}

On the other hand, observer tolerance to change on system parameters is tested; the parameters variations take place on the biomasses growth rates. A 30% positive

variation in $\mu_{2\max}$, a 30% negative variation in $\mu_{1\max}$ and a disturbance of 100% in input S_{2in} are considered. The performance of the proposed RHONO is illustrated in Fig. 4, where it is clear that biomass and substrate are well estimated. Thus, the robustness of the proposed RHONO to parameters variations is verified.

6 Conclusions

A RHONO based on RHONN, which is trained with an EFK and structured by hyperbolic tangent functions, is proposed in this paper. The objective is to estimate the biomass concentration and substrate degradation in an anaerobic digestion process considering a completely stirred tank reactor with biomass filter, which is operated in continuous mode. The training of the RHONO is performed on-line. The variables are estimated from methane and dioxide flow rates, which are commonly measured in this process. Also, pH and system inputs measurements are assumed. Simulation results illustrate the effectiveness of model adaptation to system disturbance and robustness of the proposed RHONO.

Since one of the limiting factors for the implementation of the control strategies is the lack of on-line sensors, these neural observer outcomes are an interesting alternative to be applied. Thus research efforts are proceeding in order to implement the neural observer in real-time.

Acknowledgments. This work was supported by CONACyT project 57801Y.

References

1. Carlos-Hernández, S., et al.: Fuzzy Observer for the anaerobic digestion process. In: Proceedings of IFAC Symposium on Structures Systems and Control, Oaxaca, Mexico (2004)
2. Yamuna Rani, K., Ramachandra Rao, V.S.: Control of fermenters. *Bioprocess Engineering* 21, 77–78 (1999)
3. Van Lier, J.B., et al.: New perspectives in anaerobic digestion. *Water Science and Technology* 43(1), 1–18 (2001)
4. Alcaraz-González, V., González-Álvarez, V.: Robust Nonlinear Observers for Bioprocesses: Application to Wastewater Treatment. Book Chapter in *Selected Topics in Dynamics and Control of Chemical and Biological Processes*, pp. 119–164. Springer, Heidelberg (2007)
5. Giraldo-Gomez, E., Duque, M.: Automatic startup of a high rate anaerobic reactor using a fuzzy logic control system. Fifth Latin-American workshop-seminar in wastewater anaerobic treatment. Viña del Mar, Chile (1998)
6. Bernard, O., et al.: Advanced monitoring and control of anaerobic wastewater treatment plants: software sensors and controllers for an anaerobic digester. *Water Science and Technology* 43(7), 175–182 (2001)
7. Puñal, A., et al.: Advanced monitoring and control of anaerobic wastewater treatment plants: diagnosis and supervision by a fuzzy-based expert system. *Water Science and Technology* 43(7), 191–198 (2001)

8. Poznyak, A.S., et al.: *Differential Neural Networks for Robust Nonlinear Control*. World Scientific Publishing Co., Singapore (2001)
9. Ricalde, L.J., Sanchez, E.N.: Inverse optimal nonlinear high order recurrent neural observer. In: *International Joint Conference on Neural Networks*, Montreal, Canada (2005)
10. Urrego-Patarroyo, D., et al.: Recurrent Neural Networks Observer for Anaerobic Processes. In: *XIII Latin-American Congress of Automatic Control*, Mérida, Venezuela (2008)
11. Rozzi, A.: Modelling and control of anaerobic digestion process. *Transactions Instrumentation, Modelling and Control* 6(3), 153–159 (1984)
12. Carlos-Hernández, S.: Integrated intelligent control strategy for wastewater treatment plants by anaerobic digestion. Ph. D. Thesis, INPG, France (2005) (in French)
13. <http://www.bluesens.de> (visited on April 13, 2009)
14. Haykin, S.: *Neural Networks: A comprehensive Foundation*, 2nd edn. Prentice Hall, New Jersey (1999)
15. Rovithakis, G.A., Chistodoulou, M.A.: *Adaptive Control with Recurrent High-Order Neural Networks*. Springer, Berlin (2000)
16. Song, Y., Grizzle, J.W.: The extended Kalman filter as a local asymptotic observer for discrete-time nonlinear systems. *Journal of Mathematical Systems, Estimation and Control* 5(1), 59–78 (1995)
17. Sanchez, E.N., Alanís, A.Y.: *Neural Networks: Fundamental Concepts and Applications to Automatic Control*. Pearson Education, Madrid (2006) (in Spanish)
18. Sanchez, E.N., et al.: *Discrete Time High Order Neural Control Trained with Kalman Filtering*. Springer, Germany (2008)
19. Haykin, S.: *Kalman Filtering and Neural Networks*. John Wiley & Sons, New York (2001)

Prediction of Protein Subcellular Multi-localization by Using a Min-Max Modular Support Vector Machine

Yang Yang¹ and Bao-Liang Lu^{1,2,*}

¹ Department of Computer Science and Engineering, Shanghai Jiao Tong University

² Laboratory for Computational Biology, Shanghai Center for Systems Biomedicine,
800 Dong Chuan Road, Shanghai 200240, China

{alayman, bllu}@sjtu.edu.cn

Abstract. Prediction of protein subcellular location is an important issue in computational biology because it provides important clues for characterization of protein function. Currently, much effort has been dedicated to developing automatic prediction tools. However, most of them focus on mono-localational proteins. It should be noted that many proteins bear multi-localational characteristics, and they carry out crucial functions in biological processes. This work aims to develop a general pattern classifier for predicting multiple subcellular locations of proteins. We used an ensemble classifier, called min-max modular support vector machine (M^3 -SVM), to solve protein subcellular multi-localization problem, and proposed a task decomposition method based on gene ontology (GO) semantic information for the M^3 -SVM. We applied our method to a high-quality multi-localational protein data set. The M^3 -SVMs showed better performance than traditional SVMs using the same feature vectors. And the GO decomposition also helped improve the prediction accuracy with more stable performance than random decomposition.

1 Introduction

Identification of subcellular location is an important goal of protein bioinformatics. It can provide information helpful for understanding protein function, regulation and protein-protein interaction. And efficient computational tools can save costly and laborious wet-lab experiments. Therefore, prediction of protein subcellular localization has been an active research topic in bioinformatics in the last decade.

To develop automatic tools for subcellular localization, machine learning methods, such as neural networks [1], hidden Markov models (HMMs) [2] and support vector machines (SVMs) [3], have been widely used, thanks to the abundance of proteins with known locations in the public databases. The extracted features used in these classifiers fall into two types: sequence-based and annotation-based. Sequence-based methods use single-residue composition, dimer, trimer composition, or represent sequences as condensed feature vectors using pseudo-amino acid composition [4], signal-processing and

* To whom correspondence should be addressed. This work was partially supported by the National Natural Science Foundation of China (Grant No. 60773090 and Grant No. 90820018), the National Basic Research Program of China (Grant No. 2009CB320901), and the National High-Tech Research Program of China (Grant No. 2008AA02Z315).

text processing techniques [5]. In addition, N-terminal signals are very effective in identifying mitochondrial, chloroplast, and secretory pathway proteins. But sometimes the leading sequences of the test proteins are missing, and for many locations, no obvious sorting signal could be detected. As annotation becomes more abundant, many studies use annotation-based methods, including motifs, function domains, or gene ontology (GO) [6] to improve the prediction accuracy.

Most of these studies focus on mono-local proteins, i.e., they assume that proteins may exist in only one cellular compartment. This is not always the case. Many proteins are multi-local. They may translocate into different compartments, or secrete out of the cell. In most of the previous prediction systems, such proteins were discarded or treated like mono-local ones. Cai and Chou [7] dealt with such proteins in budding yeast by unfolding the multi-label data. For example, a tri-localized protein would be unfolded into three distinct samples with different labels, and predicted respectively. In essence, their method treats the multi-local proteins like single-local ones. Certain strategies and evaluation measures are needed to deal with the multi-local cases. In our previous studies [8,9], we collected the multi-local proteins from Swiss-Prot [10]. However, the annotations on subcellular localization are incomplete for many entries. According to our statistics, around 10% proteins in Swiss-Prot are annotated with more than one location. Recently, Zhang *et al.* [11] published a high-quality database of proteins with multiple subcellular locations, called DBMLoc. Given this database, the performance of predictors for multi-local proteins can be estimated more fairly.

Moreover, the subcellular localization prediction is usually an imbalanced classification problem. The numbers of proteins located in different compartments vary significantly, i.e., the class distribution is uneven. For example, proteins in cytoplasm, membrane and nucleus are much more numerous than those in other locations. A number of approaches have been proposed to address the class imbalance problem. Oversampling and undersampling are two typical methods [12]. Oversampling approach duplicates data from the minority class, and undersampling approach eliminates data from the majority class. Both methods aim to re-balance the classes. Obviously, oversampling increases the complexity of classification problem, while undersampling results in information loss. Although SVMs make the decision boundary based on support vectors rather than all data samples, it still can not work well in class imbalance problems because of the imbalanced support vector ratio and weakness of soft-margins [13,14].

In this paper, we used a min-max modular support vector machine (M^3 -SVM) to predict protein subcellular multi-localization. The classifier is an ensemble of support vector machines (SVMs) [15]. It is suited for imbalanced classification problems. It decomposes the original problem into relatively balanced subproblems to eliminate the skew of decision boundary. The subproblems are integrated by minimization and maximization principles in the ensemble classifier.

How to decompose the data set of a class for an M^3 -classifier has not been perfectly solved so far. The random decomposition is the most straightforward way, which divides the majority and minority classes randomly into nearly equal sizes. But it cannot ensure stable performance. In this paper, we propose a new decomposition method

based on biological domain knowledge, i.e., GO annotation. We noticed that for many pattern classification problems, the training data are organized by some prior knowledge, which could be useful clues for the modular methods. Here, we calculated the semantic similarity of GO terms, used the similarity to cluster proteins and partitioned training data for the proposed ensemble classifier.

In addition, to obtain good classification performance, we developed a new feature extraction method that combines multiple knowledge sources, including amino acid composition, secondary structure, and solvent accessibility. All of these features are believed to be closely correlated with protein subcellular localization.

The proposed method was evaluated on the data set, DBMLoc [11]. The M³-SVMs showed better performance than traditional SVMs using the same feature vectors. We also compared GO decomposition and random decomposition, and found that GO decomposition helped improve the prediction accuracy.

2 Methods

In this paper, we propose a modular classifier, min-max modular support vector machine (M³-SVM) [16,17], which is an ensemble of SVMs. Each SVM classifier is trained on a subset of the original data set. When a test sample comes, each trained SVM outputs a classification result. Then all of the outputs are integrated to get a final solution to the original problem according to two module combination rules, namely the minimization and the maximization principles.

For solving a large-scale and complex multi-label problem, our method consists of three main steps: a) decompose the original problem into two-class problems; b) further decompose the two-class problems which are difficult to be learned into a number of relatively smaller and balanced two-class subproblems. c) combine all the submodules into a hierarchical, parallel, and modular pattern classifier.

2.1 Classification of Multi-label Problems

The traditional method for solving multi-label task is to split the original problem into a set of binary classification tasks using one-versus-rest decomposition strategy. For a K -class multi-label problem, let \mathcal{T} denote its training set:

$$\mathcal{T} = \{(x_m, t_m)\}_{m=1}^L, t_m = \{t_m^k\}, k = 1, \dots, \tau_m, \tag{1}$$

where $x_m \in R^n$ is the m th sample in the data set, t_m is the label set of x_m , t_m^k is the k th label of x_m , τ_m denotes the total number of labels of x_m , and L is the total number of samples.

By decomposing a K -class multi-label problem \mathcal{T} into K mono-label two-class problems \mathcal{T}_i for $i = 1, \dots, K$, we have the training set of \mathcal{T}_i as follows:

$$\mathcal{T}_i = \{(x_m^{i+}, +1)\}_{m=1}^{L_i^+} \cup \{(x_m^{i-}, -1)\}_{m=1}^{L_i^-}, \tag{2}$$

where L_i^+ is the number of positive samples of the two-class problem \mathcal{T}_i , and L_i^- is the number of negative samples. For \mathcal{T}_i , positive samples are the samples whose label sets

contain label i , and negative samples are the remaining ones. Thus x_m will appear τ_m times as positive training data, and $(K - \tau_m)$ times as negative training data.

Each binary classifier decides whether or not a novel sample belongs to a particular class. Obviously, each of the binary problems has the same size as the original problem. The data distribution would become more imbalanced because of the one-versus-rest strategy. For a complex and imbalanced binary classification problem, we can further divide it into a number of relatively small and balanced two-class subproblems. Each subproblem is solved by a SVM, and all the subproblems are combined using *MIN* and *MAX* principles. The functions of *MIN* and *MAX* are to find the minimum and maximum values of all the inputs, respectively.

For a two-class problem \mathcal{T}_i , its positive and negative training sets, \mathcal{T}_i^+ and \mathcal{T}_i^- , are further decomposed into N_i^+ and N_i^- subsets, where $1 \leq N_i^+ \leq L_i^+$ and $1 \leq N_i^- \leq L_i^-$.

$$\mathcal{T}_i^{+j} = \{(x_m^+, +1)\}_{m=1}^{L_i^{+j}}, j = 1, \dots, N_i^+, \tag{3}$$

$$\mathcal{T}_i^{-j} = \{(x_m^-, -1)\}_{m=1}^{L_i^{-j}}, j = 1, \dots, N_i^-, \tag{4}$$

where L_i^{+j} and L_i^{-j} are the numbers of samples in \mathcal{T}_i^{+j} and \mathcal{T}_i^{-j} , respectively. Each two-class problem \mathcal{T}_i is solved by an M^3 network shown in Fig. 1.

According to the *MIN* and *MAX* principles [16], N_i^+ *MIN* units and one *MAX* unit are required to combine all the $(N_i^+ \times N_i^-)$ modules. Each of the *MIN* units combines N_i^- modules. The final output is determined by the outputs of all modules.

This ensemble classifier has some advantages over other methods in dealing with imbalanced problems. Compared with under-sampling method, it makes full use of the training data without information loss. Compared with over-sampling methods, it does

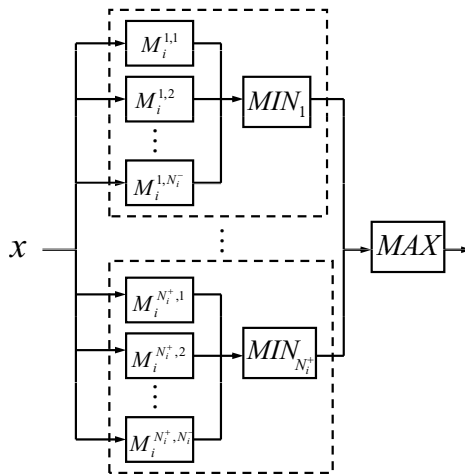


Fig. 1. Structure of M^3 network for a two-class problem \mathcal{T}_i , which is divided into $(N_i^+ \times N_i^-)$ two-class subproblems

not add to learning cost in each module, and speeds up the training process. The *MIN* and *MAX* rules provide an effective ensemble principle to obtain a solution to the original problem from the sub-problems, and they are easy to implement.

2.2 Task Decomposition

According to the one-versus-rest strategy, a K -class classification problem is decomposed into K two-class problems. Some of the two-class problems may have extremely imbalanced distribution of the positive and negative classes. Moreover, some of the two-class problems may be too large for fast learning. The most important advantage of the M^3 model is that it can further divide the large and imbalanced two-class problems into relatively smaller and more balanced subproblems.

Random partition is the simplest and most straightforward way. Given a specific module size, when we choose samples randomly from the training set to build a module, the samples may have no distribution relationship with each other. In such cases, although the subproblem has a reduced data size, it may be still hard to solve, and have complex decision boundary apt to overfit. Since the overall classification capability lies on the performance of all the modules, the poor boundaries learned by some modules would degrade prediction accuracy of the whole system. Therefore, the random partition can not obtain a stable performance.

Several decomposition strategies have been developed for M^3 model, such as hyperplane decomposition [17] and equal clustering [18]. Hyperplane decomposition uses a group of parallel hyperplanes to partition data into subsets. This method is fast and suitable for sparse data. Equal clustering (EC) works similarly to K -means clustering. The only difference is that EC pays more attention to load balance for the seek of parallel learning, so the clusters are kept in nearly equal size. All these methods aim to utilize the geometric distribution characteristics of data points in the high-dimension space.

In the past [9,18], we either divided the data randomly or based on the distance of sample points in the feature space, like hyperplane decomposition and equal clustering, but ignored the prior knowledge which may contribute useful information to do clustering within a big class. Here, we want to fully utilize the Gene Ontology information and achieve a better partition, such that the proteins sharing some common attributes could be grouped together. In the GO graph, GO terms are structured hierarchically and have semantic relations ('is-a' and 'part-of') with each other. A child node is more specialized than its parental nodes, and more than one parental node may exist. Here, we used similarity measure of GO terms based on their semantic relations to cluster proteins in a class which needs to be decomposed. Many methods have been developed to define the similarity between two GO terms. Given the similarity between two GO terms, the similarity between two sets of GO terms can be calculated. Suppose each protein corresponds to a set of GO terms, the similarity between two proteins can be obtained accordingly.

The data set used in our experiments was annotated with GO terms, including cellular component, biological process and molecular function. In this work, we adopted the method proposed by Wang *et al.* [19] to measure the semantic similarity of GO terms. We built the GO similarity matrix for our training data set, and used the clustering tool, CLUTO [20] to partition the data based on the similarity matrix. The program "scluster"

in the tool kit was used. The number of clusters should be specified for this program. We calculated the number of clusters according to the predefined module size. Suppose that Class C_i has m samples and C_i is divided into k modules when n is the predefined module size, then m/k must be the closest value to n among all possible module sizes of C_i . Random decomposition makes each module equal size, while in GO decomposition, the actual size of each module is determined by the clustering method.

2.3 Feature Extraction

The features include amino acid composition, secondary structure and solvent accessibility information [21]. The former 60 dimensions are the amino acid composition of the full sequence on three secondary structure elements, i.e., strand (E), helix (H) and coil (C). The value of each dimension is calculated by

$$f_i^k = \frac{N_i^k}{L}, \quad (5)$$

where $k=\{H, E, C\}$, N_i^k is the frequency of amino acid i in secondary structure element k , and L is the length of the sequence. The latter 40 dimensions are the amino acid composition on two solvent accessibility status, namely buried (B) and exposed (E), and is calculated similarly as Eq. 5, with $k = \{B, E\}$.

The secondary structure elements were predicted by PSIPRED [22], and solvent accessibility status were predicted by ACCpro [23]. Both of them are highly accurate prediction methods. All the feature vectors were scaled in the range of $[0, 1]$ using SVM-Scale in the LibSVM package [24].

3 Results and Discussion

In order to test the performance of our methods, we applied them to a high-quality multi-locational protein database, DBMLoc, published by Zhang *et al.* [11], which was collected from multiple databases and experimentally determined localization data. All the cellular compartments were assigned into twelve categories as shown in Table 1. Some subcellular localization annotations which can not be classified into the twelve categories are assigned to ‘others’. As a result, the number of classes used in our prediction system is 13. All of the proteins in DBMLoc database have no less than two locations. The average number of labels for each protein is 2.2.

To avoid overfitting, we used the non-redundant data with sequence similarity below 25%. The training and test data sets are mutually exclusive. Test data is a high quality set including 631 proteins. Training data has a total of 2344 proteins, extracted from the complete non-redundant set (25%) by removing the overlapping data with test set. The statistics of the data sets are shown in Table 1. From the table, we can see that the data distributions are very imbalanced on different cellular compartments. Proteins of membrane, cytoplasm, and nucleus make an overwhelming portion, which adds to classification difficulty.

Although this data set is fully annotated, we did not use gene ontology as features to build our predictor based on the consideration that many test proteins are novel and do

Table 1. Training and test data distribution of DBMLoc

Location	Training	Test
Others	134	36
Extracellular	471	43
Ribosome	58	15
Virion	31	2
Membrane	1240	283
Cytoplasm	1172	417
Mitochondrion	445	123
Nucleus	844	344
Plastid	132	8
Vacuole	16	4
Cell wall	21	5
ER	322	53
Golgi	162	45
Total label #	5048	1378
Total protein #	2344	631

not have GO information. But the prior knowledge about training data could be fully utilized. Therefore, we used GO semantic similarity to guide the module decomposition of training data. We experimented three methods with the same feature vectors. One is min-max modular support vector machine with gene ontology decomposition (M³-SVM(GO)). The second is min-max modular support vector machine with random decomposition (M³-SVM(R)). The last one is traditional SVM. The module sizes of 100, 400 and 800 were tested for both M³-SVM(GO) and M³-SVM(R). And the results of M³-SVM(R) were averaged through five repetition experiments.

Here we chose LibSVM version 2.8 [24] as the base classifier for the ensemble classifier. We experimented with polynomial, sigmoid and RBF kernels and observed that RBF kernel has the best classification accuracy. We performed ten-fold cross-validation and grid search on training data to find the optimum parameters for SVMs. The experimental results reported in the following were obtained with the kernel parameters $\gamma = 2^{-6}$ and $C = 2^4$. All experiments were conducted on a Pentium 4 double CPU (2.8GHz) PC with 2GB RAM.

Multiple measures were used to assess the performance of our proposed method, including precision (P), recall (R), F_1 , total accuracy (TA), location accuracy (LA) and average F_1 ($aveF_1$). The former three measures, P , R and F_1 , were used to measure the prediction quality of each location, and the last three measures, TA , LA and $aveF_1$, were used to measure the overall prediction quality across all locations.

Table 2 shows overall performance (TA , LA and $aveF_1$) of the three methods, traditional SVM, M³-SVM(R) and M³-SVM(GO). Three different module sizes for M³-SVMs are compared. Column 2 shows the predefined module sizes. Column 3 shows the numbers of subproblems.

From this table, several observations could be made. First, both M³-SVMs with random decomposition and with GO decomposition have higher TA and LA than traditional SVM. The M³-SVMs improve not only average location accuracy but also total

Table 2. Overall accuracy of three methods on DBMLoc

Method	Module size	Module #	TA (%)	LA (%)	$aveF_1$ (%)
SVM	2344	13	64.7	41.4	42.0
M ³ -SVM	100	848	65.1	48.0	40.7
(Random)	400	83	66.9	47.7	42.7
	800	38	66.2	45.2	43.4
M ³ -SVM	100	848	66.2	48.5	42.3
(GO)	400	83	67.1	45.2	42.6
	800	38	66.3	43.4	43.6

accuracy, which indicates that they do not sacrifice majority classes for the classification of minority classes. Second, M³-SVMs have higher $aveF_1$ than traditional SVM except M³-SVM(R) with module size 100. And M³-SVM(GO) with module size 800 achieved the highest $aveF_1$. As the module size goes down, LA increases, but $aveF_1$ decreases, which suggests a higher false positive rate of M³-SVMs when the module size becomes smaller. There is a tradeoff between location accuracy and false positive rate. Finally, M³-SVM(R) has lower TA and $aveF_1$ than M³-SVM(GO) but higher LA , suggesting that it gives more preference to the minority classes. And its performance is relatively deteriorated than M³-SVM(GO)'s when the module size is very small (100). Since random decomposition randomly divides each training class into equal size modules, while GO decomposition is based on the relationship between proteins according to GO, the latter method has a more stable performance.

Tables 3 and 4 list detailed recall and F_1 of traditional SVM and M³-SVMs(GO) on each location. Obviously, the modulization helps improve recall a lot especially for the

Table 3. Recall comparison. a: SVM, b: M³-SVM(GO) with module size 100, c: M³-SVM(GO) with module size 400, d: M³-SVM(GO) with module size 800. 1-13 correspond to the 13 subcellular locations listed in Table 1

Method	Recall (%)												
	1	2	3	4	5	6	7	8	9	10	11	12	13
a	0.0	48.8	20.0	100.0	63.3	84.7	52.8	69.2	25.0	0.0	20.0	41.5	13.3
b	5.6	62.8	20.0	100.0	61.8	82.5	57.7	73.3	50.0	0.0	60.0	43.4	13.3
c	2.8	53.5	20.0	100.0	63.3	83.5	59.3	75.0	25.0	0.0	40.0	45.3	20.0
d	2.8	55.8	26.7	100.0	66.4	83.9	57.7	71.2	12.5	0.0	40.0	32.1	15.6

Table 4. F_1 comparison. a, b, c and d are the same as in Table 3

Method	F_1 (%)												
	1	2	3	4	5	6	7	8	9	10	11	12	13
a	0.0	44.7	31.6	100.0	67.5	81.1	52.2	72.7	16.7	0.0	22.2	38.9	18.5
b	7.0	42.9	24.0	80.0	66.7	81.2	50.4	76.0	17.8	0.0	50.0	34.3	20.0
c	4.5	44.2	27.3	80.0	67.4	80.7	51.0	76.6	12.5	0.0	44.4	40.0	25.4
d	4.8	46.2	38.1	100.0	69.8	80.6	53.0	74.0	8.0	0.0	40.0	30.9	20.9

minority classes, while the two biggest classes, membrane and Cytoplasm are recognized best by M³-SVMs(GO) with module size 800 and traditional SVM, respectively.

As the F_1 values listed in Table 4 show, M³-SVMs(GO) with module size 800 has the best performance. It wins on 5 locations, extracellular, ribosome, virion, membrane, and mitochondrion, among the four methods, and has higher F_1 than traditional SVM on 8 locations.

In this experiment, M³-SVMs(GO) with module size 800 performs the best considering all the measures. A too small module size would result in numerous modules which increase computation cost and deteriorate the performance. However, we could specify different module sizes for different binary classification problems according to the ratio of training samples of the positive and negative classes.

In addition, we notice that all the classifiers failed to recognize vacuole protein from other proteins. One reason is that it has the least training samples (11 proteins) and only 4 test samples, thus the ratio of positive and negative data is too small to classify the positive data correctly. The other reason would be the current feature vectors do not contain features that are informative enough to discriminate vacuole proteins from others.

Response time should also be considered as an important factor when measuring the performance of a classifier. Table 5 exhibits a comparison of response time between traditional SVM and M³-SVMs of different module sizes. We reported two categories of run times. ‘Time1’ is the response time (including training and test time) of the classifier running all subproblems in series. ‘Time2’ is the training time for a single subproblem which costs the longest time. In parallel learning, ‘Time2’ is more important. For traditional SVM, a subproblem means a two-class problem.

Table 5. Response time comparison

Method	<i>Time1</i> (sec.)	<i>Time2</i> (sec.)
SVM	23.4	3.3
M ³ -SVM (100)	22.2	<0.1
M ³ -SVM (400)	19.2	0.4
M ³ -SVM (800)	19.7	1.3

M³-SVMs obtained shorter response time than traditional SVM even in sequential running. Regarding ‘Time1’, M³-SVM with the module size 400 is the most efficient for DBMLoc, because it achieves a tradeoff between the number and size of modules. For large-scale data, we can train modules in parallel, and choose a modules size as small as necessary.

4 Conclusion

This paper introduces an ensemble classifier for protein subcellular multi-localization. The classifier has several advantages in solving large-scale, class imbalance, multi-label problems. On the one hand, parallel and distributed training can be easily implemented because of its modularity. On the other hand, it has a balanced performance on all classes because various task decomposition strategies can be used.

Taking into account the GO information, we partitioned large classes into relatively smaller modules according to GO semantic similarity matrix. The experimental results show the effectiveness of the proposed GO decomposition method, and demonstrate that the M^3 -SVM is very competent in solving such complex problems with class imbalance and multi-label characteristics.

References

1. Reinhardt, A., Hubbard, T.: Using neural networks for prediction of the subcellular location of proteins. *Nucleic Acids Research* 26(9), 2230–2236 (1998)
2. Fujiwara, Y., Asogawa, M., Nakai, K.: Prediction of Mitochondrial Targeting Signals Using Hidden Markov Model. *Genome Inform. Ser. Workshop Genome Inform.* 8, 53–60 (1997)
3. Hua, S., Sun, Z.: Support vector machine approach for protein subcellular localization prediction. *Bioinformatics* 17(8), 721–728 (2001)
4. Chou, K.: Prediction of protein cellular attributes using pseudo-amino acid composition. *Proteins Structure Function and Genetics* 44(1), 60–60 (2001)
5. Yang, W.Y., Lu, B.L., Yang, Y.: A Comparative Study on Feature Extraction from Protein Sequences for Subcellular Localization Prediction. In: *Proc. of the 2006 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pp. 201–208 (2006)
6. Ashburner, M., Ball, C., Blake, J., Botstein, D., Butler, H., Cherry, J., Davis, A., Dolinski, K., Dwight, S., Eppig, J., et al.: Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.* 25(1), 25–29 (2000)
7. Cai, Y.D., Chou, K.C.: Predicting 22 protein localizations in budding yeast. *Biochem. Biophys. Res. Commun.* 323(2), 425–428 (2004)
8. Yang, Y., Lu, B.-L.: Prediction of protein subcellular multi-locations with a min-max modular support vector machine. In: Wang, J., Yi, Z., Žurada, J.M., Lu, B.-L., Yin, H. (eds.) *ISNN 2006*. LNCS, vol. 3973, pp. 667–673. Springer, Heidelberg (2006)
9. Chen, K., Lu, B.L., Kwok, J.T.: Efficient Classification of Multi-label and Imbalanced Data using Min-Max Modular Classifiers. In: *International Joint Conference on Neural Networks*, pp. 1770–1775 (2006)
10. Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M., Estreicher, A., Gasteiger, E., Martin, M., Michoud, K., O'Donovan, C., Phan, I., et al.: The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Research* 31(1), 365–370 (2003)
11. Zhang, S., Xia, X., Shen, J., Zhou, Y., Sun, Z.: DBMLoc: a Database of proteins with multiple subcellular localizations. *BMC Bioinformatics* 9(1), 127 (2008)
12. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* 16, 321–357 (2002)
13. Wu, G., Chang, E.: Class-boundary alignment for imbalanced dataset learning. In: *Proceedings of the ICML*, vol. 3
14. Akbani, R., Kwek, S., Japkowicz, N.: Applying Support Vector Machines to Imbalanced Datasets. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) *ECML 2004*. LNCS, vol. 3201, pp. 39–50. Springer, Heidelberg (2004)
15. Vapnik, V.: *Statistical learning theory*. Wiley, Chichester (1998)
16. Lu, B.L., Ito, M.: Task decomposition and module combination based on class relations: a modular neural network for pattern classification. *IEEE Transactions on Neural Networks* 10(5), 1244–1256 (1999)
17. Lu, B.L., Wang, K., Utiyama, M., Isahara, H.: A part-versus-part method for massively parallel training of support vector machines. In: *Proceedings. IEEE International Joint Conference on Neural Networks*, vol. 1, pp. 735–740 (2004)

18. Wen, Y.M., Lu, B.L., Zhao, H.: Equal clustering makes min-max modular support vector machine more efficient. In: Proceedings of the 12th International Conference on Neural Information Processing, pp. 77–82 (2006)
19. Wang, J., Du, Z., Payattakool, R., Yu, P., Chen, C.: A new method to measure the semantic similarity of GO terms. *Bioinformatics* 23(10), 1274 (2007)
20. Karypis, G.: CLUTO-A Clustering Toolkit (2002)
21. Shamim, M., Anwaruddin, M., Nagarajaram, H.: Support Vector Machine-based classification of protein folds using the structural properties of amino acid residues and amino acid residue pairs. *Bioinformatics* 23(24), 3320 (2007)
22. McGuffin, L., Bryson, K., Jones, D.: The PSIPRED protein structure prediction server. *Bioinformatics* 16(4), 404–405 (2000)
23. Cheng, J., Randall, A., Sweredoski, M., Baldi, P.: SCRATCH: a protein structure and structural feature prediction server. *Nucleic Acids Research* 33, W72–W76 (2005)
24. Chang, C., Lin, C.: LIBSVM: a library for support vector machines. Software (2001), <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

Application of MultiLayer Perceptron Type Neural Network to Camera Calibration

Dong-Min Woo and Dong-Chul Park

Information Engineering Department, Myongji University,
Yongin, Gyeonggido, Korea 449-728
{dmwoo, parkd}@mju.ac.kr

Abstract. The objective of camera calibration is to obtain the correlation between camera image coordinate and 3D real world coordinate. In this paper, we propose a new approach which is based on the neural network model instead of the physical camera model including position, orientation, focal length, and optical center. The neural network employed in this paper is MLPNN (MultiLayer Perceptron Type Neural Network), which is primarily used as a mapper between 2D image points and points of a certain space in 3D real world. The neural network model implicitly contains all the physical parameters, some of which are very difficult to be estimated in the conventional calibration methods. In order to show the performance of the proposed method, images from two different cameras with three different camera angles were used for calibrating the cameras. The performance of the proposed neural network approach is compared with the well-known Tsai's two stage method in terms of calibration errors. The results show that the proposed approach gives much more consistent and acceptable calibration error over Tsai's two stage method regardless of the quality of camera and the camera angles.

Keywords: camera calibration, 2D image, neural network, 3D real world, camera model.

1 Introduction

Camera calibration is a procedure to determine the the correlation between camera image coordinate and 3D real world coordinate. In the area of computer vision, many applications require camera calibration procedure for estimating an accurate 3D coordinate from 2D image point. Most calibration methods are based on the camera model which consists of physical parameters of the camera including position, orientation, focal length, and optical center. In this paper, we propose a new approach which is based on the neural network model instead of the physical camera model.

The conventional calibration methods have been mostly studied on the estimation of camera physical parameters including position, orientation, focal length, and optical center [1][2][3]. This kind of the traditional method explicitly evaluates the physical parameters, and referred to be an explicit calibration method.

However the main objective of camera calibration is to obtain the correlation between camera image coordinate and 3D real world coordinate. In this context we can use an implicit calibration method, where the nonlinear mapping model functions as an implicit model which can give us a transformation between 2D image points and points of a certain space in 3D real world. This model is called as an implicit model, and the calibration method using this model is an implicit calibration method [4,5].

In this paper, we propose an implicit calibration method by using an MLPNN (MultiLayer Perceptron Type Neural Network). MLPNN has been shown to have the ability to function as a nonlinear mapper between input patterns and output patterns. This nonlinear mapping ability can be utilized to address some physical parameters in implicit camera calibration that cannot be readily estimated by the existing calibration methods. The MLPNN-based camera calibration approach does not estimate camera physical parameters. However, this is not an issue when the objective of the camera calibration process is to obtain the correlation between the camera image coordinates and the 3D real world coordinates. The implicit camera calibration approach, which can calibrate a camera without explicitly computing its physical parameters, can be used for both the 3D measurement and the generation of image coordinates.

2 Conventional Calibration Method

Most conventional calibration methods are based on the explicit estimation of camera parameters of the camera model initially established. Tsai's two stage method [3] is one of the widely used explicit calibration methods. This paper choose Tsai's two stage method as a reference state of art method for the purpose of performance comparison. Tsai's two stage method first obtains the transformation parameters with the assumption that there exists no distortion in the camera. Tsai's two stage method then refines the transformation parameters with the distortion of the camera by using a nonlinear search. That is, first, the camera model is assumed to be ideal for the camera calibration by neglecting the lens distortion.

Fig. 1 shows the camera model used in Tsai's two stage method. A point P is an object of the real world coordinate (X_w, Y_w, Z_w) and (x, y, z) is a 3D camera coordinate. The center of the camera coordinate is the optical center O and (X, Y) is the image coordinate with the center of O_i . The distance between O and O_i is f , the focal length of the camera. (X_u, Y_u) is the corresponding point with the assumption of no lens distortion. (X_u, Y_u) is then translated to (X_f, Y_f) , which is a point in computer image coordinate on the image buffer and is expressed in pixel numbers. The basic geometry of the camera model can be written as the transformation of the two coordinates with the following displacement and orientation:

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \quad (1)$$

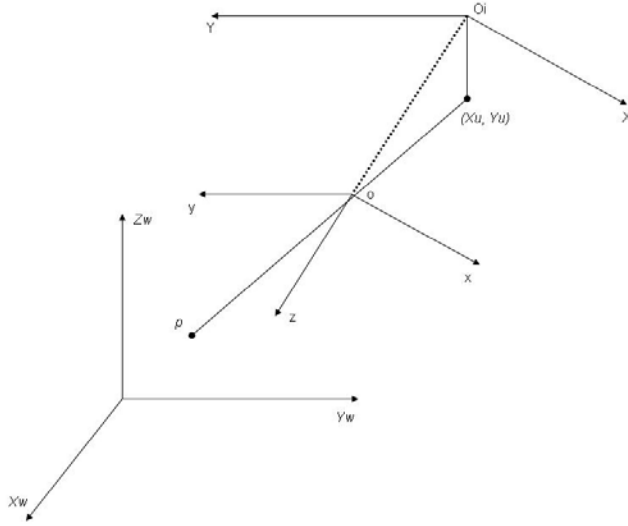


Fig. 1. Camera model defined in Tsai's two stage method

with

$$\begin{aligned}
 r_1 &= \cos \psi \cos \theta \\
 r_2 &= \sin \psi \cos \theta \\
 r_3 &= -\sin \theta \\
 r_4 &= -\sin \psi \cos \theta + \cos \psi \sin \theta \cos \phi \\
 r_5 &= \cos \psi \cos \theta + \sin \psi \sin \theta \sin \phi \\
 r_6 &= \cos \theta \sin \phi \\
 r_7 &= \sin \psi \sin \phi + \cos \psi \sin \theta \cos \phi \\
 r_8 &= -\cos \psi \sin \phi + \sin \psi \sin \theta \cos \phi \\
 r_9 &= \cos \theta \cos \phi
 \end{aligned}$$

where θ , ϕ and ψ represent yaw, pitch and roll, respectively.

As can be seen from the above equations, there are six extrinsic parameters: θ , ϕ , and ψ for rotation, and three components for the translation vector T . The problem of camera calibration is to find the six parameters θ , ϕ , ψ , T_x , T_y , and T_z by using the number of points measured in the (X_w, Y_w, Z_w) coordinate.

In the second stage of Tsai's two stage method, a distortion parameter is considered. The relations between the computer image coordinate with distortion and the real world coordinate can be derived as follows:

$$\begin{aligned}
 S_x(X_f - C_x)(1 + G(X_d^2 + Y_d^2)) \\
 = f \left(\frac{r_1 x_w + r_2 y_w + r_3 z_x + T_x}{r_7 x_w + r_8 y_w + r_9 z_w + T_x} \right) \quad (2)
 \end{aligned}$$

$$\begin{aligned}
 S_x(X_f - C_y)(1 + G(X_d^2 + Y_d^2)) \\
 = f \left(\frac{r_4 x_w + r_5 y_w + r_6 z_x + T_x}{r_7 x_w + r_8 y_w + r_9 z_w + T_x} \right) \quad (3)
 \end{aligned}$$

where (X_f, Y_f) is the image coordinate of the frame grabber, (C_x, C_y) is the image center, S_x and S_y are components of the translating scale of the x-axis and y-axis when the A/D transform is performed, (X_d, Y_d) is a distorted coordinate by lens distortion, and G is the distortion parameter. Tsai's two stage method obtained the solution by using a gradient-based nonlinear search method. In an explicit calibration, the calibration is performed with extrinsic parameters. However, the distortion parameters cannot include all the parameters involved in the distortion of the image. Even with the assumption of perfect inclusion of distortion parameters, there still remains room for errors in finding the right solution for such parameters.

3 Implicit Camera Calibration

3.1 Calibration Method Using ANN

Suppose that there is a calibration plane and the center of the calibration plane is defined as O . In the calibration plane, we have N points. A point, $P:(X_i, Y_i) \in w_i$, $i = 1, 2, \dots, N$, in the world plane is ideally projected to $\bar{p} : (\bar{x}_i, \bar{y}_i)$ in the camera CCD plane. However, because of the distance of the camera lens, the point of the world plane is projected to a distorted point, $p:(x_i, y_i)$. This point is observed through the frame buffer coordinate $p(u_i, v_i)$ in pixels.

For a back-projection problem, a transformation from the image coordinates in the frame buffer to the world coordinates in the calibration plane is required. For this purpose, an ANN is adopted in the proposed ANN-based calibration approach, where the input and the output of the ANN are the image coordinates and the world coordinates, respectively. After proper training of the ANN with training points, the ANN can map the relation of two planes. Owing to the nonlinear system modeling capability of the ANN, it is not necessary to utilize all the physical parameters involved with the camera calibration, including the lens distortion and the focal length of the camera.

With the coordinate system shown in Fig. 2, (x_1, y_1, z_1) and (x_2, y_2, z_1) are defined as two points on the calibration plane $Z = z_1$, and (x'_1, y'_1, z_2) and (x'_2, y'_2, z_2) are two other points on the plane $Z = z_2$. The line equations that pass each of the two points can be expressed by the following equations:

$$\vec{P} = (x_1, y_1, z_1) + t(x'_1 - x_1, y'_1 - y_1, z_2 - z_1) \tag{4}$$

$$\vec{Q} = (x_2, y_2, z_2) + t(x'_2 - x_2, y'_2 - y_2, z_2 - z_1) \tag{5}$$

$$\vec{P} = \vec{Q} \tag{6}$$

Since the equations given by Eq.(1) and Eq.(2) meet at the point C, i.e., Eq.(3), this point can be considered as the perspective center of the image, as shown in Fig. 2.

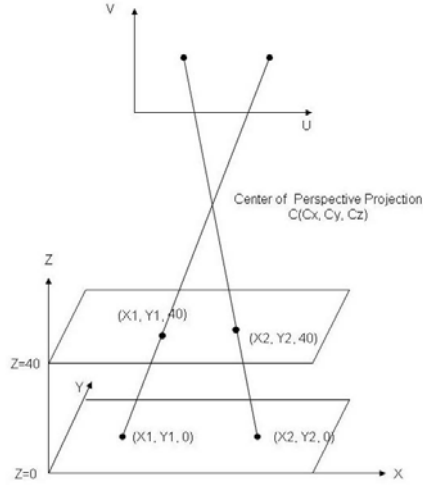


Fig. 2. The center of a perspective projection

By using the perspective center of an image, the estimation of the image coordinates of any 3D world point P can be obtained. In this case, an ANN that is trained with the real world coordinates of points on $Z = z_1$ as inputs and the image plane coordinates for the corresponding points as targets is given. It should be noted that the input and target for the ANN in this case are different from those of the back-projection problem. When the image coordinate of a point (P_1) on any calibration plane Z is needed, the line equation that passes the point (P_1) in the calibration plane Z and the perspective center of a camera (C) is first obtained. The line equation can produce P_0 on the calibration plane $Z = z_1$. By using P_0 as the input to the trained ANN, we can obtain the image coordinates of the point \hat{p} . This process is shown in Fig. 2.

3.2 MLPNN Structure for Camera Calibration

The neural network model adopted in this paper is a standard MLPNN and an error back-propagation algorithm is used for training the MLPNN. After several experiments, the architecture of the MLPNN is selected as $2 \times 10 \times 8 \times 2$, as shown in Fig. 3. Note that the selection of a specific architecture is a state of art and other architectures can be also used without any degradation of the resulting performance. With the architecture chosen, no overfitting problem was experienced with 5,000 training epochs. Note that proper numbers of training epochs are dependent on the complexity of the given problem and the number of training data. Note that the neurons in the input and output layers represent the 2D coordinates. More detailed information on the MLPNN and error back-propagation algorithm can be found in [6].

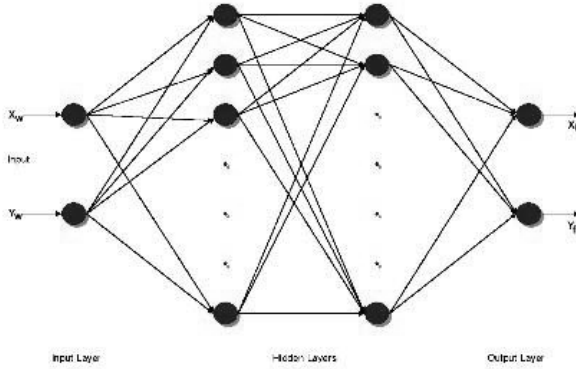


Fig. 3. ANN structure for camera calibration

Unlike the explicit camera calibration method, the proposed MLPNN-based method finds the direct relation between the world coordinates and the image coordinates. The MLPNN adopted in this implicit calibration approach can incorporate all the extrinsic parameters of the camera and the distortion parameters when the MLPNN is trained properly.

4 Experimental Results

To verify the effectiveness of the proposed camera calibration method, we use two types of cameras: the professional machine vision camera and the low quality CCTV camera for the monitoring purpose. The specifications of two cameras for our experimental environment are summarized in Table 1. In Table 1, Camera I is a professional machine vision camera, while Camera II is a low quality CCTV camera.

Table 1. Specifications of image acquisition devices used in the experiment

Frame grabber resolution	horizontal	512
	vertical	512
Camera I	cell size	$8.4\mu m \times 9.8\mu m$
	Number of cell	768 x 494
	focal length	16mm
Camera II	cell size	$9.6\mu m \times 7.5\mu m$
	Number of cell	243 x 494
	focal length	12mm

Images are acquired at two different orientations: 30° and 90° . The images used for the experiments are obtained by positioning the camera in the real world coordinate. The positions of the camera are also rotated for obtaining image data

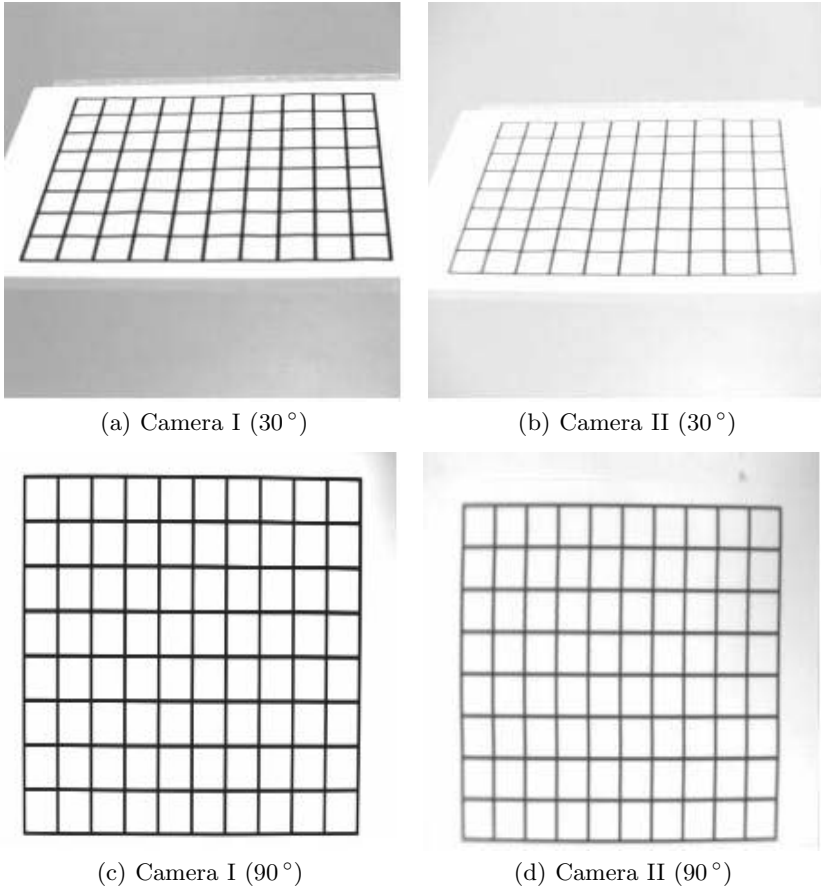
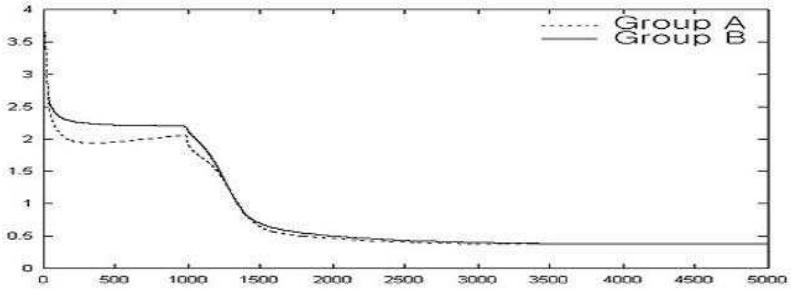


Fig. 4. Images acquired with different tilt angles by two cameras

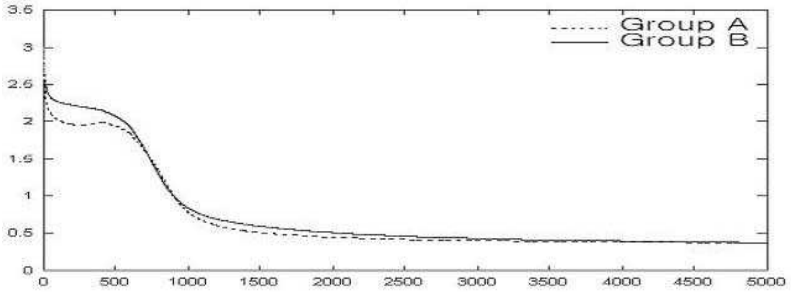
Table 2. Comparison of the proposed calibration method with Tsai's (unit: AEIP)

camera type	orientation	proposed method		Tsai's method	
		Group A	Group B	Group A	Group B
Camera I	30°	0.3790	0.3721	0.7915	0.7582
	90°	0.4373	0.4096	0.4074	0.4254
Camera II	30°	0.3700	0.3718	0.7746	1.0261
	90°	0.4388	0.3940	1.0496	1.0612

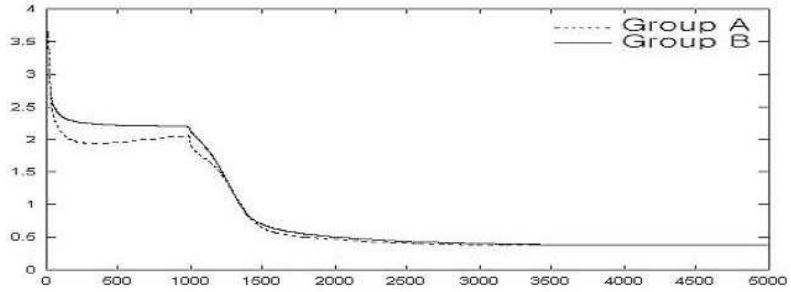
with 30° and 90°. Each image is composed of 99 calibration points (11 × 9), which have an interval of 25mm between columns and an interval of 20mm between rows. Among the calibration points acquired from two images including 99 calibration points for each different heights, 80 randomly selected calibration points in each image are used for training the MLPNN and the remaining 19



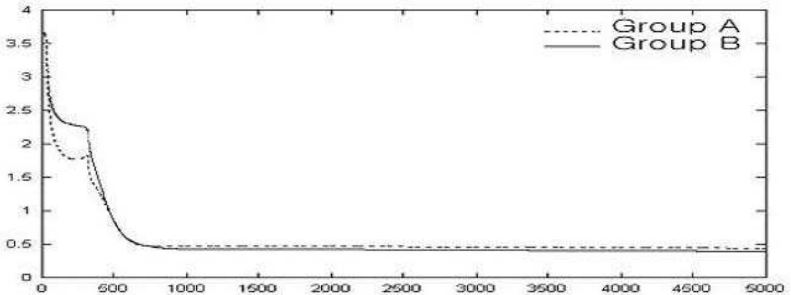
(a) Camera I (30°)



(b) Camera II (30°)



(c) Camera I (90°)



(d) Camera II (90°)

Fig. 5. Graphs of AEIPs with respect to training epoch of neural network

points are used for evaluation of the trained MLPNN. Fig. 4 shows the images with different orientations used in our experiments.

The performance of camera calibration results using artificial neural networks is compared and analyzed with that of Tsai's two stage method, the most widely used approach for explicit camera calibration. In this paper, the average error between the calibrated image coordinates and real world coordinates is used to compare the performance of the camera calibration methods. The average error in pixels (AEIP) is defined as follows:

$$AEIP = \frac{1}{N} \sum_{i=1}^N [(X_{fi} - \hat{X}_{fi})^2 + (Y_{fi} - \hat{Y}_{fi})^2]^{1/2} \quad (7)$$

where $(\hat{X}_{fi}, \hat{Y}_{fi})$ is the estimated image coordinate, which is computed by using calibrated variables from the real coordinate point (X_{wi}, Y_{wi}, Z_{wi}) corresponding to the computer image coordinate (X_{fi}, Y_{fi}) .

For training MLPNN, 80 randomly chosen calibration points are collected, and named as group A. The remaining 19 points are used for testing the performance, and named as group B. Fig. 5 shows AEIPs of group A and group B with respect to training epoch. This experimental results indicate that the AEIPs of group A and group B are similarly converged. This means that the performance of the proposed calibration method on the randomly selected data is equally reliable.

The proposed method is compared with Tsai's two stages method, which finds the physical parameters of the camera using the interrelation between the image coordinates and the known 3D space coordinates. Table 2 shows the test results for both methods. The AEIP of the proposed calibration method is a converged value after 5,000 training epochs. As shown in Table 2, the result of Tsai's two stages method shows inconsistent accuracy according to the camera type and the orientation angle. In particular, Tsai's two stages method shows unreliable performance on the image acquired by a low quality CCTV camera, Camera II. For this particular case, the average improvement of the proposed MLPNN-based method over Tsai's method in terms of AEIP is 150 %.

5 Conclusion

In this paper, we propose a camera calibration method using an artificial neural network. The proposed MLPNN-based implicit method is applied to the estimation of 2D coordinates of an image world with given 3D space coordinates. The proposed method has advantages over Tsai's two stage method in real-time applications as it can be operated in real time after proper training while Tsai's two stage method requires somewhat time consuming procedures for calculating proper parameters for a given task. The proposed method is also more consistent than Tsai's two stage method, since it is not affected by camera orientation. More importantly, the proposed MLPNN-based method is not affected by the quality of the camera. This indicates that the proposed camera calibration method can be consistently applied to all kinds of camera.

Acknowledgments. This work was supported by the Korea Research Foundation Grant funded by the Korea Government (MOEHRD-KRF-2005-042-D00265).

References

1. Xu, Q., Ye, D., Che, R., Huang, Y.: Accurate Camera Calibration with New Minimizing Function. In: Proc. 2006 IEEE ROBOT, pp. 779–784 (2006)
2. Douxchamps, D., Chihara, K.: High-Accuracy and Robust Localization of Large Control Markers for Geometric Camera Calibration. *IEEE Trans. on Pattern Anal. Mach. Intell.* 31, 376–383 (2009)
3. Tsai, R.: An Efficient and Accurate Camera Calibration Technique for 3-D Machine Vision. In: Proc. IEEE Int. Computer Vision and Pattern Recognition, pp. 364–374 (1986)
4. Martins, H.A., Birk, J.R., Kelley, R.B.: Camera Models Based on Data from Two Calibration Plane. *Computer Graphics and Image Processing* 17, 173–180 (1981)
5. Mohr, R., Morin, L.: Relative Positioning from Geometric Invariant. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 139–144 (1991)
6. Rumelhart, D., Hinton, G., Williams, R.: *Parallel Distributed Processing*. MIT Press, Cambridge (1986)

Hierarchical Neural Network Model for Water Quality Prediction in Wastewater Treatment Plants

Qiumei Cong¹, Wen Yu², and Tianyou Chai¹

¹ Key Laboratory of Process Industry Automation,
Northeastern University, Shenyang, 110006, China

² Departamento de Control Automatico
CINVESTAV-IPN, Av. IPN 2508, Mexico D.F., 07360, Mexico
yuw@ctrl.cinvestav.mx

Abstract. Water quality measurement is important for wastewater treatment plants. Up to the present moment, there are not economic on-line sensors for it. In this paper a new soft measurement method is proposed, which uses mechanism model and hierarchical neural networks to resolve a modeling accuracy problem. Since wastewater treatment plants are cascaded processes, hierarchical neural networks can match these structures and predict water quality in inner reactors. By comparing our method with the other soft measurement approaches, we find that based on mechanism model and hierarchical neural networks, the hierarchical model is effective for wastewater treatment plants.

1 Introduction

Active sludge technique is widely used in wastewater treatment plants where contaminations are removed by biological reactions of microorganisms in activated sludge [4]. Therefore, wastewater treatment is purified by separating carbonaceous and nitrogenous contaminants with biological reactions of microorganisms. Chemical oxygen demand (COD) is mostly concerned water quality index in wastewater treatment plants, which is the sum of carbonaceous components, and it is also a key factor in environmental protection. Since COD is an important index for super-nutrition rivers, it is one of the remarkable indexes for the water quality of rivers.

Normally, off-line method is used to obtain COD in wastewater treatment plants. In this way on-line measurement of water quality cannot be applied. Recently, on-line measurement methods are investigated by using potassium dichromate oxidation and ultraviolet absorption in laboratories [13]. However, there is a long way to go to use them in wastewater treatment plants, because they are expensive and difficult to maintain. Soft measurement is an alternative economic method, which uses activated sludge model (ASM) [15]. This model is published by International Association on Water Quality (IAWQ) since 1980's. ASM is

a kind of ‘hard modeling’ or parametric modeling. It includes many differential equations and parameters, which is difficult to be decided under different conditions [4]. In order to reduce the complexity, Multi-Model interpolation procedure was proposed by [10]. Multiple linear approximate model with switching algorithm was obtained by [1]. Multivariable modeling with subspace-based algorithm had been used and a linear time-invariant dynamic model was given by [11].

Components in wastewater can be divided into two types: substrates and microorganisms [4]. Each component has its own biological reactions with different growth and consumption periods. Although mass balances can be applied to biological reactions, there are many uncertainties such as unknown behaviors of microorganisms, operational styles and toxin influences. The wastewater treatment process is also large time-delay process. It is difficult to obtain an exact model of wastewater treatment plant. Recently, some intelligent methods are investigated. [22] used time-delay neural network to model wastewater treatment plant. This black-box modeling approach did not use any mechanism knowledge. A statistical learning method was employed to develop a learning machine based on the physical mechanisms of biological wastewater treatment systems by [3]. [5] and [21] used neural networks to approximate the residual uncertainties between the mechanism model and the plant to improve modeling accuracy. Since wastewater treatment plant is a cascaded process, the above methods can only give the input-output relation between the first block and the last block. On the other hand, ASM can provide all COD values even in the internal blocks.

As discussed above, there are two problems in modeling soft-sensors of water quality: 1) wastewater treatment plant is a big cascade process, one model cannot represent a plant with many sub-systems. 2) ASM model is good, but there are many nonlinear uncertainties. In this paper, we will use two techniques to resolve these problems: hierarchical neural networks and mechanism-based (or ASM-based) model.

The main applications of hierarchical models are on fuzzy systems, because rule-explosion problem can be avoided in hierarchical systems [8], for example, hierarchical fuzzy neural network [18], hierarchical fuzzy systems [14] and hierarchical fuzzy CMAC networks [17]. A statistical learning method was employed to constructing hierarchical models in [3]. Based on Kolmogorov’s theorem, [19] showed that any continuous function can be represented as a superposition of functions with the natural hierarchical structure. Sensitivity analysis of the hierarchical fuzzy model was given in [14].

The key problem for the training of a hierarchical neural model is how to get explicit expression of each internal error. To the best of our knowledge, hierarchical neural system training still is gradient descent [9], they are very complex. The normal training method for multilayer neural networks is back-propagation, which has two phases: modeling error is propagated backward and weights are trained by gradient descent and modeling error. In this paper, the first phase of back-propagation is used to hierarchical neural networks training, such that the training process becomes simpler.

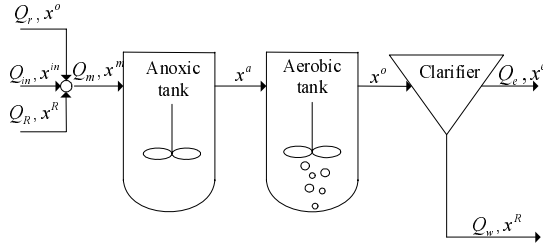


Fig. 1. Plant configuration

2 Dynamic Model of Wastewater Treatment Process

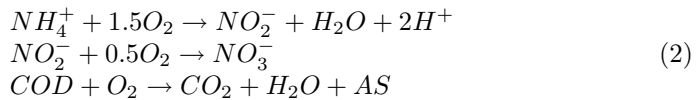
The wastewater treatment plant studied throughout this paper is an anoxic/oxidant nitrogenous removal process [12]. It consists of two biodegradation tanks and a secondary clarifier in series form, see Figure 1. Here, Q_{in} , Q_m , Q_r , Q_R and Q_w denote flow of wastewater to be disposed, flow of mixed influent, flow of internal recycles, flow of external recycles and flow of surplus sludge respectively. Water quality indices such as COD, BOD₅ (Biological Oxygen Demand), NH₄-N (ammonia), nitrate and SS (Suspended Solid) are decomposed into those components in ASM [4]. The state variable \mathbf{x} is defined as

$$\mathbf{x} = [S_I, S_S, X_I, X_S, X_P, X_{BH}, X_{BA}, S_{NO}, S_{NH}, S_O, S_{ND}, X_{ND}, S_{alk}]^T \quad (1)$$

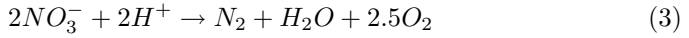
where S_I is soluble inert, S_S is readily biodegradable substrate, X_I is suspended inert, X_S is slowly biodegradable substrate, X_P is suspended inert products, X_{BH} is autotrophic biomass, X_{BA} is heterotrophic biomass, S_{NO} is nitrate, S_{NH} is ammonia, S_O is soluble oxygen, S_{ND} is soluble organic nitrogen, X_{ND} is suspended organic nitrogen, S_{alk} is alkalinity.

\mathbf{x}^c , $c = \{in, m, a, o, R, e\}$ denotes the component concentration in location c , \mathbf{x}^{in} is component concentration of wastewater to be processed, \mathbf{x}^m is component concentration of mixed influent, \mathbf{x}^a is component concentration of anoxic effluent, \mathbf{x}^o is component concentration of aerobic effluent, \mathbf{x}^R is component concentration from the thickening zone of clarifier, \mathbf{x}^e is component concentration from clarification zone of clarifier. Liquid and solid components in \mathbf{x}^o are separated in the clarifier. Clarification effluent \mathbf{x}^e is discharged into recipient river, part of thickening effluent \mathbf{x}^R is recycled to the anoxic tank, while surplus is abandoned.

Denitrification and nitrification occur in anoxic and aerobic tanks to remove nitrogenous and carbonaceous contaminations respectively. Each tank is equipped by stirrers to insure complete mix. The clarifier is a device for separation of liquid and solid substances. The aim of nitrification reactions is to transform ammonia into nitrite or nitrate, and consume biodegradable COD by heterotrophic microorganisms. In this tank, there are two major reaction processes.



where COD represents carbonous contamination; *AS* denotes activated sludge. Nitrite is recycled from the aerobic tanks. It is deoxidized by autotrophic microorganisms in denitrification phase. The reaction is



The water quality index COD depends on the control input

$$\mathbf{w} = [w_i] = [Q_{in}, Q_r, Q_R, Q_w]^T \tag{4}$$

As well as influent quality \mathbf{x}^{in} , i.e.,

$$COD = f(Q_{in}, Q_r, Q_R, Q_w, \mathbf{x}^{in})$$

COD is also affected by other external factors such as temperature, flow distribution and toxin. It is very difficult to find the nonlinear function $f(\cdot)$ [15]. COD is synthesized by carbonaceous components, it is the final output

$$COD = \sum_{i=1}^7 x_i^e \tag{5}$$

Mass balance of components in the anoxic tank is given by

$$\dot{x}_i^a(t) = \frac{1}{V} (w_1(t) + w_2(t) + w_3(t)) (x_i^m(t) - x_i^a(t)) + r_i^a(x_1^a(t), \dots, x_{13}^a(t)) \tag{6}$$

where

$$x_i^m(t) = \frac{w_1(t) x_i^{in}(t) + w_2(t) x_i^o(t) + w_3(t) x_i^R(t)}{w_1(t) + w_2(t) + w_3(t)}$$

Mass balance of components in the aerobic tank is given by

$$\dot{x}_i^o(t) = \frac{1}{V} (w_1(t) + w_2(t) + w_3(t)) (x_i^a(t) - x_i^o(t)) + r_i^o(x_1^o(t), \dots, x_{13}^o(t)) \tag{7}$$

here $i = 1, \dots, 13$; r_i^c denotes the reaction rates of i component in location c , which have nonlinear relation with component concentrations \mathbf{x}^c . Default values of kinetic and stoichiometric parameters can be found in [4].

In order to maintain a proper concentration in biological reactors, a specified sludge age is needed. It is assumed that the settling process is capable of producing the thickened sludge. The model of the clarifier is:

$$\begin{aligned} x_i^R(t) &= \begin{cases} \lambda_i x_i^o, & i = 3, 4, 5, 6, 7, 12 \\ x_i^o, & i = 1, 2, 8, 9, 10, 11, 13 \end{cases} \\ x_i^e(t) &= \begin{cases} 0, & i = 3, 4, 5, 6, 7, 12 \\ x_i^o, & i = 1, 2, 8, 9, 10, 11, 13 \end{cases} \end{aligned} \tag{8}$$

where the thickening factor is

$$\lambda_i = \frac{x_i^R}{x_i^o} = \begin{cases} w_1 + w_3 - (V_a + V_o)/\theta, & i = 3, 4, 5, 6, 7, 12 \\ 1, & i = 1, 2, 8, 9, 10, 11, 13 \end{cases}$$

here θ denotes the sludge age, V_a is the volume of anoxic tank, V_o is the volume of aerobic tank. The fitness function adopted in the model is discussed in our previous paper [2].

3 Reaction Rates Identification via Neural Network

From (5), (6) and (7) we know each biological reactor in wastewater treatment plants can be described by the following dynamic equation

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{x}_b(t) + \varphi(\mathbf{x}(t)) \tag{9}$$

where $\mathbf{x} \in R^{13}$ is inner state which is defined in (1), $\mathbf{x}_b(t) \in R^4$ is input which is defined in (4), φ is reaction rates, $\varphi(\cdot) \in R^{13}$, $A = -\frac{w_1(t)+w_2(t)+w_3(t)}{V}$, $B = \frac{w_1(t)+w_2(t)+w_3(t)}{V}$.

The following difference technique is used to get the discrete-time states of the system

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t)$$

Let us define $s_1 = A\mathbf{x}_k$, $s_2 = A(\mathbf{x}_k + s_1)$, $s_3 = A(\mathbf{x}_k + \frac{s_1+s_2}{4})$. If $|\frac{s_1-2*s_3+s_2}{3}| \leq \frac{|\mathbf{x}_k|}{1000}$ or $|\frac{s_1-2*s_3+s_2}{3}| < 1$, then $\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{s_1+4s_3+s_2}{6}$, $k = 0, 1, 2 \dots$. The discrete-time model of (9) is

$$\mathbf{x}(k+1) = G\mathbf{x}(k) + H\mathbf{x}_b(k) + \gamma(\mathbf{x}(k)) \tag{10}$$

where $H = B$, the nonlinear parts are included in $\gamma(\mathbf{x}(k))$.

There are eight types of kinetic reactions occur in the activated sludge process: 1) aerobic growth of heterotrophs, 2) anoxic growth of heterotroph, 3) aerobic growth of autotroph, 4) decay of heterotroph, 5) decay of autotroph, 6) ammonification of soluble organic nitrogen, 7) hydrolysis of entrapped organics, 8) hydrolysis of entrapped organic nitrogen. It is difficult to obtain the exact reaction rate $\gamma(\mathbf{x}(k))$ due to these unknown and complex behaviors of the microorganisms and biological reactions.

Recent results show that neural network technique seems to be very effective to identify a broad category of complex nonlinear systems when complete model information cannot be obtained. Let us define an additional variable

$$z(k) = \mathbf{x}(k+1) - G\mathbf{x}(k) - H\mathbf{x}_b(k) \tag{11}$$

From (10) we know

$$z(k) = \gamma(\mathbf{x}(k)) \tag{12}$$

The output of each tank is

$$y(k) = \mathbf{x}(k) \in R^{13}$$

Now we use a multilayer neural network to identify $\gamma(\mathbf{x}(k))$

$$\widehat{z}(k) = W(k) \phi[V(k) \mathbf{x}(k)] \quad (13)$$

where $\widehat{z}(k)$ is the output of neural networks, $W(k) \in R^{1 \times m}$ and $V(k) \in R^{m \times n}$ are the weights in output layer and hidden layer, ϕ is m -dimension vector function. The typical presentation of the element $\phi_i(\cdot)$ is sigmoid function. The normal back-propagation method can be used to train the weights $W(k)$ and $V(k)$, which is derived by minimizing the following performance index

$$J = \frac{1}{2} \|e(k)\|^2 = \frac{1}{2} [(\widehat{z}(k) - z(k))]^2$$

We use gradient descent learning law $\Delta w_i(k) = -\eta \frac{\partial J}{\partial w_i}$, $\Delta v_{i,j}(k) = -\eta \frac{\partial J}{\partial v_{i,j}}$, where $W(k) = [w_i(k)]$, $V(k) = [v_{i,j}(k)]$, and chain rule

$$\begin{aligned} \frac{\partial J}{\partial w_i} &= \frac{\partial J}{\partial \widehat{z}} \frac{\partial \widehat{z}}{\partial w_i} = e(k) \phi_i \\ \frac{\partial J}{\partial v_{i,j}} &= \frac{\partial J}{\partial \widehat{z}} \frac{\partial \widehat{z}}{\partial \phi_i} \frac{\partial \phi_i}{\partial v_{i,j}} = e(k) \phi' W^T(k) \mathbf{x}^T(k) \end{aligned}$$

The final learning law is

$$\begin{aligned} W(k+1) &= W(k) - \eta \phi e(k) \\ V(k+1) &= V(k) - \eta e(k) \phi' W^T(k) \mathbf{x}^T(k) \end{aligned} \quad (14)$$

In order to insure stable training, the learning rate η must be very small, so back-propagation method is very slow. A modified time-varying learning rate can guarantee stability and the training speed is fast [16]. The training form is the same as (14), but

$$\eta = \eta_k = \frac{\eta_0}{1 + \|\phi' W^T(k) \mathbf{x}^T(k)\|^2 + \|\phi\|^2}, \quad 0 < \eta_0 \leq 1 \quad (15)$$

4 Soft-Sensor of Water Quality via Hierarchical Neural Networks

Wastewater treatment plant is a cascaded process with several reactors and a clarifier in series. As discussed above, hierarchical neural networks are suitable for modeling this process. Each neural network corresponds to a reaction rate in a reactor. Figure 2 depicts how to apply back-propagation technique to hierarchical neural networks, here

$$\begin{aligned} \mathbf{y}_1(k) &= \mathbf{x}_1(k+1) - G_1 \mathbf{x}_1(k) - H_1 \mathbf{x}_u(t) + \gamma_1(\mathbf{x}_1(k)) \\ \mathbf{y}_2(k) &= \mathbf{x}_2(k+1) - G_2 \mathbf{x}_2(k) - H_2 y_1(k) + \gamma_2(\mathbf{x}_2(k)) \end{aligned}$$

In our approach, the output of each block is given by (13). The objective is to train the weights of the two NNs so that the error between the hierarchical

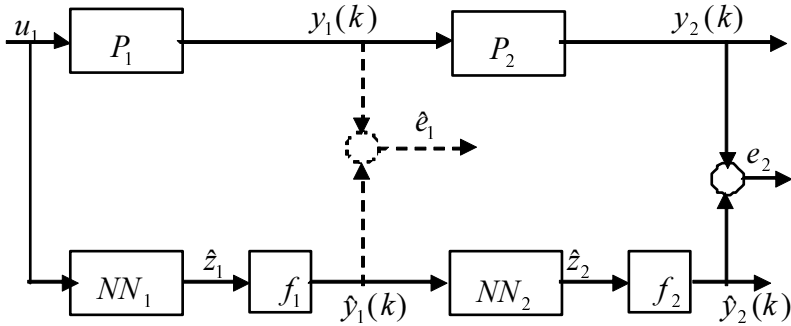


Fig. 2. Two blocks hierarchical neural networks

neural networks. $\hat{y}_2(k)$ and the output of the plant (10) $y_2(k)$ is minimized. The performance index is defined as

$$J = \frac{1}{2}e_2^2 \quad e_2 = \hat{y}_2(k) - y_2(k)$$

By (11) and chain rule

$$\frac{\partial J}{\partial W_2} = \frac{\partial J}{\partial \hat{y}_2(k)} \frac{\partial \hat{y}_2(k)}{\partial \hat{z}_2} \frac{\partial \hat{z}_2}{\partial W_2} = e_2 \frac{\partial \hat{z}_2}{\partial W_2}$$

So the gradient descent learning law for NN_2 is

$$\begin{aligned} W_2(k+1) &= W_2(k) - \eta \phi_2 e_2(k) \\ V_2(k+1) &= V_2(k) - \eta e_2(k) \phi_2' W_2^T(k) \mathbf{x}_2^T(k) \end{aligned} \quad (16)$$

where $\eta > 0$ is the learning rate. From Figure 2 we know

$$\begin{aligned} \frac{\partial J}{\partial W_1} &= \frac{\partial J}{\partial \hat{y}_2(k)} \frac{\partial \hat{y}_2(k)}{\partial \hat{z}_2} \frac{\partial \hat{z}_2}{\partial \hat{y}_1(k)} \frac{\partial \hat{y}_1(k)}{\partial \hat{z}_1} \frac{\partial \hat{z}_1}{\partial W_1} \\ &= e_2 \frac{\partial \{W(k)\phi[V(k)\mathbf{x}_1]\}}{\partial \hat{x}_1} \frac{\partial \hat{z}_1}{\partial W_1} \\ &= e_2 W_2(k) V_2(k) \phi' \frac{\partial \hat{z}_1}{\partial W_1} \\ &= e_1 \frac{\partial \hat{z}_1}{\partial W_1} \end{aligned}$$

So the gradient descent learning law for NN_1 is

$$\begin{aligned} W_1(k+1) &= W_1(k) - \eta \phi_1 e_1(k) \\ V_1(k+1) &= V_1(k) - \eta e_1(k) \phi_1' W_1^T(k) \mathbf{x}_1^T(k) \end{aligned} \quad (17)$$

We can see that (17) has the same form as (16). We can train these two blocks independently, the only difference is

$$e_1 = e_2 W_2(k) V_2(k) \phi'$$

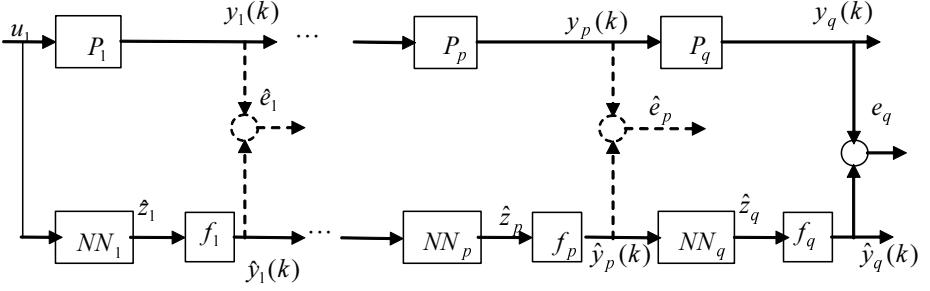


Fig. 3. General case of cascade plant modeling via hierarchical neural network

Figure 3 shows a general case in wastewater treatment plants. We can train each block independently if we know the modeling errors between neural networks and corresponding virtual process blocks $\hat{e}_1, \hat{e}_2 \dots \hat{e}_p$. For block q , we use gradient descent learning law

$$W_q(k+1) = W_q(k) - \eta \phi_q(k) e_q(k)$$

For block p , we need to calculate $\frac{\partial J}{\partial W_p} = \frac{\partial J}{\partial \hat{y}_q} \frac{\partial \hat{y}_q}{\partial \hat{z}_p} \frac{\partial \hat{z}_p}{\partial W_p}$ to update W_p . Because $\frac{\partial J}{\partial \hat{y}_p} = e_p(k)$, $\frac{\partial \hat{y}_p}{\partial \hat{z}_q} = \phi'_p W_p^T V_p$, the gradient learning for W_p is

$$\begin{aligned} W_p(k+1) &= W_p(k) - \eta \phi_p(k) e_p(k) \\ e_p &= e_q \phi'_p W_p^T V_p \end{aligned} \quad (18)$$

The identification error $e_o(k)$ can be propagated to the other blocks, see Figure 3. If we know the output error of each block, we can train this block. The training procedure is as follows:

1. Calculate the output of each mechanism-based neural network by (13). The output of multilayer neural networks should be the inputs of the next level.
2. Calculate the modeling error for each block. We start from the last block, the identification error is

$$e_o(k) = \hat{y}(k) - y(k) \quad (19)$$

where $e_o(k)$ is the identification error of the whole system, $\hat{y}(k)$ is the output of the whole hierarchical neural networks, y is the output of the cascaded plant. Then we propagate the error back from the structure of the hierarchical neural networks by (18).

3. Train the weights for each block independently. For p -th block the gradient descent algorithm is

$$\begin{aligned} W_p(k+1) &= W_p(k) - \eta \phi_p(k) e_p(k) \\ V_p(k+1) &= V_p(k) - \eta e_p(k) \phi'_p W_p^T(k) \mathbf{x}_p^T(k) \end{aligned} \quad (20)$$

5 Application to a Wastewater Treatment Plant

In this section, we will use the above modeling approach to the North Wastewater Treatment Plant in Shenyang, China. The plant including 6 water lines in parallel to deal with municipal wastewater, where lines No.1~No.3 use traditional activated sludge technics and lines No.4~No.6 use anoxic-aerobic (A/O) technics to remove nitrogen. We will use line No.4 as a case study, which consists of an anoxic reactor with the volume of $7772.5m^3$, an aerobic reactor with the volume of $10326m^3$, and a secondary settler with the height of $4.2m$ and the diameter of $57m$. The set-up of line No.4 is shown in Figure 1. The fluid volumes (m^3/h) are: $Q_{in} \in [2778, 3700]$, $Q_R \in [1389, 3700]$, $Q_r \in [2778, 7400]$, $Q_w \in [0, 760]$. It is supposed that there are no biological reactions in secondary settler, each reactor is completely mixed regardless of the influences of stream status.

Steady simulations are implemented before dynamic modeling. From these simulations, initial values of dynamic model are obtained. There are some missing data and outliers in the measured data. Principal component analysis (PCA) and Expectation-Maximum algorithms are used to recover the data [20]. The iterative robust least square method is employed to reduce the influence of noise, outliers and missing data inherent in measured values. The pretreated real data are shown as Figure 2. We can see that the missing data from August to October are recovered, although there are large offset after being coordinated. The big changes in COD and SS after the first ten days of May is caused by snow melt, while NH_4-N is not effected by this phenomenon. The effluents COS and SS are relatively smooth after A/O biological treatment. It indicates the plant possesses redundancy ability to fluctuation of water qualities. The removal efficiency of NH_4-N is relative low to other contaminations, which should be improved by optimal control using existing equipment.

The resulting steady values of anoxic and aerobic reactors are shown in Table 1, which is from the data in year 2003.

Table 1. Steady values of anoxic and aerobic reactors

	S_S	X_{BH}	X_S	X_I	S_{NH}	S_I	
anoxic	1.2518	3249	74.332	642.4	7.9157	38.374	
aerobic	0.6867	3244.8	47.392	643.36	0.1896	38.374	
	S_{ND}	X_{ND}	S_O	X_{BA}	S_{NO}	X_P	S_{alk}
	0.7868	5.7073	0.0001	220.86	3.9377	822.19	4.9261
	0.6109	3.7642	1.4988	222.39	12.819	825.79	3.7399

The modeling strategy is shown as Figure 3. The input data to the anoxic block and the first neural networks is $\Omega = [S_I S_S X_I X_S X_P X_{BH} X_{BA} S_{NO} S_{NH} S_O S_{ND} X_{ND} S_{alk} Q_{in} Q_R Q_r Q_w]$, which is converted from influent indices $[COD_i, NH_{4,i}]$ of line No.4, $[Q_{in}, Q_r, Q_R, Q_w]^T$ is control input \mathbf{w} , the output is effluent indices COD_e , the output of the first neural networks is the reaction rates $[r_2^a r_7^a r_8^a]^T$. The output of mechanism anoxic block is effluent

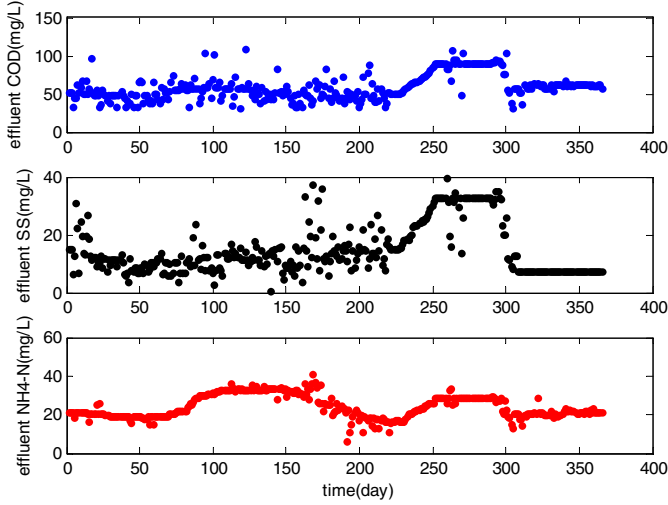


Fig. 4. Effluent qualities in 2003

components of anoxic reactor $\hat{\mathbf{x}}^a$. The outputs of neural networks are sent to the corresponding mechanism model. The neural networks in the second level are for the aerobic process, whose inputs are the outputs from anoxic block. The output of these networks is the reaction rates $[r_2^o \ r_6^o \ r_9^o \ r_{10}^o]^T$. Furthermore, these four values are sent to the second mechanism block. The final output of the cascaded process is effluent *COD* which comes from the secondary settler.

The design choices are as follows: both two neural networks have one hidden layer, each hidden layer has 50 hidden nodes. The training algorithms of each neural model are (14) and (15), here the activation function $\phi_i(\cdot) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, $\eta_0 = 1$, the initial weights of $W(1)$ and $V(1)$ are random number between $[0, 1]$.

Dynamic modeling uses the resulting steady values of steady simulation as initial values with hydraulic residence time of $10.8h$ and sludge age of $15d$. 100 input/output data pairs from the records of year 2003 are used as the training data, the other 30 input/output pairs as the testing data. The testing results of effluent *COD* are shown in Figure 5.

We compare the hierarchical neural networks (HNN) with the other three modeling methods. They are activated sludge models (ASM) [4], linear models (LM) [6], neural networks (NN) [5]. The model parameters of ASM model are the default values in [4]. The numbers of concerned variables in linear models are selected 2, 3 and 4. The hidden nodes of neural networks (NN) are chosen as 30, 50 and 70, they are the same as those in hierarchical neural network (HNN). The initial values for all weights are chosen randomly from interval $(0, 1)$. The experiments conditions are: the software is Matlab under Windows XP, the hardware is a PC with Pentium P4 2.20GHz.

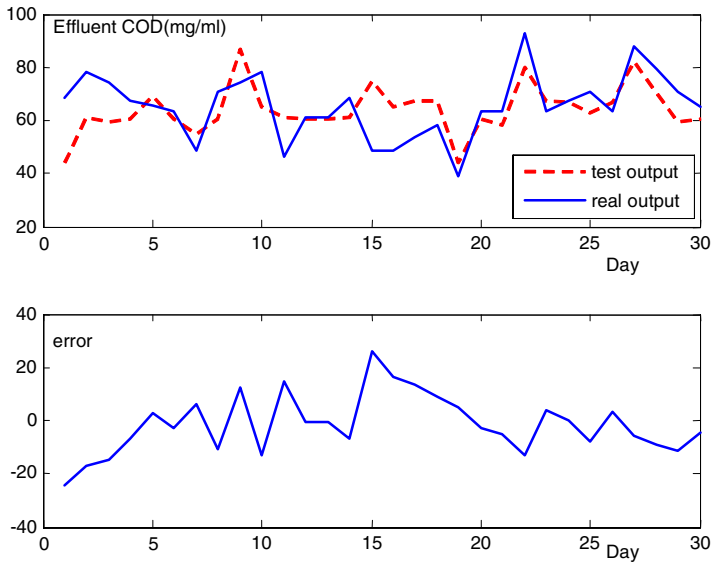


Fig. 5. The results via hierarchical neural networks

6 Conclusions

In this paper, we present a new soft sensor for water quality in wastewater treatment plants. It uses two techniques: mechanism model (ASM) and hierarchical neural networks (HNN). The combination of these two techniques can improve modeling accuracy, and provide water quality in inner reactors. These are advantages compared with the other soft measurement approaches for water quality.

References

1. Anderson, J.S., Kim, H., McAvoy, T.J., Hao, O.J.: Control of an alternating aerobic–anoxic activated sludge system - Part 1: development of a linearization-based modeling approach. *Control Engineering Practice* 8, 271–278 (2000)
2. Cong, Q., Chai, T.: Effluent COD of SBR Process Prediction Model Based on Fuzzy-Neural Network. In: *International Conference on Neural Networks and Brain*, vol. 2, pp. 821–825 (2005)
3. Guergachi, A.A., Patry, G.G.: Constructing a model hierarchy with background knowledge for structural risk minimization: Application to biological treatment of wastewater. *IEEE Trans. Systems, Man and Cybernetics - Part A: Systems and Humans* 36, 1–11 (2006)
4. Henze, M., Gujer, W., Mino, T., Matsuo, T., Wentzel, M.C., Marais, G.v.R., van Loosdrecht, M.C.M.: Activated Sludge Model No.2d, ASM2D. *Wat. Sci. Tech.* 39, 165–182 (1999)

5. Lee, D.S., Vanrolleghem, P.A., Park, J.M.: Parallel hybrid modeling methods for a full-scale cokes wastewater treatment plant. *Journal of Biotechnology* 115, 317–328 (2005)
6. Lin, Z.-C., Wu, W.-J.: Multiple linear regression analysis of the overlay accuracy model. *IEEE Transactions on Semiconductor Manufacturing* 12(2), 229–237 (1999)
7. Narendra, K.S., Parthasarathy, K.: Identification and Control of Dynamical Systems Using Neural Networks. *IEEE Trans. Neural Networks* 1(1), 4–27 (1990)
8. Raju, G.V.S., Zhou, J., Kisner, R.A.: Hierarchical fuzzy control. *Int. J. of Control* 54, 1201–1216 (1991)
9. Rumelhart, D., McClelland, J.: *Parallel distributed processing: exploitations in the microstructure of cognition*, vol. 1 and 2. MIT press, Cambridge (1986)
10. Smets, I.Y., Haeghebaert, J.V., Carrette, R., Impe, J.F.V.: Linearization of the Activated Sludge Model ASM1 for fast and reliable predictions. *Wat. Res.* 37, 1831–1851 (2003)
11. Sotomayor, O.A.Z., Park, S.W., Garcia, C.: Multivariable identification of an activated sludge process with subspace-based algorithms. *Control Engineering Practice* 11, 961–969 (2003)
12. Takacs, I., Patry, G.G., Nolasco, D.: A dynamic model of the clarification thickening process. *Wat. Res.* 25, 1263–1271 (1991)
13. Vanrolleghem, P.A., Lee, D.S.: On-line monitoring equipment for wastewater treatment processes: state of the art. *Water Science and Technology* 47(2), 1–34
14. Wang, L.X.: Analysis and Design of Hierarchical Fuzzy Systems. *IEEE Transactions on Fuzzy Systems* 7(3), 617–624 (1999)
15. Weijers, S.R.: Modelling, identification and control of activated sludge plants for nitrogen removal. PhD thesis, Technology University of Eindhoven, The Netherlands (2000)
16. Yu, W., Li, X.: Discrete-time neuro identification without robust modification. *IEE Proceedings - Control Theory and Applications* 150(3), 311–316 (2003)
17. Yu, W., Ortiz, F., Moreno, M.A.: Hierarchical fuzzy CMAC for nonlinear systems modeling. *IEEE Transactions on Fuzzy Systems* (accepted for future publication)
18. Yu, W., Moreno, M.A., Ortiz, F.: System identification using hierarchical fuzzy neural networks with stable learning algorithm. *Journal of Intelligent & Fuzzy Systems* 18, 171–183 (2007)
19. Zeng, X.J., Keane, J.A.: Approximation Capabilities of Hierarchical Fuzzy Systems. *IEEE Transactions on Fuzzy Systems* 13(5), 659–672 (2005)
20. Zhao, L.J., Chai, T.Y.: Wastewater BOD forecasting model for optimal operation using robust time-delay neural network. In: Wang, J., Liao, X.-F., Yi, Z. (eds.) *ISNN 2005. LNCS*, vol. 3498, pp. 1028–1033. Springer, Heidelberg (2005)
21. Zhao, H., Hao, O.J., McAvoy, T.J.: Approaches to modeling nutrient dynamics: ASM2, simplified process model and neural nets. *Water Science Technology* 39(1), 227–234 (1999)
22. Zhu, J.B., Zurcher, J., Rao, M., Meng, M.Q.-H.: An on-line wastewater quality predication system based on a time-delay neural network. *Engineering Applications of Artificial Intelligence* 11, 747–758 (1998)

Third Generation Neural Networks: Spiking Neural Networks

Samanwoy Ghosh-Dastidar¹ and Hojjat Adeli²

¹ Department of Biomedical Engineering, The Ohio State University

² Abba G. Lichtenstein Professor, Departments of Biomedical Engineering, Biomedical Informatics, Civil and Environmental Engineering and Geodetic Science, Electrical and Computer Engineering, and Neuroscience, The Ohio State University, 470 Hitchcock Hall, 2070 Neil Avenue, Columbus, Ohio 43210

Abstract. Artificial Neural Networks (ANNs) are based on highly simplified brain dynamics and have been used as powerful computational tools to solve complex pattern recognition, function estimation, and classification problems. Throughout their development, ANNs have been evolving towards more powerful and more biologically realistic models. In the last decade, the *third generation* Spiking Neural Networks (SNNs) have been developed which comprise of *spiking* neurons. Information transfer in these neurons models the information transfer in biological neurons, i.e., via the precise timing of spikes or a sequence of spikes. Addition of the temporal dimension for information encoding in SNNs yields new insight into the dynamics of the human brain and has the potential to result in compact representations of large neural networks. As such, SNNs have great potential for solving complicated time-dependent pattern recognition problems defined by time series because of their inherent dynamic representation. This article presents an overview of the development of spiking neurons and SNNs within the context of feedforward networks, and provides insight into their potential for becoming the next generation neural networks.

1 Introduction

Artificial neural networks (ANNs), inspired by the structure and function of the human brain, have been used as powerful computational tools to solve complex pattern recognition, function estimation, and classification problems not amenable to other analytical tools [1,2,3,4,5,6]. Over time, ANNs have evolved into more powerful and more biologically realistic models [7,8,9,10,11]. Improved understanding of the brain and its modes of information processing has led to the development of networks such as feedforward neural networks [12,13], recurrent networks [14,15], radial basis function neural networks [16,17,18], self-organizing maps, modular neural networks, and dynamic neural networks [19,20,21].

Feedforward ANNs are the most common and utilize various mechanisms for a forward transfer of information across the neural network starting from the input node to the output node. The popularity of feedforward ANNs stems from their

conceptual simplicity and the fact that the primary (but not the only) mode of information transfer in both real and artificial neural networks is feedforward in nature [22,23,24,25]. In fact, other modes of information transfer often involve or are based on feedforward mechanisms to some degree.

Although ANNs have gone through various stages of evolution, until recently, there had not been many attempts to categorize generations of neural networks. This is a particularly difficult task because ANN developments have branched out in many directions and it would not be accurate to label one development as more advanced than another. In addition, such a categorization is subjective and dependent on what is considered advancement. However, in the authors' opinion, if a single clearly identifiable, major conceptual advancement were to be isolated, it would be the development of the mathematically-defined activation or transfer function as the information processing mechanism of the artificial neuron. Due to the importance of the activation function in feedforward ANNs, the discussion on generations of ANN in this article is restricted to the evolution of the artificial neuron from the perspective of feedforward neural networks.

2 Information Encoding and Evolution of Spiking Neurons

Studies of the cortical pyramidal neurons have shown that the timing of individual spikes as a mode of encoding information is very important in biological neural networks [26,27,28]. A presynaptic neuron communicates with a postsynaptic neuron via trains of spikes or action potentials. Biological spikes have a fixed morphology and amplitude [29]. The transmitted information is usually encoded in the frequency of spiking (*rate encoding*) and/or in the timing of the spikes (*pulse encoding*). Pulse encoding is more powerful than rate encoding in terms of the wide range of information that may be encoded by the same number of neurons [30]. In fact, rate encoding can be considered to be a special case of pulse encoding. If the spike timings are known, the average firing rate can be computed.

The early *first generation* neurons developed in the 1940s and 1950s did not involve any encoding of the temporal aspect of information processing. These neurons acted as simple integrate-and-fire units which fired *if* the *internal state* (defined as the weighted sum of inputs to each neuron) reached a threshold. It did not matter when the threshold was exceeded. Translating this assumption to a biological perspective, it implied that all inputs to the neuron were synchronous, i.e. contributed to the internal state at exactly the same time and therefore, could be directly summed. However, unlike biological neurons, the magnitude of the input was allowed to contribute to the internal state. Arguably, this may have represented a primitive form of rate encoding in the sense that a larger input (representing a higher firing rate of the presynaptic neuron) may cause the postsynaptic neuron to reach the threshold. For the sake of simplicity, the mathematical abstraction avoided the modeling of the actual spike train and the input from the presynaptic neuron approximated the average firing rate of

the presynaptic neuron. The *fire* state for the postsynaptic neuron was a binary-valued output which returned a value of 1 if the neuron fired and 0 otherwise. This implied that the output from the postsynaptic neuron was not based on rate encoding.

The *second generation* neurons developed from the 1950s to 1990s were also based loosely on rate encoding and defined the internal state in a similar manner. However, they used a mathematically-defined activation function, often a smooth sigmoid or radial basis function, instead of a fixed threshold value, for output determination [27]. In the postsynaptic neuron, the activation function was used to transform the input into a proportionate output which approximated the average firing rate of the postsynaptic neuron. With this development, it became possible for the output to be real-valued. In contrast to the first generation neurons, even the postsynaptic neuron could generate rate encoded information. This model gained widespread acceptance as processing elements in feedforward ANNs because it was compatible with the Rumelhart's widely-used backpropagation (BP) training algorithm [31] which required a continuous and differentiable activation function. The model was significantly more powerful than the one based on first generation neurons and could solve complex pattern recognition problems (the most notable of which in the 1950s was the XOR problem) [32,33,34,35,36,37,38,39,40]. However, the computational power of the neuron still did not reach its full potential because the temporal information about individual spikes was not represented.

In the last decade, *spiking* neurons have been developed and adapted for ANNs to overcome this shortcoming by communicating via the precise timing of spikes or a sequence of spikes. In the literature, spiking neurons have been referred to as *third generation* neurons. Similar to the first generation neurons, a spiking neuron acts as an integrate-and-fire unit and has an *all or none* response. The spiking neuron, however, has an inherent dynamic nature characterized by an internal state which changes with time and each postsynaptic neuron fires an action potential or spike at the time instance its internal state exceeds the neuron threshold. Similar to biological neurons, the magnitude of the spikes (input or output) contains no information. Rather, all information is encoded in the timing of the spikes as discussed in the next section. Even though spiking neurons are discussed within the context of feedforward networks in this article, it must be noted that their application is not limited to only feedforward networks. Spiking neurons have also been used with ANNs similar in concept to Radial Basis Function Neural Networks and Self Organizing Maps with applications in unsupervised clustering and pattern classification.

3 Mechanism of Spike Generation in Spiking Neurons

In general, action potentials or spikes from various presynaptic neurons reach a postsynaptic neuron at various times and induce *postsynaptic potentials* (PSPs). The PSP represents the internal state of the postsynaptic neuron induced in response to the presynaptic spike and is affected by synaptic characteristics

such as travel time or delay through the synapse, strength of the synaptic connection, and other biological factors some of which are unknown. Multiple neurons, each with multiple spikes, induce multiple PSPs over time. The postsynaptic neuron acts as a temporal integrator of PSPs induced by all presynaptic neurons and fires when the integrated internal state crosses a threshold.

The effects of various presynaptic spike trains on the postsynaptic potential and the postsynaptic output spike train are illustrated in Fig. 1. In the first two cases, Figs. 1(a) and 1(b), each spike train is considered individually whereas in the third case, Fig. 1(c), the combined effect of the two spike trains shown in Figs. 1(a) and 1(b) is illustrated. Each spike train consists of a sequence of three spikes. The first and the third spike in the presynaptic spike trains occur at the same time instant. The timing of the second spike, however, is different in the two cases. From the perspective of rate encoding, both these spike trains are identical, i.e. the average firing frequency is identical (3 per given time period). This highlights the approximate nature and lower computational power of rate encoding which makes it impossible to differentiate between the two cases in Figs. 1(a) and 1(b).

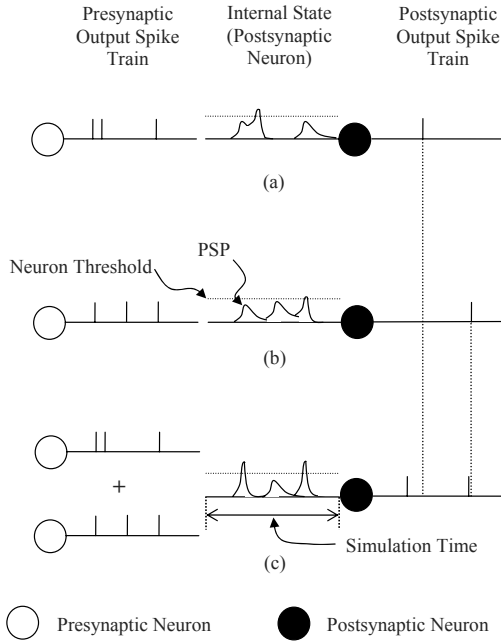


Fig. 1. The effect of various presynaptic spike trains on the postsynaptic potential and the postsynaptic output spike train. (a) and (b) show two spike trains and their individual effects on the postsynaptic neuron, and (c) shows the combined effect of the aforementioned two spike trains on the postsynaptic neuron.

In contrast, the timing of the spikes is considered in pulse encoding. Each spike in the spike train induces a PSP in the postsynaptic neuron at different times. The PSPs are temporally integrated to compute the internal state of the postsynaptic neuron over time as shown in Figs. 1(a) and 1(b). The internal states in the two cases are entirely different and their values exceed the neuronal threshold at different times. This leads to different output spike times from the postsynaptic neuron. An additional source of variation in the PSP is the dependence of the internal state of the postsynaptic neuron on the time of its own output spike. The internal state of a postsynaptic neuron in response to a presynaptic spike is shown in Fig. 2. Had the threshold not been exceeded the internal state of neuron in Fig. 2 would have been represented by the dashed line. The solid line in Fig. 2 shows the internal state of neuron when the threshold is exceeded. Immediately after the firing of an output spike, the internal state of the neuron exhibits a sharp decrease as a result of various biological processes. This phase is known as *repolarization* (Fig. 2) [29,41].

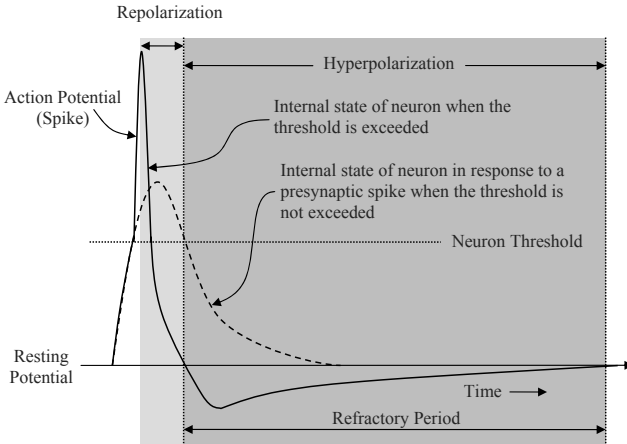


Fig. 2. The internal state of a postsynaptic neuron in response to a presynaptic spike (not shown in the figure) showing the action potential, and repolarization and hyperpolarization phases

In the third case shown in Fig. 1(c), both presynaptic spike trains are input simultaneously to the postsynaptic neuron by two presynaptic neurons. In this case, the internal state of the postsynaptic neuron is not simply the sum of the internal states in the first two cases. An additional factor needs to be considered for the postsynaptic neuron. After the firing of a spike and the resultant sharp decrease in the internal state of the neuron, the internal state is kept at a value lower than the resting potential of the neuron (Fig. 2) by various biological processes that are beyond the scope of this discussion. This phase is known as *hyperpolarization* and shown in Fig. 2 [29,41]. As a result, it becomes difficult for the neuron to reach the threshold and fire again for a certain period of time, known as *refractory period*

(Fig. 2). The internal state of the postsynaptic neuron is obtained by the algebraic summation of the internal states in the first two cases and modified during the repolarization and hyperpolarization phases. The three processes of summation, repolarization, and hyperpolarization lead to the postsynaptic neuron firing output spikes at times different than those for the first two cases. In Fig. 1, the first spike in the third case occurs earlier than the first spike in the first case because the postsynaptic neuron in the third case exceeds the threshold value earlier. The three cases shown in Fig. 1 highlight the importance of the timing of spikes in the presynaptic spike train for encoding information.

4 Models of Spiking Neurons

Spiking neurons can be modeled in many different ways. Many detailed mathematical models have been developed to quantitatively characterize neuronal behavior based on detailed modeling of the neuronal membrane potential and ion channel conductances [42,43,44,45,46]. Networks of such neuronal models have proved to be very valuable in studying the behavior of biological neural networks, neuronal learning mechanisms such as long-term potentiation and depotentiation, and neurotransmitter-based signaling [47]. However, the level of detail, although ideal for reproducing electrophysiological responses accurately, increases the complexity of the model making them difficult to analyze [48,49]. This complexity also imposes a significant computational burden for large neural network based classification or pattern recognition tasks that employ BP as the learning mechanism.

Another obstacle to the use of these detailed models in feedforward ANNs is imposed by the dynamics of the BP algorithm which usually requires a single activation function (representing changes in membrane potential) for backpropagating the error term through the neuron. The detailed models are usually based on multiple differential equations that capture the behavior of different ion channels and currents that affect the membrane potential. It remains to be seen if error backpropagation is even mathematically possible in the face of such complexity.

Spike response models are phenomenological models that are simpler than the detailed models and offer a compromise between computational burden and electrophysiological detail [50,43,51,52,53,54]. As a result, spike response models are preferred for systemic studies of memory, neural coding, and network dynamics. Bohte et al. [55] employed such a spike response model (originally presented by Gerstner [51]) to demonstrate that BP-based learning is possible in such a network. Other spike response models may also be adapted provided that their activation function can be adapted for error backpropagation. In principle, detailed biophysical models and more complicated phenomenological models appear to be better suited to SNNs that are similar in concept to Radial Basis Function Neural Networks and Self Organizing Maps and are not restricted by the requirements and computational burden of BP-based learning.

5 Spiking Neural Networks (SNNs)

SNNs are, simply, networks of spiking neurons. The SNN architecture, as shown in Fig. 3(a) is similar to that of a traditional feedforward ANN. The network is assumed to be fully connected i.e. a neuron in any layer is connected to all neurons in the preceding layer. However, unlike feedforward ANNs where two neurons are connected by one synapse only, the connection between two SNN neurons is modeled by multiple (K) synapses as shown in Fig. 3(b) [56,55]. The number K is constant for any two neurons and each synapse has a weight and a delay associated with it.

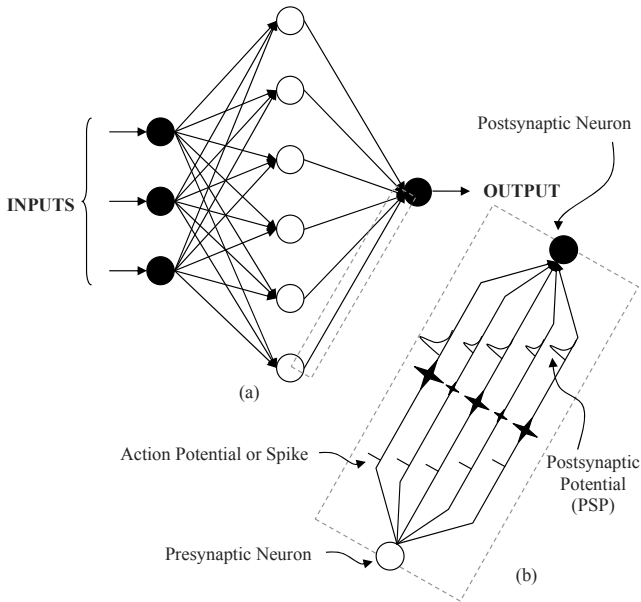


Fig. 3. (a) Spiking neural network architecture; (b) multiple synapses connecting a presynaptic neuron to a postsynaptic neuron

Assuming that presynaptic neuron fires a spike at time t , the k^{th} synapse transmits that spike to the postsynaptic neuron at time $t + d_k$ where d_k is the delay associated with the k^{th} synapse. This architecture enables a presynaptic neuron to affect a postsynaptic neuron by inducing PSPs of varying magnitudes at various time instants. The magnified connection in Fig. 3(b) displays the temporal sequence of spikes (short vertical lines) from the presynaptic neuron, the synaptic weights (proportionate to the size of the star shaped units in the center), and the resulting PSPs (proportionate to the size of the waveform). The modeling of synapses is identical for all neurons, and the k^{th} synapse between any two neurons has the same delay d_k . For the sake of simplicity, neurons in Bohte et al.'s model were restricted to the emission of a single spike. Recently, networks based on spiking neurons that convey information via spike trains (multiple spikes) have also been presented [57,58].

Similar to traditional ANNs, SNN architecture consists of an input layer, a hidden layer, and an output layer (Fig. 3). The number of neurons in the hidden layer is usually selected by trial and error. Since the SNN model is based on spike times, inputs to the SNN have to be preprocessed to convert the continuous real-valued input *features* (or classification variables) into discrete spike times. As a result, the number of original features is converted into a new number of features for input to the SNN. This is known as *input encoding*. Similarly, the number of neurons in the output layer depends on the *output encoding* scheme selected for the classification problem. In SNNs the inputs and outputs can be encoded in a variety of ways. This variety, however, is limited by the assumption of only one spike per neuron.

Until recently, the lack of a continuous and differentiable activation function relating the internal state of the neuron to the output spike times made spiking neurons incompatible with the error backpropagation required for supervised learning. Bohte et al. [55] presented a BP learning algorithm for SNN, dubbed *SpikeProp* similar in concept to the BP algorithm developed for traditional neural networks [31]. Subsequently, SNN was used with various training algorithms such as backpropagation with momentum [59,60], *QuickProp* [59,60], resilient propagation (*RProp*) [60], and Levenberg-Marquardt BP [61] to improve network training performance [11]. QuickProp is a faster converging variant of the original BP learning rule [31] that searches for the global error minimum by approximating the error surface on the basis of local changes in the gradient and weights [62]. RProp is also a fast variant of the BP algorithm where the weights are adjusted based on the direction of the gradient rather than the magnitude. This strategy is specially effective or *resilient* when the error surface is highly uneven and the gradient is not an accurate predictor of the learning rate [63]. Compared with SpikeProp, the aforementioned improved algorithms reportedly provide faster convergence by approximately 600% [11]. Some preliminary research has also been reported regarding the adjustment of other SNN parameters such as neuron threshold, synaptic delays, and the time decay constant defining the shape of the PSP [64]. Recently, new learning algorithms have also been presented for training SNN models that convey information in the form of spike trains [57,58] instead of single spikes.

Computationally, SNN training is usually at least two orders of magnitude more intensive than the traditional ANNs for two reasons [11]. First, multiple weights have to be computed for multiple synapses connecting a presynaptic neuron to a postsynaptic neuron. Second, the internal state of each neuron has to be computed for a continuous duration of time, called *simulation time* (see Fig. 1), to obtain the output spiking times. The time resolution, called *time step*, employed for this computation along with the simulation time and the number of convergence epochs are key factors that affect the actual computation time (real-time) required to train the network. Another difficulty with SNN training is the highly uneven nature of the error surface that can wreak havoc with the gradient descent-based training algorithms. Slight changes in the synaptic weights result in proportionate changes in the postsynaptic potential. But slight

changes in the postsynaptic potential may result in disproportionate changes in the output spike times of the postsynaptic neuron. To overcome this training difficulty various heuristic rules are used to limit the changes of the synaptic weights [65,57,11,58].

6 Concluding Remarks

SNNs have been used for complicated time-dependent pattern recognition problems defined by time series because of their inherent dynamic representation. Further, SNNs have been shown theoretically to have the ability to approximate any continuous function [66]. Addition of the temporal dimension for information encoding has the potential to result in compact representations of large neural networks, another advantage for SNNs. However, their widespread acceptance and application is currently limited by the excessive computing times required for training [11]. It may be expected that this will change in the near future for two reasons. First, technology is advancing at a rapid rate and the computational limitations outlined in this manuscript may not remain as limiting. Second, the field of SNNs is of great research interest and developing rapidly as well.

From the perspective of SNN development, in the opinion of the authors, an adaptive adjustment of the number of synapses [64] needs to be investigated with the goal of reducing the number of weights and consequently computational effort, without compromising the classification accuracy. An additional source of computational effort is the input encoding that increases the number of features many times. New methods of input encoding that do not increase the number of features should be explored. Currently, there is great interest in the development of efficient and accurate learning algorithms for feedforward as well as other networks. Novel combinations of these strategies along with improved understanding of biological information processing will contribute significantly to the development of SNNs as the next generation neural networks.

References

1. Adeli, H., Hung, S.L.: Machine Learning - Neural Networks, Genetic Algorithms, and Fuzzy Sets. John Wiley and Sons, New York (1995)
2. Adeli, H., Park, H.S.: Neurocomputing for Design Automation. CRC Press, Boca Raton (1998)
3. Adeli, H., Karim, A.: Fuzzy-wavelet RBFNN model for freeway incident detection. *Journal of Transportation Engineering* 126(6), 464–471 (2000)
4. Adeli, H.: Neural networks in civil engineering: 1989-2000. *Computer-Aided Civil and Infrastructure Engineering* 16(2), 126–142 (2001)
5. Ghosh-Dastidar, S., Adeli, H.: Wavelet-clustering-neural network model for freeway incident detection. *Computer-Aided Civil and Infrastructure Engineering* 18(5), 325–338 (2003)
6. Adeli, H., Karim, A.: Wavelets in Intelligent Transportation Systems. John Wiley and Sons, Hoboken (2005)
7. Adeli, H., Ghosh-Dastidar, S., Dadmehr, N.: Alzheimer's disease and models of computation: Imaging, classification, and neural models. *Journal of Alzheimer's Disease* 7(3), 187–199 (2005a)

8. Adeli, H., Ghosh-Dastidar, S., Dadmehr, N.: Alzheimer's disease: Models of computation and analysis of EEGs. *Clinical EEG and Neuroscience* 36(3), 131–140 (2005b)
9. Adeli, H., Jiang, X.: Dynamic fuzzy wavelet neural network model for structural system identification. *Journal of Structural Engineering* 132(1), 102–111 (2006)
10. Ghosh-Dastidar, S., Adeli, H.: Neural network-wavelet microsimulation model for delay and queue length estimation at freeway work zones. *Journal of Transportation Engineering* 132(4), 331–341 (2006)
11. Ghosh-Dastidar, S., Adeli, H.: Improved spiking neural networks for EEG classification and epilepsy and seizure detection. *Integrated Computer-Aided Engineering* 14(3), 187–212 (2007)
12. Adeli, H., Hung, S.L.: A concurrent adaptive conjugate gradient learning algorithm on MIMD machines. *Journal of Supercomputer Applications* 7(2), 155–166 (1993)
13. Ghosh-Dastidar, S., Adeli, H., Dadmehr, N.: Mixed-band wavelet-chaos-neural network methodology for epilepsy and epileptic seizure detection. *IEEE Transactions on Biomedical Engineering* 54(9), 1545–1551 (2007)
14. Adeli, H., Park, H.S.: Counter propagation neural network in structural engineering. *Journal of Structural Engineering* 121(8), 1205–1212 (1995a)
15. Panakkat, A., Adeli, H.: Recurrent neural network for approximate earthquake time and location prediction using multiple seismicity indicators. *Computer-Aided Civil and Infrastructure Engineering* 24(4), 280–292 (2009)
16. Karim, A., Adeli, H.: Comparison of the fuzzy-wavelet RBFNN freeway incident detection model with the california algorithm. *Journal of Transportation Engineering* 128(1), 21–30 (2002)
17. Karim, A., Adeli, H.: Radial basis function neural network for work zone capacity and queue estimation. *Journal of Transportation Engineering* 129(5), 494–503 (2003)
18. Ghosh-Dastidar, S., Adeli, H., Dadmehr, N.: Principal component analysis-enhanced cosine radial basis function neural network for robust epilepsy and seizure detection. *IEEE Transactions on Biomedical Engineering* 55(2), 512–518 (2008)
19. Jiang, X., Adeli, H.: Dynamic wavelet neural network model for traffic flow forecasting. *Journal of Transportation Engineering* 131(10), 771–779 (2005)
20. Jiang, X., Adeli, H.: Pseudospectra, MUSIC, and dynamic wavelet neural network for damage detection of highrise buildings. *International Journal for Numerical Methods in Engineering* 71(5), 606–629 (2007)
21. Jiang, X., Adeli, H.: Dynamic fuzzy wavelet neuroemulator for nonlinear control of irregular highrise building structures. *International Journal for Numerical Methods in Engineering* 74(7), 1045–1066 (2008)
22. Adeli, H., Hung, S.L.: An adaptive conjugate gradient learning algorithm for effective training of multilayer neural networks. *Applied Mathematics and Computation* 62(1), 81–102 (1994)
23. Adeli, H., Park, H.S.: Optimization of space structures by neural dynamics. *Neural Networks*, Vol 8(5), 769–781 (1995b)
24. Adeli, H., Karim, A.: Neural dynamics model for optimization of cold-formed steel beams. *Journal of Structural Engineering* 123(11), 1535–1543 (1997)
25. Adeli, H., Jiang, X.: Neuro-fuzzy logic model for freeway work zone capacity estimation. *Journal of Transportation Engineering* 129(5), 484–493 (2003)
26. Sejnowski, T.J.: Open questions about computation in the cerebral cortex. In: Rumelhart, D.E., McClelland, J.L. (eds.) *Parallel Distributed Processing*, vol. 2, pp. 372–389. MIT Press, Cambridge (1986)

27. Maass, W.: Lower bounds for the computational power of spiking neural networks. *Neural Computation* 8(1), 1–40 (1996)
28. Maass, W.: Networks of spiking neurons: The third generation of spiking neural network models. *Neural Networks* 10(9), 1659–1671 (1997a)
29. Bose, N.K., Liang, P.: *Neural Network Fundamentals with Graphs, Algorithms, and Applications*. McGraw-Hill, New York (1996)
30. Maass, W.: Noisy spiking neurons with temporal coding have more computational power than sigmoidal neurons. In: Mozer, M., Jordan, M.I., Petsche, T. (eds.) *Advances in Neural Information Processing Systems*, vol. 9, pp. 211–217. MIT Press, Cambridge (1997b)
31. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In: Rumelhart, D.E., McClelland, J.L. (eds.) *Parallel Distributed Processing*, vol. 1, pp. 318–362. MIT Press, Cambridge (1986)
32. Hung, S.L., Adeli, H.: Parallel backpropagation learning algorithms on Cray Y-MP8/864 supercomputer. *Neurocomputing* 5(6), 287–302 (1993)
33. Park, H.S., Adeli, H.: Distributed neural dynamics algorithms for optimization of large steel structures. *Journal of Structural Engineering* 123(7), 880–888 (1997)
34. Adeli, H., Wu, M.: Regularization neural network for construction cost estimation. *Journal of Construction Engineering and Management* 124(1), 18–24 (1998)
35. Adeli, H., Samant, A.: An adaptive conjugate gradient neural network - wavelet model for traffic incident detection. *Computer-Aided Civil and Infrastructure Engineering* 15(4), 251–260 (2000)
36. Sirca, G., Adeli, H.: Neural network model for uplift load capacity of metal roof panels. *Journal of Structural Engineering* 127(11), 1276–1285 (2001)
37. Sirca, G., Adeli, H.: Neural network model for uplift load capacity of metal roof panels - closure. *Journal of Structural Engineering* 129(4), 562–563 (2003)
38. Dharia, A., Adeli, H.: Neural network model for rapid forecasting of freeway link travel time. *Engineering Applications of Artificial Intelligence* 16(7-8), 607–613 (2003)
39. Panakkat, A., Adeli, H.: Neural network models for earthquake magnitude prediction using multiple seismicity indicators. *International Journal of Neural Systems* 17(1), 13–33 (2007)
40. Jiang, X., Adeli, H.: Neuro-genetic algorithm for nonlinear active control of highrise buildings. *International Journal for Numerical Methods in Engineering* 75(8), 770–786 (2008)
41. Kandel, E.R., Schwartz, J.H., Jessell, T.M.: *Principles of Neural Science*, 4th edn. McGraw-Hill, New York (2000)
42. Hodgkin, A.L., Huxley, A.F.: A quantitative description of ion currents and its applications to conduction and excitation in nerve membranes. *Journal of Physiology* 117, 500–544 (1952)
43. Rinzel, J., Ermentrout, G.B.: Analysis of neuronal excitability and oscillations. In: Koch, C., Segev, I. (eds.) *Methods in Neuronal Modeling*, pp. 135–169. MIT Press, Cambridge (1989)
44. Hille, B.: *Ionic channels of excitable membranes*, 2nd edn. Sinauer Associates, Sunderland (1992)
45. Ermentrout, G.B.: Type I membranes, phase resetting curves, and synchrony. *Neural Computation* 8, 979–1001 (1996)
46. Hoppensteadt, F.C., Izhikevich, E.M.: *Weakly Connected Neural Networks*. Springer, New York (1997)
47. Izhikevich, E.M.: Solving the distal reward problem through linkage of STDP and dopamine signaling. *Cerebral Cortex* 17, 2443–2452 (2007)

48. Abbott, L.F., Kepler, T.B.: Model neurons: From Hodgkin-Huxley to Hopfield. In: Garrido, L. (ed.) *Statistical Mechanics of Neural Networks*. Springer, Berlin (1990)
49. Kepler, T.B., Abbott, L.F., Marder, E.: Reduction of conductance-based neuron models. *Biological Cybernetics* 66, 381–387 (1992)
50. Ermentrout, G.B., Kopell, N.: Parabolic bursting in an excitable system coupled with a slow oscillation. *SIAM Journal on Applied Mathematics* 46, 233–253 (1986)
51. Gerstner, W.: Time structure of the activity in neural network models. *Physical Review E* 51, 738–758 (1995)
52. Kistler, W.M., Gerstner, W., van Hemmen, J.L.: Reduction of Hodgkin-Huxley equations to a single-variable threshold model. *Neural Computation* 9, 1015–1045 (1997)
53. Izhikevich, E.M.: Resonate-and-fire neurons. *Neural Networks* 14, 883–894 (2001)
54. Gerstner, W., Kistler, W.M.: *Spiking Neuron Models. Single Neurons, Populations, Plasticity*. Cambridge University Press, New York (2002)
55. Bohte, S.M., Kok, J.N., La Poutré, J.A.: Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing* 48(1-4), 17–37 (2002)
56. Natschläger, T., Ruf, B.: Spatial and temporal pattern analysis via spiking neurons. *Network: Computation in Neural Systems* 9(3), 319–332 (1998)
57. Booi, O., Nguyen, H.T.: A gradient descent rule for multiple spiking neurons emitting multiple spikes. *Information Processing Letters* 95(6), 552–558 (2005)
58. Ghosh-Dastidar, S., Adeli, H.: A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection. *Neural Networks* 22 (in press, 2009)
59. Xin, J., Embrechts, M.J.: Supervised learning with spiking neural networks. In: *Proceedings of the International Joint Conference on Neural Networks*, Washington, DC, vol. 3, pp. 1772–1777 (2001)
60. McKennoch, S., Liu, D., Bushnell, L.G.: Fast modifications of the SpikeProp algorithm. In: *Proceedings of the International Joint Conference on Neural Networks*, Vancouver, Canada, pp. 3970–3977 (2006)
61. Silva, S.M., Ruano, A.E.: Application of Levenberg-Marquardt method to the training of spiking neural networks. In: *Proceedings of the International Conference on Neural Networks and Brain*, vol. 3, pp. 1354–1358 (2005)
62. Fahlman, S.E.: Faster-learning variations of back-propagation: An empirical study. In: *Proceedings of the 1988 Connectionist Models Summer School*, San Mateo, CA, pp. 38–51. Morgan Kaufmann, San Francisco (1988)
63. Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: The Rprop algorithm. In: *IEEE International Conference on Neural Networks*, San Francisco, CA, vol. 1, pp. 586–591 (1993)
64. Schrauwen, B., van Campenhout, J.: Extending SpikeProp. In: *Proceedings of the International Joint Conference on Neural Networks*, Budapest, pp. 471–476 (2004)
65. Moore, S.C.: *Backpropagation in spiking neural networks*. Master's thesis, University of Bath (2002)
66. Maass, W.: Fast sigmoidal networks via spiking neurons. *Neural Computation* 9(2), 279–304 (1997c)

Choquet Fuzzy Integral Applied to Stereovision Matching for Fish-Eye Lenses in Forest Analysis

P. Javier Herrera¹, Gonzalo Pajares¹, María Guijarro², José J. Ruz³,
and Jesús M. de la Cruz³

¹ Dpto. Ingeniería del Software e Inteligencia Artificial, Facultad de Informática, Universidad Complutense, 28040 Madrid, Spain

pjherrera@pdi.ucm.es, pajares@fdi.ucm.es

² Centro Superior de Estudios Felipe II, Ingeniería Técnica en informática de Sistemas 28300 Aranjuez, Madrid, Spain

mgujarro@cesfelipesecondo.com

³ Dpto. Arquitectura Computadores y Automática, Facultad de Informática, Universidad Complutense, 28040 Madrid, Spain

{jjru, jmcruz}@dacya.ucm.es

Abstract. This paper describes a novel stereovision matching approach based on omni-directional images obtained with fish-eye lenses in forest environments. The goal is to obtain a disparity map as a previous step for determining the volume of wood in the imaged area. The interest is focused on the trunks of the trees, due to the irregular distribution of the trunks; the most suitable features are the pixels. A set of six attributes is used for establishing the matching between the pixels in both images of the stereo pair. The final decision about the matched pixel is taken based on the Choquet Fuzzy Integral paradigm, which is a technique well tested for combining classifiers. The use and adjusting of this decision approach to our specific stereo vision matching problem makes the main finding of the paper. The procedure is based on the application of three well known matching constraints. The proposed approach is compared favourably against the usage of simple features and other fuzzy strategy that combines the simple ones.

Keywords: Choquet Fuzzy Integral, Fish-eye stereo vision, Stereovision matching, omni-directional forest images.

1 Introduction

One important task in forests maintenance is to determine the volume of wood in an area for different purposes, including the control of growth of the trees. This task can be carried out by stereovision systems. Fish-eye lenses allow imaging a large sector of the surrounding space with omni-directional vision. This justifies its use.

According to [1] we can view the classical problem of stereo analysis as consisting of the following steps: image acquisition, camera modelling, feature acquisition, image matching, depth determination and interpolation. The key step is that of image matching. This is the process of identifying the corresponding points in two images

that are cast by the same physical point in the 3-D space. This paper is devoted solely to the matching one. Two sorts of techniques have been used for matching: area-based and feature based [2].

Area-based stereo techniques [3] use correlation between brightness (intensities) patterns in the local neighbourhood of a pixel in one image with brightness patterns in the local neighbourhood of the other image. Also statistical textures can be considered under this category. Feature-based methods [4] use set of pixels with similar attributes, colour, gradient (module and direction) or Laplacian. These are the six attributes available to be used in our matching procedure.

Figure 1(a) displays one omni-directional image (let's say the left one) of the stereo pair captured with a fisheye lens. Figure 1(b) displays the signed and expanded area on Figure 1(a). In Figure 1(c) the corresponding area in the right image of the stereo pair is displayed. Due to the different locations of the tree's crowns there exists an important lighting variability between both areas; this makes the matching process a difficult task. This is applicable for the whole image.

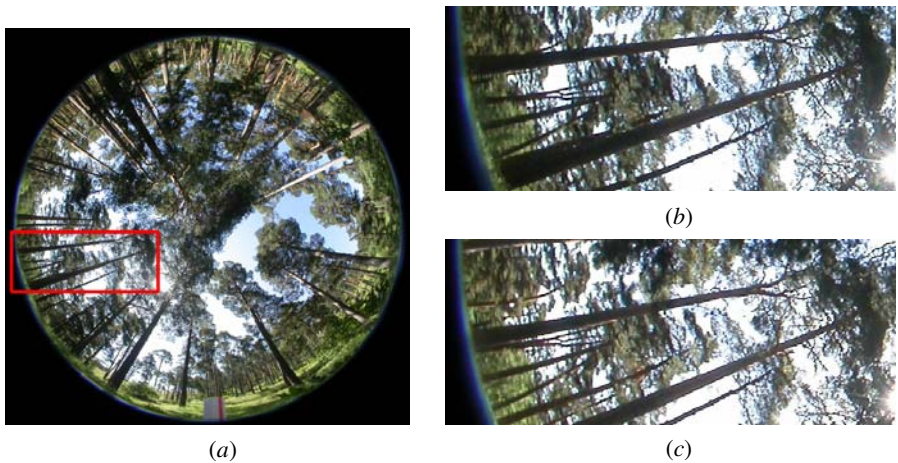


Fig. 1. (a) Omni-directional left image; (b) left expanded area; (c) corresponding right expanded area

The following three stereovision constraints can be applied for solving the matching problem. *Epipolar*: derived from the system geometry, given a pixel in one image its correspondence will be on the epipolar line. *Similarity*: matched pixels display similar attributes. *Uniqueness*: a pixel in the left image must be matched to a unique pixel in the right one.

Given a pixel in the left image, we apply the epipolar constraint for determining a list of candidates, which are potential matches, in the right image. Each candidate becomes an alternative for the first pixel. We also apply the similarity constraint based on the six attributes, obtaining six similarity measures, which are conveniently combined. The final decision about the correct match, among the list of candidates, is made according to the support that each candidate receives by applying the Choquet Fuzzy Integral (CFI) paradigm. This unique selection implies the application of the uniqueness constraint. The matching through the CFI makes the main contribution of

the paper. The proposed approach is compared favourably against the usage of individual area-based and feature-based correspondence techniques.

This work is organized as follows. Section 2 describes the design of the matching process; including a brief overview of the CFI paradigm. Section 3 describes the results obtained by using the combined CFI approach, and comparing these results with those obtained by applying each individual strategy. Section 4 presents the conclusions and future work.

2 Design of the Matching Process

2.1 Epipolar: System Geometry

Figure 2 displays the stereo vision system geometry [5]. The 3D object point P with world coordinates with respect to the systems (X_1, Y_1, Z_1) and (X_2, Y_2, Z_2) is imaged as (x_{i1}, y_{i1}) and (x_{i2}, y_{i2}) in image-1 and image-2 respectively in coordinates of the image system; α_1 and α_2 are the angles of incidence of the rays from P ; y_{12} is the baseline measuring the distance between the optical axes in both cameras along the y -axes; r is the distance between image point and optical axis; R is the image radius, identical in both images.

According to [6], the following geometrical relations can be established,

$$r = \sqrt{x_{i1}^2 + y_{i1}^2}; \quad \alpha_1 = (r90^\circ)/R; \quad \beta = tg^{-1}(y_{i1}/x_{i1}) \tag{1}$$

Now the problem is that the 3D world coordinates (X_1, Y_1, Z_1) are unknown. They can be estimated by varying the distance d as follows,

$$X_1 = d \cos \beta; \quad Y_1 = d \sin \beta; \quad Z_1 = \sqrt{X_1^2 + Y_1^2} / \tan \alpha_1 \tag{2}$$

From (2) we transform the world coordinates in the system $O_1X_1Y_1Z_1$ to the world coordinates in the system $O_2X_2Y_2Z_2$ taking into account the baseline as follows:

$$X_2 = X_1; \quad Y_2 = Y_1 + y_{12}; \quad Z_2 = Z_1 \tag{3}$$

Assuming no lenses radial distortion, we can find the imaged coordinates of the 3D point in image-2 as [6],

$$x_{i2} = \frac{2R \arctan\left(\sqrt{X_2^2 + Y_2^2} / Z_2\right)}{\pi \sqrt{(Y_2/X_2)^2 + 1}}; \quad y_{i2} = \frac{2R \arctan\left(\sqrt{X_2^2 + Y_2^2} / Z_2\right)}{\pi \sqrt{(X_2/Y_2)^2 + 1}} \tag{4}$$

Using only a camera, we capture a unique image and the 3D points belonging to the line $\overline{O_1P}$, are all imaged in the unique point represented as (x_{i1}, y_{i1}) . So, the 3D coordinates with a unique camera cannot be obtained. When we try to match the imaged point (x_{i1}, y_{i1}) into the image-2 we follow the epipolar line, i.e. the projection of $\overline{O_1P}$ over the image-2. This is equivalent to vary the parameter d in the 3-D space.

So, given the imaged point (x_{i1}, y_{i1}) in the image-1 (left) and following the epipolar line, we obtain a list of m potential corresponding candidates represented by (x_{i2}, y_{i2}) in the image-2 (right).

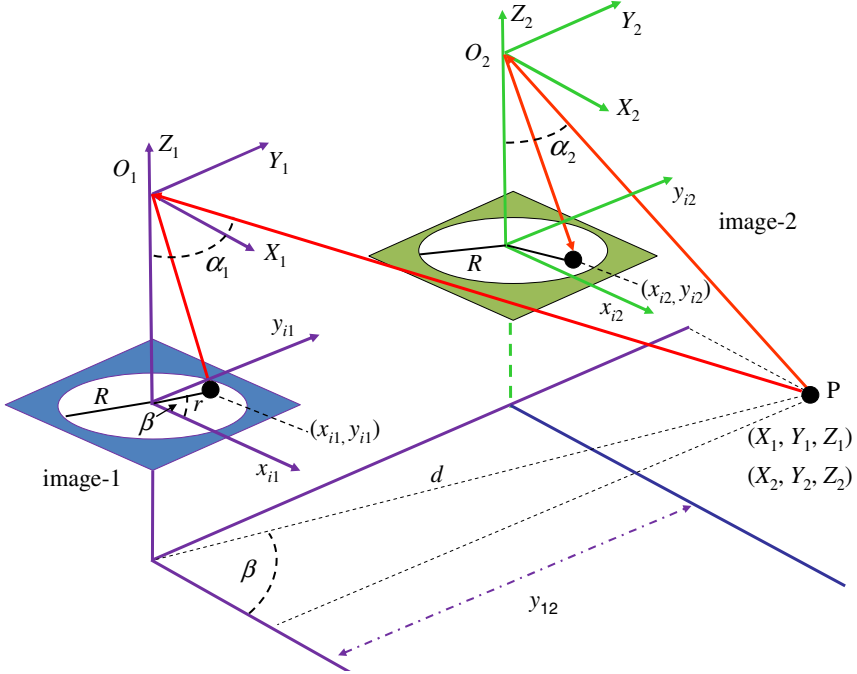


Fig. 2. Geometric projections and relations for the fish-eye based stereo vision system

2.2 Similarity: Attributes for Area and Feature-Based

Each pixel l in the left image is characterized by its attributes; one of such attributes is denoted as A_l . In the same way, each candidate i in the list of m candidates is described by identical attributes, A_i . So, we can compute differences between attributes of the same type A , obtaining a similarity measure for each attribute as,

$$s_{iA} = (1 + |A_l - A_i|)^{-1}; \quad i = 1, \dots, m \quad (5)$$

$s_{iA} \in [0, 1]$, $s_{iA} = 0$ if the difference between attributes is large enough (minimum similarity), otherwise if they are equal ($s_{iA} = 1$, maximum similarity).

As mentioned before, in this paper we use the following six attributes for describing each pixel (feature): *a*) correlation; *b*) texture; *c*) colour; *d*) gradient magnitude; *e*) gradient direction and *f*) Laplacian. Both first ones are area-based computed on a 3×3 neighbourhood around each pixel through the correlation coefficient [7] and standard deviation [8]. The four remaining ones are considered as feature-based [4]. The colour involves the three red-green-blue spectral components

(R,G,B) and the absolute value in the equation (5) is extended as:
 $|A_l - A_i| = \sum_H |H_l - H_i|, H = R,G,B.$

Gradient (magnitude and direction) and Laplacian are computed by applying the first and second derivatives [8], over the intensity image after its transformation from the RGB plane to the HSI (hue, saturation, intensity) one. Given a pixel in the left image and the set of m candidates in the right one, we compute the following similarity measures for each attribute A : s_{ia} (correlation), s_{ib} (colour), s_{ic} (texture), s_{id} (gradient magnitude), s_{ie} (gradient direction) and s_{if} (Laplacian). The identifiers in the sub indices identify the attributes according to the above assignments.

2.3 Uniqueness: Applying the Choquet Fuzzy Integral Paradigm

Now we must match each pixel l in the left image with the best of the potential candidates (uniqueness). This is a decision based on the CFI paradigm. This paradigm allows combining the individual similarities, which are computed through the equation (5).

The CFI requires the computation of the relevance assigned for each attribute from which we can compute the so-called fuzzy densities. This is solved by computing the λ – fuzzy measure using the data [9]. The calculation starts with selecting a set of six fuzzy values, $g^a, g^b, g^c, g^d, g^e, g^f$, each one representing the individual relevance of the associated attribute. The attributes are the six described above, i.e. $\Omega \equiv \{a,b,c,d,e,f\}$ associated to correlation, texture, color, gradient magnitude, gradient direction and Laplacian.

The value of λ needed for calculating the fuzzy densities g is obtained as the unique real root greater than -1 of the polynomial,

$$\lambda + 1 = \prod_{j \in \Omega} (1 + \lambda g^j) \tag{6}$$

The individual relevancies for each attribute are computed from the data, as described later in the section 3a.

Once the g^a, \dots, g^f are obtained and λ is found, the fuzzy integral works as follows:

1. For a given pixel l in the left image, we compute the similarities through the equation (5) between l and every candidate i , with $i = 1, \dots, m$, obtaining a column vector as: $[s_{ia}, s_{ib}, s_{ic}, s_{id}, s_{ie}, s_{if}]^T$; without loss of generality assume that s_{ia} is the highest similarity value and s_{if} the lowest. This vector is arranged under this criterion, i.e. $s_{ia} > s_{ib} > s_{ic} > s_{id} > s_{ie} > s_{if}$.
2. Arrange the above fuzzy values correspondingly with the mentioned arrangement, i.e. $g^a, g^b, g^c, g^d, g^e, g^f$ and set the first fuzzy density $g(a) = g^a$.
3. Compute the remaining fuzzy densities according to the recursive procedure given in the equation (7).

$$\begin{aligned} g(b) &= g^b + g(a) + \lambda g^b g(a) \\ g(c) &= g^c + g(b) + \lambda g^c g(b) \\ &\dots\dots\dots \\ g(f) &= g^f + g(e) + \lambda g^f g(e) \end{aligned} \tag{7}$$

4. Calculate for each candidate i , the support received to be matched with l as,

$$\mu_i(l) = s_{ia} + \sum_{h=b}^f [s_{i(h-1)} - s_{ih}]g(h-1) \quad (8)$$

5. The decision about the best match is made by selecting the maximum support $\mu_i(l)$ among all candidates.

3 Results

The system is based on the scheme of the figure 2, with a baseline of 1 meter. The cameras are equipped each one with Nikon FC-E8 fisheye lens, with an angle of 183°. The valid colour images in the circle contain 6586205 pixels.

The tests have been carried out with twelve pairs of stereo images. We use two of them for computing the relevance of each attribute, from which the fuzzy densities can be obtained. At a second stage, we apply the CFI approach pixel by pixel for the remainder ten stereo pairs.

Our interest consists of determining the disparity of the trees trunks located in an area of 25 m² around the stereo vision system.

The disparity is the absolute difference value in sexagesimal degrees, taking into account the imaged circle, between the pixel in the left image and its matched pixel in the right one.

We have available the information of disparities provided by the end users. Thus, for each pixel in a trunk we know its correct disparity value according to this expert knowledge; which allows us to compute the percentage of error. For each one of the ten stereo images used for testing, we compute the disparity error for the trunks and then average these errors among the ten pairs of stereo images.

a) Computing the Relevance for Each Criterion

Given both available stereo images for this purpose, for each pixel in the left images, we compute the disparity with respect its matched pixel in the right ones, but considering each one of the six attributes separately through the equation (5). Each match is established according to the maximum similarity value computed for each attribute individually. But this does not imply that maximum similarity corresponds to a true match. Therefore, we need a mechanism for computing the relevance based on the rate of success or failure of each attribute. So, we compute the averaged percentage of error for both stereo images and for each attribute, based on the expert knowledge available about the disparities in the trunks. These probabilities are: $p_a = 28$ (correlation), $p_b = 10$ (colour), $p_c = 14$ (texture), $p_d = 9$ (gradient magnitude), $p_e = 30$ (gradient direction) and $p_f = 27$ (Laplacian). So, the individual relevancies are computed as $g^h = p_h / \sum_k p_k$, $h, k = a, b, c, d, e, f$. Finally, these fuzzy values are exactly the following: $g^a = 0.150$, $g^b = 0.179$, $g^c = 0.187$, $g^d = 0.189$, $g^e = 0.145$ and $g^f = 0.152$. As one can see, the most relevant attribute is the gradient magnitude.

b) CFI Performance

As before, for each pixel in each one of the ten stereo images, available for testing, we obtain its disparity considering the six attributes separately by applying the equation (5) and a maximum similarity criterion among the m candidates and also by applying the IF approach based on maximum supports, equation (8).

Figures 3(a) and 3(b) are the same that Figures 1(a) and 1(b) respectively. Figure 3(c) displays the disparity map obtained by the CFI approach in the area. The colour bar shows the disparity level values according to the colour.

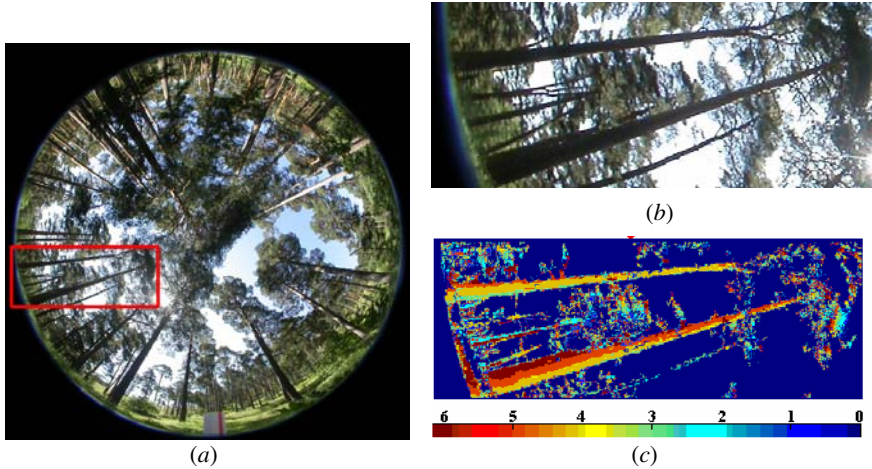


Fig. 3. (a) Left image; (b) expanded area; (c) disparity map obtained by the CFI approach

Table 1 displays the averaged percentage of errors and standard deviations based on the similarity for the six attributes when used separately, identified under the follows columns: $(s_a, s_b, s_c, s_d, s_e, s_f)$. The averaged percentage of error obtained with the CFI approach is also displayed. For comparative purposes we have tested the performance of our approach against the decision making approach proposed by Yager [10] based on fuzzy sets aggregation. The combination is made two to two similarity measures according to the following expression,

$$\mu_i(I) = 1 - \min \left\{ 1, \left((1 - s_{ih})^p + (1 - s_{ik})^p \right)^{\frac{1}{p}} \right\} \quad p \geq 1 \tag{9}$$

where h and k denote two similarity measures. Then, by applying the associative property of this aggregation operator we compute a final support for the six similarity values. The parameter p is estimated from the two stereo pairs used for computing the relevances of each attribute. Indeed, we vary p from 1 to 4, which is the range generally used, and compute the percentage of error, obtaining the best results with p set to 2.0.

Table 1. Averaged percentage of errors and standard deviations obtained through maximum similarity criteria for each attribute separately and the CFI decision making approach

Averaged percentage of error and standard deviations															
s_a		s_b		s_c		s_d		s_e		s_f		YAG		CFI	
%	σ	%	σ	%	σ	%	σ	%	σ	%	σ	%	σ	%	σ
30	2.9	16	1.3	18	1.7	14	1.1	35	3.6	32	3.1	13	1.9	11	1.3

From results in table 1, one can see that both combined strategies, Yager and CFI outperforms the individual similarity based approaches. This means that the combination of similarities between attributes improve the results obtained by using similarities separately. The CFI approach obtains better results than the Yager’s one.

The best individual similarity results are obtained through the similarities provided by the gradient magnitude attribute (s_d). This implies that it is the most relevant attribute. This agrees with its relevance obtained previously, as it has turned out to be the most relevant attribute.

4 Concluding Remarks

In this paper we have proposed a method for stereovision matching, in omnidirectional images, in a system equipped with fish-eye lenses. The method applies three well-known constraints (*epipolar*, *similarity* and *uniqueness*) by combining area-based and feature-based matching strategies, which are classical constraints used in conventional stereovision systems.

For each pixel in the left image, a list of possible candidates in the right image is obtained for determining its correspondence.

The similarity between attributes establishes measures for the matching between the pixel and its candidates. Each candidate receives a support, which establishes the degree of similarity, consequently of correspondence between it and the pixel in the left image.

Under the CFI paradigm, we combine the similarities between six attributes and make a decision for choosing the unique candidate, if any, for the given pixel in the left image. The proposed combined strategy outperforms the methods that use similarities separately and it is compared favorably.

Although the results achieved can be considered satisfactory, they could be improved by applying additional constraints such as *smoothness* or *ordering*, which have been used for matching in conventional stereovision systems.

Another issue still open in future works is that concerning with the correspondence between pixels out of the trunks. This will allows discarding all pixels belonging to the background and also those pixels which belong to the leaves or the sky.

Moreover, the disparity map can be still refined by applying smoothing techniques, perhaps optimization ones, such as simulated annealing or Hopfield neural networks, where unmatched pixels in the trunks can be removed for obtaining a disparity map without gaps or another undesired artifacts.

Acknowledgements

The authors wish to acknowledge to the Council of Education of the Autonomous Community of Madrid and the Social European Fund for the research contract with the first author. Also to Dra. Isabel Cañellas and Fernando Montes from the Forest Research Centre (CIFOR, INIA) for his support and the material supplied.

References

1. Barnard, S., Fishler, M.: Computational Stereo. *ACM Computing Surveys* 14, 553–572 (1982)
2. Cochran, S.D., Medioni, G.: 3-D Surface Description from binocular stereo. *IEEE Trans. Pattern Anal. Machine Intell.* 14(10), 981–994 (1992)
3. Tang, L., Wu, C., Chen, Z.: Image dense matching based on region growth with adaptive window. *Pattern Recognit. Letters* 23, 1169–1178 (2002)
4. Lew, M.S., Huang, T.S., Wong, K.: Learning and feature selection in stereo matching. *IEEE Trans. Pattern Anal. Machine Intell.* 16, 869–881 (1994)
5. Abraham, S., Förstner, W.: Fish-eye-stereo calibration and epipolar rectification. *Photogrammetry and Remote Sensing* 59, 278–288 (2005)
6. Schwalbe, E.: Geometric Modelling and Calibration of Fisheye Lens Camera Systems. In: *Proc. 2nd Panoramic Photogrammetry Workshop, Int. Archives of Photogrammetry and Remote Sensing*, vol. 36, Part 5/W8 (2005)
7. Barnea, D.I., Silverman, H.F.: A Class of Algorithms for Fast Digital Image Registration. *IEEE Trans. Computers* 21, 179–186 (1972)
8. Pajares, G., de la Cruz, J.M.: *Visión por Computador: Imágenes digitales y aplicaciones*, RA-MA (2008)
9. Kuncheva, L.: *Combining Pattern Classifiers: Methods and Algorithms*. Wiley, Chichester (2004)
10. Yager, R.R.: On ordered weighted averaging aggregation operators in multi-criteria decision making. *IEEE Trans. System Man and Cybernetics* 18(1) (1988)

Fuzzy OLAP: A Formal Definition

Claudia González¹, Leonid Tineo¹, and Angélica Urrutia²

¹ Universidad Simón Bolívar
Caracas, Venezuela

claudia@ldc.usb.ve, leonid@usb.ve

² Universidad Católica del Maule
Talca, Chile
aurrutia@ucm.cl

Abstract. Real world is pervaded of imprecision and uncertainty. These characteristics are well represented in computational systems by means of fuzzy logic. Some systems produce vital data that must be stored for its posterior analysis supporting decision making through OLAP. At present time this data may involve imprecision and uncertainty, therefore fuzzy OLAP operators must be provided. We do that in this paper in a formal way, giving a rigorous definition of fuzzy logic extended OLAP operators.

Keywords: OLAP, Fuzzy Cube, Fuzzy Database.

1 Introduction

Classic databases suffer of rigidity [8][11]. Therefore, associated technologies, systems and applications, such as Data Warehouse, suffer of rigidity too. In fact in classic databases, a twofold hypothesis has been maintained: precisely known data is assumed and queries are intended to retrieve elements that qualify for a given Boolean condition. In recent past, Fuzzy Sets [17] application to databases has been proposed, in order to avoid rigidity problems. Several efforts have been done in this sense, such as: OMRON[16], FQUERY[12], ISKREOT[13], FSQL[10][11][3], SQLf [8] and others [9][17][14]. Up to our knowledge, few works has been done in Fuzzy Data Warehouse: a fuzzy OLAP [6] to support qualitative analysis over crisp data, a semantic model [5] with a three layers: quantitative summary, qualitative summary and quantifier summary, and a very initial idea for extension of the cube and its operators in presence of fuzzy data attributes [7]. We focus our attention here in the formal definition of fuzzy Data Warehouse operators OLAP.

2 Motivating Example

In order to illustrate this work practical utility: Suppose a subway company takes a survey, where persons report the time between stations occurred while they used the service. People could have imprecise perception in aspects such as: Waiting Time, Time between Stations, Waiting Area Comfort, Wagon Comfort, Quality of Service (QoS), User Treatment, Frequency of Use, and so on. For example: *I don't exactly how much time it takes, but around of 30 seconds; I guess it was fast; maybe normal.*

Also, the user could not remember exact times (hour:min) in which one has used the subway: *I arrived near to 8:30 am; early in the morning; before 6 pm.* These imprecision in survey must be modeled as fuzzy data.

The representation of this information as fuzzy data allows better processing of this survey’s answers. Instead of be searching for patterns matches, a fuzzy query engines, over the data with fuzzy representation, allows the inference of information which results very useful. For example, if we need to now if the mean time between stations was adequate, we can use a measure already defined as *the fuzzy data mean* and also defining the predicate “adequate” we can’t write a query to ask if *fuzzy mean of time between stations is major or equal than adequate.*

For sake of simplicity and space save, let’s suppose survey result in Table 1, these data might be represented, stored and manipulated in relations with fuzzy attributes as those provided by FSQL[11].

Table 1. Survey Result. Twelve users have given its measures for Time Between Stations and Quality of Service using imprecise data values, also the Hour in which them have used the service is imprecise known.

μ	Hour	QoS	Time Between Stations (in seconds)
0,5	Near to 10 am	Almost 90	Near of 30
0,75	Between 11:30 am and 12:30 pm	60	Less than 60
0,33	After the 8:30 but before 8:45	Near to 20	Almost 25
0,25	After the 12:00 but before 12:15	80	Fast
0,5	Between 4:30 am and 5:00 pm	85	Slow
1	Near to 6:30 pm	Between 65 and 75	Less than 30
0,5	Before 3 pm	50	Fast
0,66	Between 5:45 and 6:00 pm	70	More than 45
0,75	After 8 pm	95	Slow
1	Near to 7 pm	50	Between 30 and 60
0,25	After 11 pm	80	Fast
0,75	Between 6:30 pm and 7:30.	More or less 50	Between 25 and 30

Suppose someone wants to provide a statistic, from the survey, about the time between stations by turn. If we would have a precise time between stations, this statistic could be reduced to a fuzzy query over fuzzy data. However, we also have a fuzzy time between stations. It means that an hour could have a fuzzy membership to a turn shift. This problem can be easily expressed in terms of a data warehouse with a base cube as these presented in Fig. 1. a, in which the time between stations represents the measure of a cube and the dimension time having a hierarchy of hour and turn shift, as it is shown in Fig. 1. The possible values for the Turn Shift could be Morning, Afternoon and Night. In this case, the fuzzy grade is given by the fuzzy intervals as in Fig. 2, which relates an observation in the hour dimension with its corresponding turn shift. It is plausible, because some involuntary delays could occurs and also due to the fact that users do provide inexact information about hours. This case leverage the use of a fuzzy data warehousing and therefore the standard SQL ROLAP could be extended for that.

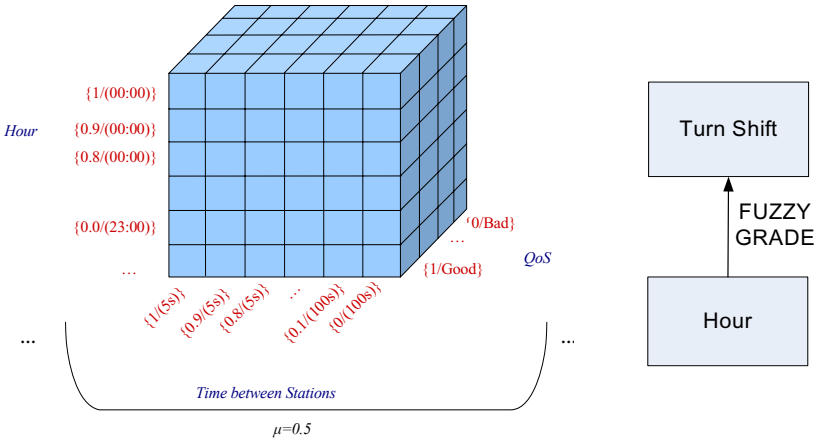


Fig. 1. Left: Fuzzy Cube representation for the survey and Right: Fuzzy Dimension Time

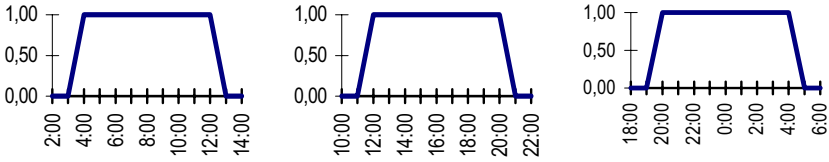


Fig. 2. Turn Shift Soft Representation. Left: Morning, Center: Afternoon, Right: Night.

3 Fuzzy Multidimensional Model

There is a need of adding fuzziness to multidimensional spaces in order to capture more meaning. We take from GEFRED model [11] [15], the definition of generalized fuzzy domain and the possibly equal comparator. This is based in Possibility Theory that allows the representation of imprecise data by means of possibility distributions.

Definition 1. Let's U be a universe, a **possibility distribution** over U , is a function $\pi: U \rightarrow [0,1]$.

Definition 2. Let's U be a universe domain, $P(U)$ the set of all possibility distributions over U , including those that define the types Unknown and Undefined. Let Null be defined as a type. The **Generalized Fuzzy Domain** of U is defined as $FD(U) \subseteq P(U) \cup \text{Null}$. Where Unknown, Undefined and Null definitions are those of Umano, Fukami et al [2][3].

Definition 3. Let's $x, y \in FD(U)$, with possibility distributions π_x and π_y respectively, the **possibly equal comparator** EQ is defined by the truth value $\mu_{EQ}(x, y)$ in (1).

$$\mu_{EQ}(x, y) = \sup_{u \in U} (\min(\pi_x(u), \pi_y(u))) \tag{1}$$

Definition 4. Let's θ a fuzzy comparator on U , defined by $\mu_\theta : U \times U \rightarrow [0,1]$, we define the extension of θ to the fuzzy domain $FD(U)$ by $\mu_\theta(x, y) = \sup_{z \in U} \left(\sup_{w \in U} (\min(\pi_x(z), \pi_y(w), \mu_\theta(z, w))) \right)$.

Definition 5. A fuzzy relation extension of a classic relation $R(A_1, \dots, A_n)$, is defined as a fuzzy set of tuples in $FD(dom(A_1)) \times FD(dom(A_2)) \times \dots \times FD(dom(A_n))$ represented with the schema $R_f(A_{f1}, \dots, A_{fn})$ where each tuple r is provided of a satisfaction degree with membership function $\mu_{R_f}(r) \in [0,1]$ which indicates the membership of a tuple to the fuzzy relation.

In the following, we present our extension for fuzzy multidimensional spaces giving formal definitions of involved concepts.

Definition 6. A dimension D is a lattice $(levels(D), \leq)$ whose Hasse Diagram, named the **hierarchy of the dimension D** is $H(D) = (levels(D), \infty)$. For any $L_1, L_2 \in levels(D)$, $L_1 < L_2$ means that L_1 is of **lower semantic** than L_2 . The lowest bound of the lattice $(levels(D), \leq)$ will be denotes **lowest(D)**. The **space of all dimensions** is denoted by Ω .

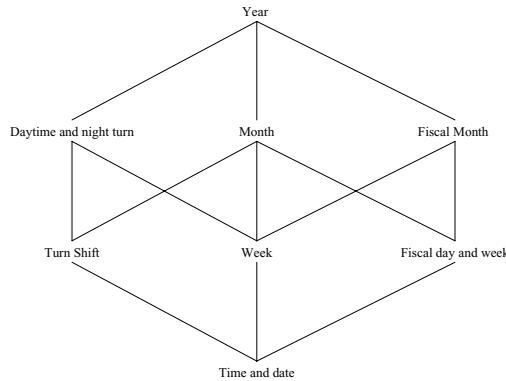


Fig. 3. Hasse Diagram for Dimension Time's Lattice. $H(\text{Time})$.

Definition 7. Let's $D \in \Omega$, a **dimensional path** dp in D is a list of levels ordered by ∞ in $H(D)$ from a minimal to a maximal. We will denote that with the predicate $isPath(dp, H(D))$. We also denote as $paths(D)$ the set of all dimensional paths in D .

Definition 8. Let's $\Psi = \{L \mid \exists_{D \in \Omega} L \in Levels(D)\}$ be the space of all **dimensional levels**. The **dimension of a dimensional level** $L \in \Psi$ is denoted as $h(L)$, $h(L) = D$ iff $L \in levels(D)$.

Definition 9. Let's $L \in \Psi$, the set of values for the level L , named the **associated domain**, is denoted with $dom(L)$, for short we will use just L for $dom(L)$.

A value in the domain of a dimensional level can have ancestors or descendants in other dimensional levels of that dimension. In [4] the relationship ancestor is defined

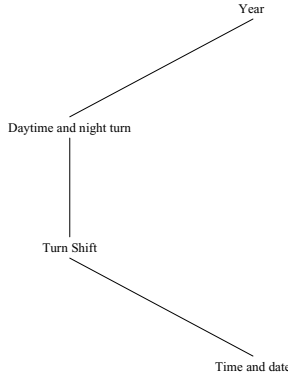


Fig. 4. Dimensional Path Turns in the H(Time)

as a functional dependence between the dimensional levels of a dimension that are in the same dimensional path, for any dimensional path. It is a function returning a perfect known ancestor element, for each element in a level. The dependency function ancestor is obviously crisp; it has not taken in account that data might be imprecise. Therefore we must extend it in order to allow fuzziness representation.

Before, to present the fuzzy definition of ancestors, it is necessary to define a function to associate an element of a dimensional level L1 with its corresponding element in the immediately superior level L2. For each possible value in L2 domain we define the function μ_{l_2} from L1 domain to the [0,1] interval as follows.

Definition 10. Let's $D \in \Omega$, $L_1, L_2 \in Levels(D)$ such that $L_1 \prec L_2$, let $l_2 \in dom(L_2)$, we define the fuzzy set of elements in L_1 represented by l_2 by the membership function $\mu_{l_2} : dom(L_1) \rightarrow [0,1]$.

Here bellow, we present the fuzzy ancestor definition to associate an element of a dimensional level with other in a higher dimensional level, when the relation between them is fuzzy. This relation is recursive and therefore will not be limited to immediately consecutive levels. This definition applies for the time dimension in the study case which dimensional levels are: time and turn shift.

Also, for the reader's sake, we present a symbols table which resumes the definition of the symbols presented in the rest of the section.

Definition 11. Let's $D \in \Omega$, $L_1, L_2 \in Levels(D)$ such that $L_1 \prec L_2$ then the fuzzy predicate ancestor_f is defined as returning a fuzzy value on L_2 with possibility distribution given by:

$$\forall t \in R_f(A_{f1}, \dots, A_{fn}), A_{fx}(t) \in FD(dom(L_1)) (ancestor_f(A_{fx}(t), L_2)) = \left\{ \frac{\pi(y)}{y} \mid y \in dom(L_2) \right\} \quad (2)$$

where

$$\pi(y) = \begin{cases} \sup_{z \in dom(L_1)} \left(\min(\pi_{A_{fx}}(z), \mu_y(z)) \right) & \text{when } L_1 \prec L_2 \\ \sup_{w \in dom(L')} \left(\inf_{z \in dom(L_1)} \left(\pi_{A_{fx}}(z), \mu_w(z), ancestor_f(w, L_2)(y) \right) \right) & \text{when } L_1 \prec L' \prec L_2 \end{cases}$$

Table 2. Used Symbols for the ancestor definition

Symbol	Description
Levels(D)	Levels set of the Dimension D
A _{f_i}	Fuzzy attribute-i
A _{f_i} (t)	Value of fuzzy attribute-i in the row t.
∞	Direct relation between two dimensional levels. L ₁ ∞ L ₂ means that L ₁ is of semantic immediately lower than L ₂ .
<	General relation between two dimensional levels. L ₁ < L ₂ means that L ₁ is of semantic lower than L ₂ .
π	Possibility distribution.
π _{A_{f_i}}	Possibility distribution for the attribute A _{f_i} .
π _{A_{f_i}} (z)	Possibility for the attribute A _{f_i} to take the value z.

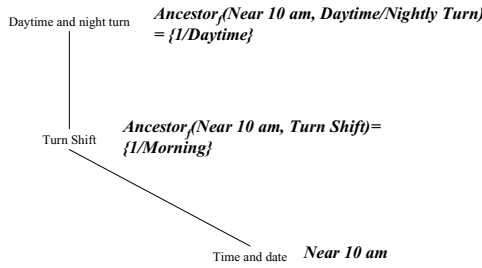


Fig. 5. Ancestors for the value Near 10 am

Definition 12. Let's $D \in \Omega$, $L_1, L_2 \in Levels(D)$ such that $L_1 < L_2$ then the fuzzy predicate descendant_f is defined as returning a fuzzy value on L₁ with possibility distribution given by:

$$\forall t \in R_f(A_{f_1}, \dots, A_{f_n}), A_{f_i}(t) \in FD(dom(L_2)) (descendant_f(A_{f_i}(t), L_1)) = \{\pi(x)_x \mid x \in dom(L_1)\}$$

where

$$\pi(x) = \begin{cases} \sup_{z \in dom(L_2)} (\min(\pi_{A_{f_i}}(z), \mu_z(x))) & \text{when } L_1 \infty L_2 \\ \sup_{w \in dom(L')} \left(\inf_{z \in dom(L_2)} (\pi_{A_{f_i}}(z), \mu_z(w), descendant_f(w, L_1)(y)) \right) & \text{when } L_1 < L' \infty L_2 \end{cases} \quad (3)$$

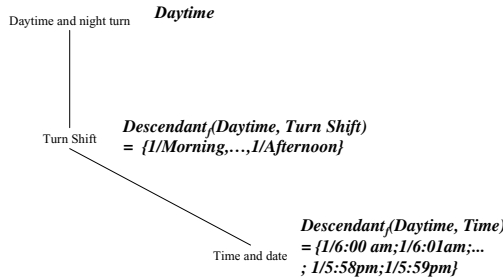


Fig. 6. Descendant for the value Daytime

Hereafter is presented a symbol list that will be used in the section.

Table 3. Used Symbols for the fuzzy cube operators definitions

Symbol	Description
C_f	Fuzzy Cube
Ω_{C_f}	Dimensional Set for the Cube C_f
Di	i-dimension.
M	Dimension of the fuzzy cube measure.
Ψ_{C_f}	Mapping function between the levels and their dimensions.
R_{C_f}	Fuzzy relation

Definition 13. Let's $D \in \Omega$, $L_1, L_2 \in \text{Levels}(D)$, the **dependency degree** $G^{dep}(L_1, L_2)$ is defined as the possibility that each element in L_1 has an ancestor in L_2 by formula given in

$$G^{dep}(L_1, L_2) = \inf_{x \in FD(dom(L_1))} \left(\sup_{y \in L_2} (\text{ancestor}_f(x, L_2)(y)) \right) \quad (4)$$

Definition 14. A **fuzzy cube** is defined as a triplet $C_f = \langle \Omega_{C_f}, \Psi_{C_f}, R_{C_f} \rangle$ where $\Omega_{C_f} \subset \Omega$ is a set of dimensions $\Omega_{C_f} = \{D_1, \dots, D_n, M\}$, being M an special dimension called the **measure of the cube** C_f and denoted as **measureDim**(C_f), $\Psi_{C_f} : \Omega_{C_f} \rightarrow \Psi$ a mapping of levels for dimensions such that $\forall_D (\Psi_{C_f}(D) \in \text{Levels}(D))$ and R_{C_f} is a fuzzy relation on $FD(dom(\Psi_{C_f}(D_1))) \times \dots \times FD(dom(\Psi_{C_f}(D_n))) \times FD(dom(M))$.

Definition 15. A **base fuzzy cube** is a fuzzy cube $C_f = \langle \Omega_{C_f}, \Psi_{C_f}, R_{C_f} \rangle$ such that $\forall_D (\Psi_{C_f}(D) = \text{lowest}(D))$. Each fuzzy cube C_f has an **associated base fuzzy cube** $CB(C_f)$. In particular, if C_f is a base fuzzy cube, its associated base fuzzy cube is itself $CB(C_f) = C_f$.

4 Level Climbing of the Fuzzy Cube

Intuitively, the operation level climbing rolls up a cube in some dimensions to desired levels. This operation is based in the fuzzy ancestor relationship.

Definition 16. Let's $C_f = \langle \Omega_{C_f}, \Psi_{C_f}, R_{C_f} \rangle$, $\Omega' = \{D'_1, \dots, D'_k\} \subseteq \Omega_{C_f} - \{M\}$, $\Psi' : \Omega' \rightarrow \Psi$ a mapping of levels for dimensions such that $\forall_{D'} (\Psi'(D') \in \text{Levels}(D') \wedge \Psi_{C_f}(D') < \Psi'(D'))$ we define the **level climbing of the fuzzy cube** $LC(C_f, \Omega', \Psi')$ as $C'_f = \langle \Omega_{C'_f}, \Psi_{C'_f}, R_{C'_f} \rangle$ where $\Omega_{C'_f} = \Omega_{C_f}$, $\forall_{D \in \Omega'} (\Psi_{C'_f}(D_i) = \Psi'(D_i))$, $\forall_{D \in \Omega} (\Psi_{C'_f}(D_i) = \Psi_{C_f}(D_i))$ and $R_{C'_f}$ is a fuzzy relation on $FD(dom(\Psi_{C'_f}(D_1))) \times \dots \times FD(dom(\Psi_{C'_f}(D_n))) \times FD(dom(M))$ defined by the level **climbing**

of tuples x in R_{C_f} , being $lc(x, \Psi_{C_f}) = y$ such that $y[M] = x[M]$, $\forall_{D \in \Omega' \wedge D \neq M} (y[D] = x[D])$, $\forall_{D \in \Omega'} (y[D] = ancestor_f(x[D], \Psi'(D)))$, $R_{C_f} = \{\mu / lc(x, \Psi_{C_f}) \mid x \in R_{C_f} \wedge \mu = \mu_{R_{C_f}}(x)\}$, $CB(C_f) = CB(C_f)$.

Let R_{C_f} the fuzzy relation presented in Table 1, then $LC(C_f, \{Time, QoS\}, \{(turn_shift, time), (QoS, level)\})$ is given in Table 4.

Table 4. Resulting R_{C_f} from $LC(C_f, \{Time, QoS\}, \{(turn_shift, time), (QoS, level)\})$

μ	Turn Shift	QoS	Time Between Stations (in seconds)
0,5	{ 1/morning }	{ 1/high }	Near of 30
0,75	{ 1/ morning, 1/ afternoon }	{ 1/regular }	Less than 60
0,33	{1/ morning }	{ 1/low }	Almost 25
0,25	{0,99/morning, 1/afternoon }	{ 1/high }	Fast
0,5	{1/afternoon }	{ 1/high }	Slow
1	{1/afternoon }	{ 1/high, 0,5/regular }	Less than 30
0,5	{1/morning, 1/afternoon }	{ 1/regular }	Fast
0,66	{1/afternoon }	{ 1/high }	More than 45
0,75	{0,9999/afternoon, 1/night }	{ 1/high }	Slow
1	{1/afternoon, 0,5/night }	{ 1/regular }	Between 30 and 60
0,25	{1/night }	{ 1/high }	Fast
0,75	{1/afternoon, 0,5/night }	{ 1/regular }	Between 25 and 30

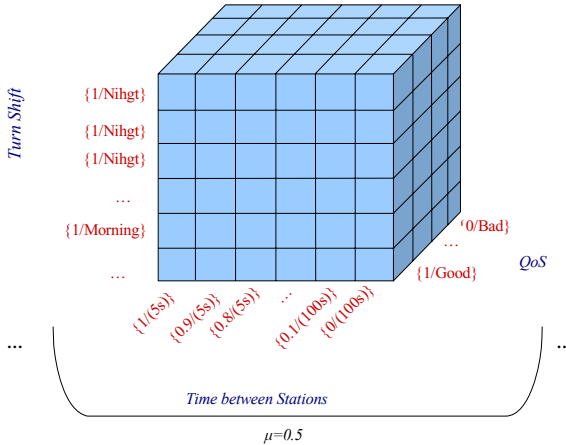


Fig. 7. Resulting Cube after applying the operator Level Climbing

5 Conclusion

We have provides a formal definition of Data Warehouse OLAP operators in presence of fuzzy information. It would be useful in decision making support form imperfect

data. Our contribution is mainly theoretical. In future works, realization of these operators must be discussed in practical way.

Acknowledgment

This work is supported by Venezuela FONACIT Project G-2005000278, France IRISA/ENSSAT Pilgrim Project and Chile UCM Project DINPER-UCM-2009-2010. Claudia and Leonid would like to praise the Lord Jesus Christ for Him amazing grace and the forces that He has give us during the realization of this work.

References

1. Fukami, S., Umamo, M., Muzimoto, M., Tanaka, H.: Fuzzy Database Retrieval and Manipulation Language IEICE Technical Reports, vol. 78, pp. 65-72. AL-78-85 (Automata and Language) (1979)
2. Umamo Freedom-O, M.: A Fuzzy Database System. In: Gupta-Sanchez (ed.) Fuzzy Information and Decision Processes. North-Holland Pub. Comp., Amsterdam (1982)
3. Galindo, J., Urrutia, A., Piattini, M.: Representation of Fuzzy Knowledge in Relational Databases. In: Proceedings of the 15th International Workshop on Databases and Expert Systems Applications, DEXA 2004, pp. 917-921 (2004)
4. Vassiliadis, P.: Modeling Multidimensional Databases, Cubes and Cube Operations. Statistical and Scientific Database Management, 53-62 (1998)
5. Feng, L., Dillon, T.S.: Using Fuzzy Linguistic Representations to Provide Explanatory Semantics for Data Warehouses. IEEE Transactions on Knowledge and Data Engineering 15(1), 86-102 (2003)
6. Pavan Kumar, K.V.N.N., Radha Krishna, P., Kumar De, S.: Fuzzy OLAP cube for qualitative analysis. Intelligent Sensing and Information Processing, 290-295 (2005)
7. González, C., Mirisola, R., Tineo, L.J., Urrutia, A.: A model for fuzzy multidimensional spaces. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2007, Part I. LNCS, vol. 4805, pp. 21-22. Springer, Heidelberg (2007)
8. Bosc, P., Pivert, O.: SQLf: A Relational Database Language for Fuzzy Querying. IEEE Trans. on Fuzzy Systems 3(1) (1995)
9. Fukami, S., Umamo, M.: Fuzzy Relational Algebra for Possibility-Distribution-Fuzzy-Relational Model of Fuzzy Data. Journal of Intelligent Information System 3, 7-27 (1994)
10. Galindo, J.: New Characteristics in FSQL, a Fuzzy SQL for Fuzzy Databases. WSEAS Transactions on Information Science and Applications 2 2, 161-169 (2005)
11. Galindo, J., Urrutia, A., Piattini, M.: Fuzzy Database Modeling, Design and Implementation. Idea Group Publishing (2006)
12. Kacprzyk, J., Zadrozny, S.: Fuzzy Queries in Microsoft Access™ v.2. In: Proceedings of Fuzzy IEEE 1995 Workshop on Fuzzy Database Systems and Information Retrieval, pp. 61-66 (1995)
13. Loo, G., Lee, K.: An interface to databases for flexible query answering: A fuzzy-set approach. In: Ibrahim, M., Küng, J., Revell, N. (eds.) DEXA 2000. LNCS, vol. 1873, p. 654. Springer, Heidelberg (2000)
14. Ma, Z.M., Yan, L.: Generalization of Strategies for Fuzzy Query Translation in Classical Relational Databases. Information and Software Technology 49(2) (February 2007)
15. Medina, J., Vila, M.A., Pons, O.: GEFRED: A Generalized Model of Fuzzy Relational Databases. Informatizan Sciences (1993)

16. Nakajima, H., Sogoh, T., Arao, M.: Fuzzy Database Language and Library-Fuzzy Extension to SQL. In: Proceedings of Second IEEE International Conference on Fuzzy Systems, pp. 477–482 (1983)
17. Wong, M., Leung, K.: A fuzzy Database-Query Language. *Information Systems* 15(5), 583–590 (1990); Zadeh, L.A.: Fuzzy sets. *Information and Control* 8 (1965)
18. Zadeh, L.A.: Fuzzy sets. *Information and Control* 8 (1965)

Caller Behaviour Classification: A Comparison of SVM and FIS Techniques

Pretesh B. Patel¹ and Tshilidzi Marwala²

¹ Researcher at the Faculty of Engineering and the Built Environment, University of Johannesburg, visiting Phd student from School of Electrical and Information Engineering, P.O. Box 524, Auckland Park, 2006, Johannesburg, South Africa
p.patel@ee.wits.ac.za

² Faculty of Engineering and the Built Environment, University of Johannesburg, P.O. Box 524, Auckland Park, 2006, Johannesburg, South Africa
tmarwala@uj.ac.za

Abstract. Accurate classification of caller interactions within Interactive Voice Response systems would assist corporations to determine caller behaviour within these telephony applications. This paper proposes a classification system with these capabilities. Fuzzy Inference Systems, Support Vector Machine and ensemble of field classifiers for a pay beneficiary application were developed. Accuracy, sensitivity and specificity performance metrics were computed and compared for these classification solutions. Ideally, a field classifier should have high sensitivity and high specificity. The Support Vector Machine field classifiers are the preferred models for the 'Say account', 'Select beneficiary' and 'Say confirmation' fields as these solutions yield the best performance results. However, the ensemble of field classifiers is the most accurate for the 'Say amount' field.

1 Introduction

Call centers experience operational challenges on a daily basis. The centers have to determine an optimal balance between reducing average call handling times and improve customer satisfaction rates. They have to reduce staffing expenses as well as decrease average call hold times.

Interactive Voice Response (IVR) systems can assist in resolving these challenges by providing a convenient, reliable as well as repeatable caller experience. An IVR system is an automated telephony system that interacts with callers, gathers relevant information and routes calls to the appropriate destinations [1]. The inputs to the IVR system can be voice, Dual Tone Multi-Frequency (DTMF) key-pad selection or a combination of the 2. IVR systems can provide appropriate responses in the form of voice, fax, callback, e-mails and other media [1]. An IVR system solution may consist of telephony equipment, software applications, databases and supporting infrastructure.

However, there are many IVR systems that have application design problems or are configured poorly that result in caller frustration when calling the system. An

example of a cause of caller frustration would be an IVR application that does not provide the caller sufficient time to respond to a prompt. Due to this, the system does not interpret the caller responses correctly. The caller experience is poor and this would probably result in a caller disconnect.

The aim of this research is to develop a field classification application, using computational intelligent methods, which could assist companies in quantifying caller behaviour within their IVR systems. It is anticipated that this application would be used in conjunction with other customer behaviour analysis techniques such as listening to recorded calls. As a result, this application should be used to confirm the system performance in relation to customer interaction.

IVR applications are developed in Voice Extensible Markup Language (VXML). VXML applications are voice-based dialog scripts that consist of form or dialog elements. The form or dialog elements are used to group input and output sections together. A field element is used to obtain and interpret user input information. As a result, the form or dialog elements contain field elements [2].

The classification system developed categorizes caller behaviour at a field within the IVR applications into specific interaction classes. As a result, these interaction classes can assist in determining trends of caller behaviour within the self service systems. For example, the field classification application can identify calls where the automated speech recognition at a particular field is low. Thereafter, analysts can listen to a sample of these calls and determine the reason for this. The field classification system can also identify the fields that resulted in the majority of the callers transferring to a Customer Service Agent (CSA) or caller disconnecting due to difficulties experienced.

In order to develop such an application, the classification of data must be accurate. This paper compares field classifiers that were developed utilizing Support Vector Machine (SVM) and Fuzzy Inference System (FIS) techniques. Ensembles of classifiers were also developed.

Support Vector Machines (SVMs) perform well for modeling challenging high-dimensional data. SVMs have been used successfully in text mining [3], image mining [4], bioinformatics [5] and information fusion [6]. SVM performance has been demonstrated to be superior to the performance of decision trees, neural networks and Bayesian techniques [3][5][6].

A fundamental method in data mining and pattern recognition is clustering of data. Fuzzy clustering involves the natural grouping of data in a large data set and provides a basis for constructing rule-based fuzzy model [7]. Fuzzy c-means, mountain clustering, subtractive clustering and entropy-based fuzzy clustering are among the fuzzy clustering algorithms used. In this paper we are interested in subtractive clustering.

The classification of data into various classes has been an important research area for many years. ANNs have been applied to pattern classification [8]. Research has also been conducted on fuzzy classification. This resulted in many algorithms, such as fuzzy K-nearest neighbour [9] and fuzzy c-means [10], being applied to classification problems. Fuzzy systems constructed using genetic algorithms have been utilized [11]. Fuzzy neural networks have also been employed in pattern classification applications [12].

SVMs have been applied to multi-category classification problems [13]. These classification tasks have also been implemented by combining multiple simpler specialized classifiers [14].

The sections to follow examine the caller behaviour classification system as well as its implementation methodology. The paper ends with the comparison of the various field classifiers developed and the selection of the superior networks.

2 The Developed System

As the developed system is to be used to identify trends of caller behaviour at a field within the IVR VXML applications, the system is trained based on data extracted from IVR log event files. These files are generated by the IVR platform as specific events occur during a call to the system. Events such as call begin, form enter, form select, automatic speech recognition events, transfer events and call end events are written to the logs [15].

Table 1 shows the inputs and outputs of the field classification system. These specific inputs have been selected to characterize the caller experience at a field within a VXML application. The outputs of the classifiers summarize the caller field behaviour through the use of interaction classes.

The confidence input illustrates the IVR speech recognition probability. The value is a percentage. The larger the percentage, the greater the probability the system interpreted the caller successfully.

A caller may answer a question the VXML application prompts with a response the application does not accommodate. These events are represented by the no match inputs. In general, most VXML applications accommodate 3 no match events per field. On a third no match event, the call is transferred to a DTMF field. If the caller fails to complete the DTMF field successfully on attempt 1, the call is transferred to a CSA. The same process is used for the third no input and maximum speech time out events. The no match field classifier inputs assist in identifying callers that misunderstood the VXML prompt as well as unique responses that the VXML application can use to improve field recognition coverage.

In response to a prompt, a caller may remain silent. These events are represented by the no input parameters. VXML applications normally accommodate 3 no input events on each field. These input parameters assist in identifying callers that were confused when prompted with the automated application question. As a result, the caller remained silent.

Callers may reply to VXML applications by talking beyond the allocated timeout period of the field. These events are represented by the maximum speech timeout input parameters of the field classifiers [16]. Maximum speech timeout input parameters are important as they assist in determining whether the timeout periods are adequate for callers to complete their responses.

Barge-in input parameters illustrate whether or not a caller interrupted the application while the automated question prompt played. Caller disconnects, transfer to DTMF, transfers to Customer Service Agents (CSAs) and System errors are represented by the hang-up, DTMF transfer, transfer to service agent and system error input parameters, respectively. These inputs can also assist in determining the level of

difficulty the caller experienced in the field. The duration input parameter illustrates the time the caller spent completing the field. Confirmation of transaction represents whether or not the caller verified the application recognition as being true.

Table 1. The inputs and outputs of the field classifier

Inputs	Outputs	Output interaction class
Confidence	Field performance	Good, acceptable, investigate, bad
No matches	Field transfer reason	Unknown, difficulty
No inputs	Field hang-up reason	Unknown, difficulty
Max speech timeouts	Field difficulty attempt	Attempt 1, attempt 2, attempt 3
Barge-ins	Field duration	High, medium, low
Hang-up	Field recognition level	High, medium, low
Transfer to Service Agent	Experienced caller	True, false
DTMF transfer		
Duration		
System error		
Confirmation of transaction		

The field performance output interaction class of the classifier will illustrate whether the caller behaviour is good, acceptable, investigate or bad. The field transfer reason and field hang-up reason interaction classes attempt to identify the motivation for the transfer to CSA or caller disconnect, respectively. Field difficulty attempt interaction class computes the number of difficulty events that occurred during the field interaction. The field duration as well as field recognition level classes illustrate 3 categories of performance, low, medium and high. As a result, these output parameters will assist in characterizing the caller experience at a VXML field.

Experienced caller output parameter categorizes whether or not the caller is a regular user of the application. In determining the number of experienced callers, the contact center can determine the usage of the application.

3 Selection and Preprocessing of Data

The data utilized in developing the classifiers is based on data extracted from IVR log event files. A business intelligence solution that involved Extract, Transform and Load (ETL) processes was created to extract and compute information such as recognition confidence values, duration values and call completion information. This information was stored within a database and was then manipulated utilizing specific rules to create the data sets. Rules such as if no hang-up, transfer to CSA, DTMF transfer, system error, no inputs, no matches or maximum speech timeouts occur, but the confidence level at the field is greater than 80%, the duration to complete the field is less than the average field duration and the field confirmation is true, the field performance interaction class would be computed as 'good', were followed.

No match, no input and maximum speech timeout information is presented to the field classifiers, using a binary notation. These inputs are presented by 3 digit binary words. For example, if a no match 1 and a no match 2 occur at a field, the binary notation will be '011'. A similar binary notation is employed for the no input and maximum speech timeout classifier inputs. The barge-in, hang-up, transfer to CSA, DTMF transfer, system error and confirmation of transaction input information were represented by bit binary words. A similar binary notation scheme has also been utilized to interpret the interaction classes outputted.

The confidence and duration input parameters of the classifiers were preconditioned by normalizing the data. Normalizing the data entails manipulating the data sets such that the values within the sets are between 0 and 1. Normalization is accomplished by acquiring the minimum and maximum values within the data sets. These values are then utilized to compute the normalized values.

The research conducted entailed the creation of 'Say account', 'Say amount', 'Select beneficiary' and 'Say confirmation' field classifiers. Caller behaviour per field is unique. For example, at a 'Say confirmation' field the caller is required to say 'yes' or 'no'. However, the caller is requested to say the account name at the 'Say account' field. As a result, the duration to complete the VXML application field is much shorter at the confirmation field. Therefore, each classifier is trained with data relevant to the field.

In order to ensure that over-fitting and under-fitting were avoided, the data has been divided into 3 sets. The data is divided into training, validation and test sets. The training data set is used to train the algorithms to find the general classification groups within the data. The validation data set is used to assess the classifier and the test data is used to confirm the classification capability of the developed models.

4 Support Vector Machine Field Classifiers

SVM is a reputable computational intelligent technique for resolving classification problems. SVMs have many advantages in solving small sample size, nonlinear and high-dimensional pattern recognition problems [17]. SVM utilizes support vector (SV) kernel functions to map the data in the input space to a higher dimensional feature space where the problem can be processed in a linear form [17]. As a result the kernel function is a key technology of SVM. The type of kernel function will affect the learning ability and generalization ability. Different kernel functions will construct different SVM classifiers.

This research considers the linear, polynomial, Radial Basis Function (RBF) and sigmoid kernel functions. Linear kernel function is suitable to problems where the number of training instances is less than the number of features within the data [18]. RBF kernel function has the ability to accommodate non-linear relationships between input instances and output classes. The sigmoid kernel function behaves similar to the RBF kernel functions for certain parameters. The RBF kernel function has less hyperparameters than the polynomial kernel function [18]. For detailed information on these kernel functions refer to [19].

SVM implementation process involved creating field classifiers that employed the kernel functions mentioned above. The validation and test data set accuracies of the resulting SVM classifiers were then compared to determine the kernel function most suitable for this application. Since this is a classification implementation, a confusion

matrix is employed to identify the number of true and false classifications that are generated by the model developed. This is then utilized to compute the true accuracy of the classifiers, using the accuracy equation stated in [8].

Good results were obtained that yielded field classifiers with excellent generalization capabilities. Table 2 illustrates the results of the SVM implementation. It is evident that the polynomial kernel function resulted in the most accurate ‘Say account’ field classifier. The linear and RBF kernel function classifiers were only 2% less accurate on unseen ‘Say account’ data. Similarly, the sigmoid kernel function ‘Say amount’ classifiers were most accurate. The linear kernel function proved to be most appropriate for the ‘Select beneficiary’ and ‘Say confirmation’ field classifiers.

The SVM field classifiers created employed a classification threshold value of 0.5. This threshold value of 0.5 proved to be adequate for the implementations, resulting in 90% accurate classifications on the training, validation and test data sets.

Table 2. Results of support vector machine implementation

Kernel function	Field classifier	Accuracy (Validation)	Accuracy (Test)
Linear	‘Say account’	0.9688	0.8814
	‘Say amount’	0.9068	0.9691
	‘Select beneficiary’	0.9447	0.9453
	‘Say confirmation’	0.9630	0.9029
Polynomial	‘Say account’	0.9047	0.9052
	‘Say amount’	0.8521	0.8473
	‘Select beneficiary’	0.8756	0.8418
	‘Say confirmation’	0.9005	0.8583
RBF	‘Say account’	0.9627	0.8832
	‘Say amount’	0.9085	0.9401
	‘Select beneficiary’	0.8950	0.8594
	‘Say confirmation’	0.9257	0.8994
Sigmoid	‘Say account’	0.9519	0.8776
	‘Say amount’	0.9093	0.9263
	‘Select beneficiary’	0.8939	0.8571
	‘Say confirmation’	0.9064	0.8703

5 Fuzzy Inference System Field Classifiers

The FIS utilized in the development of the field classifiers, employed subtractive clustering to generate the required membership functions and set of fuzzy inference rules. The objective of clustering is to locate “natural classes” in a set of given inputs such that similar inputs are grouped together in the same class [20].

The cluster radius indicates the range of influence of a cluster. A small cluster radius results in small clusters in the data and, therefore, many fuzzy rules. Large cluster radii yield few large clusters in the data and, hence, fewer fuzzy rules [20].

The cluster radius has been optimized by minimizing an error function that mapped the radius to the accuracy of the developed inference systems. This process was performed on the validation data sets.

The optimization process followed entailed the construction of various inference systems with the cluster radius ranging from 0.01 to 1. During the cluster radius optimization, classification threshold of 0.5 has been employed. Once the optimal cluster radii have been identified, the classification threshold is optimized.

Table 3 illustrates the cluster radii that resulted in the most accurate field classifiers. FIS ‘Say account’ field classifier proved to be the most accurate, yielding an accuracy of 78.00% on validation data set. However, the FIS ‘Say amount’ classifier is the least accurate, producing an accuracy of 63.11% on validation data set.

In order to improve the accuracy of the FIS field classifiers, the classification threshold is optimized. The classification threshold is optimized by minimizing an error function that mapped the classification thresholds to the accuracy of the developed classifiers. The process is performed on the validation and test data sets.

This optimization process involved varying the classification threshold from 0.1 to 0.5 in iterations of 0.01. During this process, the optimized cluster radii identified above has been used. For each of the threshold values the accuracy of the FIS is calculated using the accuracy equation mentioned in [8].

Table 3 illustrates the threshold values that resulted in the largest accuracy value for the validation and test data sets. It is evident that the validation data set accuracy of the field classifiers has improved. The FIS ‘Say amount’ classifier has become the most accurate with an accuracy of 82.54% on test data. The least accurate is the FIS ‘Select beneficiary’ classifier, yielding an accuracy of 77.82% on test data.

Table 3. Results of FIS optimization

Radius	Threshold	Field classifier	Accuracy (Validation)	Accuracy (Test)
Cluster radius optimization				
0.16	0.50	‘Say account’	0.7800	0.8723
0.26	0.50	‘Say amount’	0.6311	0.9566
0.40	0.50	‘Select beneficiary’	0.7288	0.9339
0.78	0.50	‘Say confirmation’	0.7074	0.8674
Classification threshold optimization				
0.16	0.16	‘Say account’	0.8068	0.8077
0.26	0.15	‘Say amount’	0.8265	0.8254
0.40	0.11	‘Select beneficiary’	0.7843	0.7782
0.78	0.21	‘Say confirmation’	0.7951	0.7947

6 Comparison of the Support Vector Machine and Fuzzy Inference System Field Classifiers

It is evident, from the investigations conducted, that the SVM ‘Say account’, SVM ‘Say amount’, SVM ‘Select beneficiary’ and SVM ‘Say confirmation’ field

classifiers are more accurate than the corresponding FIS classifiers by approximately 10%, 8%, 16% and 11%, respectively. The 3 most accurate SVM field classifiers were used in ensembles of classifiers. The outputs of classifiers that employed these kernel functions were fed into a voting system. The voting system determined the final output of the ensemble. If the majority of the classifiers within the ensemble categorized an output into a certain class, the voting system would generate an output as the class. If all of the models within the ensemble classified an output into different classes, the voting system would classify the output of the ensemble as undecided.

In relation to this application, the ensemble does not always result in a more accurate solution. The 'Say account', 'Say amount', 'Select beneficiary' and 'Say confirmation' ensemble of field classifiers produced validation data accuracy values of 96.16%, 90.88%, 89.66% and 92.68%, respectively. These ensembles achieved test data accuracy values of 88.04%, 94.45%, 86.09% and 89.84%, respectively.

The most accurate SVM classifier outperforms the ensemble on validation data. This is true for all fields. The 'Say account' ensemble of field classifiers is 6% more accurate on validation data. However, the SVM 'Say account' field classifier produces similar accuracy values on both data sets. This demonstrates good generalization capabilities.

The 'Say amount' ensemble of classifiers is only 0.05% more accurate on validation data. The 'Say amount' field classifier is almost 2% less accurate on test data. As a result, the 'Say amount' ensemble of classifiers is appropriate for this application. The FIS classifiers are outperformed by the ensemble of SVM models.

In order to confirm the accuracy of the various classifiers developed, sensitivity and specificity were calculated using the equations in [21]. In this research, sensitivity is defined as the probability that the field classifier categorizes a set of caller behaviour inputs to the correct specific interaction classes. Specificity is defined as the probability that the classifier indicates that a set of caller behaviour inputs does not correctly belong to specific interaction classes. The former measure describes the effectiveness of the classifier at categorizing interaction classes correctly, while the latter characterizes the performance of the classifier at discarding the other interaction classes.

Table 4 illustrates these performance metrics for the most accurate FIS, SVM and ensemble of SVM field classifiers. Ideally, a field classifier should have high sensitivity as well as high specificity. This is evident in the SVM and ensemble of SVM field classifiers performance metrics.

The SVM 'Say account' field classifier yields similar performance results on both validation and test data. This is not the case for the ensemble. As a result, the SVM 'Say account' field classifier has good generalization capabilities and is the preferred model for this field. The ensemble is more accurate, has greater sensitivity and specificity values for the 'Say amount' test data. Therefore, the ensemble of SVM 'Say amount' field classifiers is the preferred model. The SVM 'Select beneficiary' and 'Say confirmation' field classifiers are the preferred models for these fields due to the best performance metrics achieved.

Table 4. Performance metrics of field classifiers

Field classifier	Method	Sensitivity	Specificity	Sensitivity	Specificity
		Validation		Test	
'Say account'	SVM (Polynomial)	0.8655	0.9457	0.8766	0.9347
	FIS	0.7200	0.9943	0.6615	0.9863
	Ensemble	0.9439	0.9798	0.8042	0.9638
'Say amount'	SVM(Sigmoid)	0.8681	0.9524	0.8948	0.9589
	FIS	0.7022	0.9860	0.6847	0.9951
	Ensemble	0.8623	0.9577	0.9180	0.9717
'Select beneficiary'	SVM (Linear)	0.9151	0.9754	0.9159	0.9756
	FIS	0.6129	0.9900	0.6097	0.9931
	Ensemble	0.8468	0.9493	0.7795	0.9509
'Say confirmation'	SVM (Linear)	0.9498	0.9764	0.8523	0.9564
	FIS	0.7138	0.9958	0.6576	0.9604
	Ensemble	0.8944	0.9603	0.8439	0.9564

7 Conclusion

This research entailed the development of a field classification application. 'Say account', 'Say amount', 'Say confirmation' and 'Select beneficiary' field classifiers were created using FIS, SVM and ensemble of classifiers.

The implementation process entailed identifying the SVM kernel function that yielded the most accurate results. The polynomial and sigmoid kernel function resulted in the most accurate 'Say account' and 'Say amount' field classifiers, respectively. However, the linear kernel function provided to be most appropriate for the 'Select beneficiary' and 'Say confirmation' field classifiers.

The FIS classifiers were developed by initially identifying the cluster radius that resulted in the most accurate model. Thereafter, the thresholds used to interpret the classification were optimized. The accuracy of the FIS classifiers did improve, but the SVM approach has been found to be more accurate.

Ensemble of field classifiers, consisting of the 3 most accurate SVM classifiers, has also been developed. Performance metrics were computed and compared for the computational intelligent solutions. The SVM and ensemble of field classifiers achieved high accuracy, sensitivity and specificity. The SVM field classifiers are the preferred models for the 'Say account', 'Select beneficiary' and 'Say confirmation' fields as they yield the best performance results. It has also been determined that the ensemble of field classifiers is the most accurate for the 'Say amount' field.

References

- [1] Nichols, C.: The Move from IVR to Speech – Why This is the Right Time to Make the Move to Speech Applications in Customer-Facing Operations, Intervoice (2006)
- [2] VoiceXML 2.0/VoiceXML 2.1 Reference, <http://developer.voicegenie.com/voicexml2tagref.php?tag=st&display=standardtags> (last accessed: 8 April 2009)

- [3] Joachims, T.: Text categorization with Support Vector Machines: Learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998)
- [4] Tolambiya, A., Kalra, P.K.: Contrast sensitive epsilon-SVR and its application in image compression. In: 2008 IEEE International conference on Systems, Man and Cybernetics (SMC 2008), pp. 359–364 (2008)
- [5] Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J., Poggio, T., Gerald, W., Loda, M., Lander, E., Golub, T.: Multi-Class Cancer Diagnosis Using Tumor Gene Expression Signatures. Proc. National Academy of Sciences U.S.A. 98(26), 15149–15154 (2001)
- [6] Pal, M., Mather, P.M.: Assessment of the Effectiveness of Support Vector Machines for Hyperspectral Data. *Future Generation Computer Systems* 20(7), 1215–1225 (2004)
- [7] Elmzabi, A., Bellafkih, M., Ramdani, M.: An Adaptive Fuzzy Clustering Approach for the Network Management. *International Journal of Information Technology* 3(1), 12–17 (2007)
- [8] Patel, P.B., Marwala, T.: Interactive Voice Response field classifiers. In: 2008 IEEE International conference on Systems, Man and Cybernetics (SMC 2008), pp. 3425–3430 (2008)
- [9] Keller, J.M., Gray, M., Givens, J.: A fuzzy k-nearest neighbor algorithm. *IEEE Transaction on System, Man and Cybernetics* 15(4), 580–585 (1985)
- [10] Bezdek, J.C.: *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum (1981)
- [11] Russo, M.: FuGeNeSys – A fuzzy genetic neural system for fuzzy modeling. *IEEE Transaction on Fuzzy Systems* 6(3), 373–388 (1998)
- [12] Patel, P.B., Marwala, T.: Neural Networks, Fuzzy Inference Systems and Adaptive-Neuro Fuzzy Inference Systems for Financial Decision Making. In: *International Conference on Neural Information Processing*, vol. 4234(13), pp. 430–439 (2006)
- [13] Hsu, C.-W., Lin, C.-J.: A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks* 13(2), 415–425 (2002)
- [14] Zadrozny, B.: Reducing multiclass to binary by coupling probability estimates. In: *Neural Information Processing Systems Foundation*, vol. 14, pp. 1041–1048 (2002)
- [15] VoiceGenie Technologies Inc., *VoiceGenie 7 Tools User's Guide*, VoiceGenie Technologies Inc. (2005)
- [16] VoiceXML properties, <http://community.voxeo.com/vxml/docs/nuance20/VXMLproperties.html> (last accessed: April 8, 2009)
- [17] Taylor, J.S., Cristianini, N.: *Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, Cambridge (2000)
- [18] Hsu, C.W., Chang, C.C., Lin, C.J.: *A practical guide to support vector classification*, Taipei, Tech. Rep. (2003), <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>
- [19] LIBSVM - A library for Support Vector Machines, <http://www.csie.ntu.edu.tw/~cjlin/libsvm> (last accessed: April 8, 2009)
- [20] Chiu, S.: Fuzzy Model Identification Based on Cluster Estimation. *Journal of Intelligent and Fuzzy Systems* 2(3), 267–278 (1994)
- [21] Salzberg, S.L., Searls, D.B., Kasif, S.: *Computational methods in molecular biology*. Elsevier, Amsterdam (1998)

A Dual-Model Discrete-Time Jumping Fuzzy System Approach to NCS Design

Fengge Wu, Fuchun Sun, and Huaping Liu

State Key Laboratory of Intelligent Technology and Systems,
Tsinghua National Laboratory for Information Science and Technology (TNList),
Department of Computer Science and Technology,
Tsinghua University, Beijing 100084, P.R. China

Abstract. A discrete-time jump fuzzy system with two Markov chains is proposed to portray the asymmetric network characteristic of a class of nonlinear NCSs with random but bounded communication delays and packets dropout. The less conservative dual-mode-depend guaranteed cost controller is designed base on the model. The design approach can be cast into a set of nonlinear matrix inequalities (NMIs) solved by a homotopy- based iterative algorithm. Simulation examples are carried out to show the effectiveness of the proposed approaches.

Keywords: Discrete-time jump fuzzy systems, guaranteed cost control, networked control systems.

1 Introduction

Compared with the traditional control systems, networked control systems(NCSs) don't possess the data with two different characteristics, namely fixity and integrality. Usually, these NCSs could be modeled as Markovian jump linear control systems [1,2], in which random variation of system delays corresponds to randomly varying structure of the state-representation. Most of them only study the condition of sigle type communication network in the NCS. Consider the communication network between sensor and controller may not have the same structure with the one between controller and actuator. Recently, some literature considered NCSs with sensors and actuators connected to a controller via two communication networks[4,5]. The NCSs in[4] are modeled as jump linear systems with two modes characterized by two Markov chains and [5] modeled the NCSs in the the continuous-time domain. The discrete-time nonlinear model may be appropriate for NCSs because the NCSs are complicated digit systems. Therefore, advanced approaches for nonlinear NCSs are required.

Motivated by these approaches, The dual-mode discrete-time jumping Takagi-Sugeno fuzzy model is proposed to portray the network characteristic of asymmetry and nonlinear system in this paper. Then new stability theorems and The less conservative dual-mode-depend guaranteed cost controller design methods are developed for discrete-time jump fuzzy systems.

Throughout the paper, \mathbb{R}^n denotes the n -dimensional Euclidean space. I_n represents $n \times n$ unit matrix and $\text{diag}\{\cdot \cdot \cdot\}$ represents block diagonal matrix. The symmetric items in symmetric matrices are represented by “*”.

2 Dual-Mode Discrete-Time Jumping Fuzzy Model

The configuration of the NCSs composed of a controller and a remote system containing a physical plant, sensors and actuators, where the plant is nonlinear time-variant discrete-time system [6]. $\tau(k) \geq 0$ is the network induced time delay in time k from the sensor to the controller and $\rho(k) \geq 0$ is the one in time k from the controller to the actuator. These delays are assumed to be upper bounded, that is $0 \leq \tau(k) \leq \tau < \infty$, $0 \leq \rho(k) \leq \rho < \infty$. Two random delays $\tau(k)$ and $\rho(k)$ can be modeled as two homogeneous Markov chains, respectively. $\{\tau(k)\}$ is a discrete-time discrete-state Markov process taking values in a finite set $\mathcal{M} = \{0, 1, \dots, \tau\}$. The transition probability is defined as follows.

$$\lambda_{ij} = \text{Prob}\{\tau(k+1) = j | \tau(k) = i\}, \lambda_i = \text{Prob}(\tau(k) = i), i, j \in \mathcal{M}. \quad (1)$$

Here $\lambda_{ij} \geq 0$ is the transition rate from mode i to mode j , and $\sum_{j=0}^{\tau} \lambda_{ij} = 1$. $\{\rho\}$ takes values in $\mathcal{N} = \{0, 1, \dots, \rho\}$ with probability $\pi_{\kappa\ell}$, which is defined by

$$\pi_{\kappa\ell} = \text{Prob}\{\rho(k+1) = \ell | \rho(k) = \kappa\}, \pi_{\kappa} = \text{Prob}(\rho(k) = \kappa), \kappa, \ell \in \mathcal{N}. \quad (2)$$

where $\pi_{\kappa\ell} \geq 0$ and $\sum_{\ell=0}^{\rho} \pi_{\kappa\ell} = 1$.

The structured transition probability matrixes A and H similar to [6] are:

$$A = \begin{bmatrix} \lambda_{00} & \lambda_{01} & 0 & 0 & \cdots & 0 \\ \lambda_{10} & \lambda_{11} & \lambda_{12} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \lambda_{\tau-1,\tau} \\ \lambda_{\tau 0} & \lambda_{\tau 1} & \lambda_{\tau 2} & \lambda_{\tau 3} & \cdots & \lambda_{\tau,\tau} \end{bmatrix} \quad H = \begin{bmatrix} \pi_{00} & \pi_{01} & 0 & 0 & \cdots & 0 \\ \pi_{10} & \pi_{11} & \pi_{12} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \pi_{\rho-1,\rho} \\ \pi_{\rho 0} & \pi_{\rho 1} & \pi_{\rho 2} & \pi_{\rho 3} & \cdots & \pi_{\rho,\rho} \end{bmatrix}. \quad (3)$$

Assume that the model of the plant is a discrete-time jump fuzzy model as follows: For plant rule l , if $z_{k,1}$ is $M_{l,1}$ and \dots and $z_{k,p}$ is $M_{l,p}$, then $x_{k+1} = A_l x_k + B_l u_k$, where $l \in I_R = \{1, \dots, r\}$, r is the number of fuzzy rules. Suppose that $h_l(z_k)$ is determined by a local premise variable vector $z_k = [z_{k,1} \ z_{k,2} \ \cdots \ z_{k,p}]^T$. By using a center-average defuzzifier, product inference and singleton fuzzifier, the simplifying presentation of the discrete-time jump fuzzy system is inferred as follows:

$$x_{k+1} = A(H(z_k))x_k + B(H(z_k))u_k, \quad (4)$$

$$[A(H(z_k)) \ B(H(z_k))] \triangleq \sum_{l=1}^r h_l(z_k) [A_l \ B_l], \quad h_l(z_k) = \frac{\prod_{j=1}^p M_{l,j}(z_{k,j})}{\sum_{l=1}^r \prod_{j=1}^p M_{l,j}(z_{k,j})}$$

and $M_{l,j}(z_{k,j})$ is the grade of membership of $z_{k,j}$ in $M_{l,j}$.

Consider $\tau(k)$ and $\rho(k - 1)$ are available when the control law is calculated [4]. Then the mode-dependent jump state feedback control gain is:

$$u_k = K(H(z_{k-\tau(k)-\rho(k-1)}))x_{k-\tau(k)-\rho(k)}, \tag{5}$$

$$K(H(z_{k-\tau(k)-\rho(k-1)})) = \sum_{l=1}^r h_l(z_{k-\tau(k)-\rho(k-1)})K_l(\tau(k), \rho(k - 1)).$$

Substituting (5) into (4), the closed-loop system can be expressed as:

$$x_{k+1} = A(H(z_k))x_k + B(H(z_k))K(H(z_{k-\tau(k)-\rho(k-1)}))x_{k-\tau(k)-\rho(k)}. \tag{6}$$

At sampling time k , we augment the state variable $X_k = [x_k^T \ x_{k-1}^T \ \cdots \ x_{k-\tau-\rho}^T]^T$, where $X_k \in \mathbb{R}^{(\tau+\rho+1)n}$. Thus the controller gain can be designed as following:

$$U_k = K(H(z_{k-\tau(k)-\rho(k-1)}))\tilde{G}_{(\tau(k),\rho(k))}X_k \tag{7}$$

and then the closed-loop system in (6) can be written as:

$$X_{k+1} = (\tilde{A}(H(z_k)) + \tilde{B}(H(z_k))K(H(z_{k-\tau(k)-\rho(k-1)}))\tilde{G}_{(\tau(k),\rho(k))})X_k, \tag{8}$$

$$X_0 = [x_0^T \ x_{-1}^T \ \cdots \ x_{-\tau-\rho}^T]^T,$$

$$\tilde{A}(H(z_k)) = \begin{bmatrix} A(H(z_k)) & 0 & \cdots & 0 & 0 \\ I & 0 & \cdots & 0 & 0 \\ \vdots & I & \cdots & 0 & 0 \\ 0 & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I & 0 \end{bmatrix} \quad \tilde{B}(H(z_k)) = \begin{bmatrix} B(H(z_k)) \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

and $\tilde{G}_{(\tau(k),\rho(k))}$ has all elements being zeros except for the $(\tau(k) + \rho(k) + 1)$ th block being an identity matrix.

Equation(8) corresponds to a discrete-time jump fuzzy system with two modes modeled by different homogeneous Markov chains. Besides the difference in the number of modes, the other difference of the system in (8) from one-mode jump nonlinear systems is that it depends not only on the current mode $\rho(k)$, but also on the previous mode $\rho(k - 1)$. Let $\tau(k) = \iota$ and $\rho(k - 1) = \kappa$ for the convenience of notations. In this paper, assume $u_k = 0$ before the first control signal reaches the plant.

3 Guaranteed Cost Controller Design

The definition 1.1 of [7] is extended, and we have the following definitions.

Definition 1. For system (8) with $u_k \equiv 0$, the equilibrium point 0 is called to be stochastically stable, if for every initial state $X_0 = X(0)$, and initial mode

$\tau_0 = \tau(0) \in \mathcal{M}$ and $\rho_{-1} = \rho(-1) \in \mathcal{N}$ there exists a finite $W > 0$ such that the following holds:

$$\mathcal{E}\left\{\sum_{k=0}^{\infty} \|X_k\|^2 \mid X_0, \tau_0, \rho_{-1}\right\} < X_0^{\mathbb{T}} W X_0. \tag{9}$$

Lemma 1. *The closed loop system in (8) is said to be stochastically stable if and only if for any given set of symmetric matrices $O_{i\kappa} > 0, i \in \mathcal{M}, \kappa \in \mathcal{N}$, there exists a set of symmetric matrices $P_{i\kappa}$ satisfying the following coupled matrix inequalities:*

$$\sum_{\ell=0}^{\rho} \pi_{\kappa\ell} [\tilde{A}(H(z_k)) + \tilde{B}(H(z_k))K(H(z_{k-i-\kappa}))\tilde{G}_{i\ell}]^{\mathbb{T}} \bar{P}_{i\ell} \left[\left\{ \tilde{A}(H(z_k)) + \tilde{B}(H(z_k))K(H(z_{k-i-\kappa}))\tilde{G}_{i\ell} \right\} \right] - P_{i\kappa} + O_{i\kappa} < 0, \bar{P}_{i\ell} = \sum_{j=0}^{\tau} \lambda_{ij} P_{j\ell}. \tag{10}$$

Proof. Assume that for any given symmetric matrices $O_{i\kappa} > 0$, there exists a set of symmetric matrices $P_{i\kappa}$ satisfying the following coupled matrix inequalities:

$$\sum_{\ell=0}^{\rho} \pi_{\kappa\ell} [\tilde{A}(H(z_k)) + \tilde{B}(H(z_k))K(H(z_{k-i-\kappa}))\tilde{G}_{i\ell}]^{\mathbb{T}} \bar{P}_{i\ell} \times [\tilde{A}(H(z_k)) + \tilde{B}(H(z_k))K(H(z_{k-i-\kappa}))\tilde{G}_{i\ell}] - P_{i\kappa} + O_{i\kappa} < 0.$$

The weak infinitesimal operator $\tilde{A}V(X_k, k)$ is defined by:

$$\begin{aligned} \tilde{A}V(X_k, k) &= \mathcal{E}\{X_{k+1}^{\mathbb{T}} P_{\tau(k+1), \rho} X_{k+1} \mid X_k, \tau(k) = i, \rho(k-1) = \kappa\} - X_k^{\mathbb{T}} P_{i\kappa} X_k \\ &= \sum_{\ell=0}^{\rho} \sum_{j=0}^{\tau} \pi_{\kappa\ell} (X_k^{\mathbb{T}} [\tilde{A}(H(z_k)) + \tilde{B}(H(z_k))K(H(z_{k-i-\kappa}))\tilde{G}_{i\ell}]^{\mathbb{T}}) \lambda_{ij} P_{j\ell} \\ &\times \left([\tilde{A}(H(z_k)) + \tilde{B}(H(z_k))K(H(z_{k-i-\kappa}))\tilde{G}_{i\ell}] X_k \right) - X_k^{\mathbb{T}} P_{i\kappa} X_k < -X_k^{\mathbb{T}} O_{i\kappa} X_k. \end{aligned}$$

Thus, by using the Rayleigh quotient, we have:

$$\begin{aligned} \tilde{A}V(X_k, k) &= \mathcal{E}\{V(X_{k+1}, k+1) \mid X_k, \tau(k) = i, \rho(k-1) = \kappa\} - V(X_k, k) \\ &< -\lambda_{\min}(O_{i\kappa}) X_k^{\mathbb{T}} X_k = -\lambda_{\min}(O_{i\kappa}) \|X_k\|^2. \end{aligned} \tag{11}$$

Assume that $X_k \neq 0$, then

$$\frac{\tilde{A}V(X_k, k)}{V(X_k, k)} < -\frac{X_k^{\mathbb{T}} O_{i\kappa} X_k}{X_k^{\mathbb{T}} P_{i\kappa} X_k} \leq -\frac{\lambda_{\min}(O_{i\kappa}) \|X_k\|^2}{\lambda_{\max}(P_{i\kappa}) \|X_k\|^2} \leq \alpha - 1, \tag{12}$$

$$\alpha \triangleq 1 - \min_{i \in \mathcal{M}, \kappa \in \mathcal{N}} \left\{ \frac{\lambda_{\min}(O_{i\kappa})}{\lambda_{\max}(P_{i\kappa})} \right\} < 1.$$

Note that $P_{i\kappa} > 0$ for all $i \in \mathcal{M}, \kappa \in \mathcal{N}$. From (12), we have

$$\alpha \geq \frac{\tilde{A}V(X_k, k)}{V(X_k, k)} + 1 = \frac{\mathcal{E}\{V(X_{k+1}, k+1)\}}{V(X_k, k)} > 0 \tag{13}$$

That is $0 < \alpha < 1$. Thus from (11) and (12), one has

$$\begin{aligned} \tilde{A}V(X_k, k) &< (\alpha - 1)\lambda_{max}(P_{i\kappa})\|X_k\|^2 \leq (\alpha - 1)\beta\|X_k\|^2 \\ \beta &= \sup\{\lambda_{max}(P_{i\kappa}), i \in \mathcal{M}, \kappa \in \mathcal{N}\} > 0. \end{aligned}$$

From the previous inequality, we can see that for any $T \geq 1$

$$\mathcal{E}\{V(X_{T+1}, T + 1)\} - \mathcal{E}\{V(X_0, 0)\} \leq (\alpha - 1)\beta \mathcal{E}\left\{\sum_{t=0}^T \|X_t\|^2\right\},$$

Then

$$\begin{aligned} \mathcal{E}\left\{\sum_{t=0}^T \|X_t\|^2\right\} &\leq \frac{1}{(\alpha-1)\beta}(\mathcal{E}\{V(X_0, 0)\} - \mathcal{E}\{V(X_{T+1}, T + 1)\}) \\ &\leq \frac{1}{(\alpha-1)\beta}\mathcal{E}\{V(X_0, 0)\} \triangleq X(0)^\top \tilde{P}_{\tau(0)\rho(-1)} X(0). \end{aligned} \tag{14}$$

Since $0 < \alpha < 1$, $\tilde{P}_{\tau(0)\rho(-1)}$ is bounded matrix. From Definition (1), (14) implies that system (8) is stochastically stable. The proof is completed.

Lemma (1) gives sufficient conditions on the existence of the state-feedback stabilizing gain. However, since the given conditions in (8) are nonlinear in the controller gains, a method should be found to solve them. Now for the performance criterion, an upper bound of LQ cost associated with states and inputs in the global systems called Guaranteed cost is describes as follows:

$$\min \max_{h_l(z_k) \in \mathcal{H}} \sum_{k=0}^{\infty} \{X_k^\top Q_{i\kappa} X_k + U_k^\top R_{i\kappa} U_k\} \tag{15}$$

where $Q_{i\kappa} > 0, R_{i\kappa} > 0, \mathcal{H}$ and is defined as a set of all possible fuzzy weighting functions, in this case, the LQ cost will be a function of the grades $h_l(z_k) \in \mathcal{H}$. In the following theorem, we shall summarize how to find the controller (5) associated with the PQLF approach.

Theorem 1. *The closed loop jump fuzzy system (8) is stochastically stable in large and the cost (15) will be bounded by $X_0^\top P_{i\kappa} X_0$ for any nonzero initial state $X_0 = X(0)$, if there exist $\bar{X}_{i\ell} > 0, \bar{P}_{i\kappa} > 0$ and $K(H(z_{k-i-\kappa}))$ satisfying the following conditions for all $(i, j) \in \mathcal{M}, (\kappa, \ell) \in \mathcal{N}$:*

$$\begin{bmatrix} -\bar{P}_{i\kappa} & (*) & \cdots & (*) & (*) & (*) \\ \left(\begin{array}{c} \sqrt{\pi_{\kappa 0}} \left(\tilde{A}(H(z_k)) + \tilde{B}(H(z_k)) \right) \\ \times K(H(z_{k-i-\kappa})) \tilde{G}_{i0} \right) \bar{P}_{i\kappa} \end{array} \right) & -(\bar{X}_{i0}) & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \left(\begin{array}{c} \sqrt{\pi_{\kappa \rho}} \left(\tilde{A}(H(z_k)) + \tilde{B}(H(z_k)) \right) \\ \times K(H(z_{k-i-\kappa})) \tilde{G}_{i\rho} \right) \bar{P}_{i\kappa} \end{array} \right) & 0 & \cdots & -(\bar{X}_{i\rho}) & 0 & 0 \\ \bar{P}_{i\kappa} & 0 & \cdots & 0 & -Q_{i\kappa}^{-1} & 0 \\ \left(\begin{array}{c} K(H(z_{k-i-\kappa})) \\ \times \tilde{G}_{i\ell} \bar{P}_{i\kappa} \end{array} \right) & 0 & \cdots & 0 & 0 & -R_{i\kappa}^{-1} \end{bmatrix} < 0 \tag{16}$$

Furthermore, a sub-optimal guaranteed cost controller can be obtained via the following semi-definite programming:

$$\text{Minimize } \gamma \text{ subject to (16) and } \begin{bmatrix} \gamma & X_0^\top \\ X_0 & \bar{P}_{i\kappa} \end{bmatrix} \leq 0. \tag{17}$$

Proof: Consider the cost (15) associated with states as follows:

$$\min_{h_i(z_k) \in \mathcal{H}} \max_{k=0}^{\infty} \left\{ X_k^\top Q_{i\kappa} X_k + X_k^\top (K(H(z_{k-i-\kappa})) \tilde{G}_{i\ell})^\top \right. \\ \left. \times R_{i\kappa} K(H(z_{k-i-\kappa})) \tilde{G}_{i\ell} X_k \right\} \tag{18}$$

Then, the closed loop system is stable via the guaranteed cost controller, if there exists positive-definite symmetric $\bar{X}_{i\ell} > 0, P_{i\kappa} > 0$ such that for all X_k and $(i, j) \in \mathcal{M}, (\kappa, \ell) \in \mathcal{N}$, the following condition holds:

$$\tilde{A}V(X_k, k) + X_k^\top Q_{i\kappa} X_k + X_k^\top (K(H(z_{k-i-\kappa})) \tilde{G}_{i\ell})^\top R_{i\kappa} K(H(z_{k-i-\kappa})) \tilde{G}_{i\ell} X_k < 0$$

Then we obtain:

$$\sum_{\ell=0}^{\rho} \pi_{\kappa\ell} [\tilde{A}(H(z_k)) + \tilde{B}(H(z_k))K(H(z_{k-i-\kappa})) \tilde{G}_{i\ell}]^\top \bar{P}_{i\ell} \\ \times [\tilde{A}(H(z_k)) + \tilde{B}(H(z_k))K(H(z_{k-i-\kappa})) \tilde{G}_{i\ell}] - P_{i\kappa} + Q_{i\kappa} \\ + (K(H(z_{k-i-\kappa})) \tilde{G}_{i\ell})^\top R_{i\kappa} K(H(z_{k-i-\kappa})) \tilde{G}_{i\ell} < 0 \tag{19}$$

Using Schur complements, we have the following matrix inequality:

$$\begin{bmatrix} -P_{i\kappa} & (*) & \cdots & (*) & (*) & (*) \\ \left(\begin{array}{c} \sqrt{\pi_{\kappa 0}} \left(\tilde{A}(H(z_k)) + \tilde{B}(H(z_k)) \right) \\ \times K(H(z_{k-i-\kappa})) \tilde{G}_{i0} \end{array} \right) & -(\bar{P}_{i0})^{-1} \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \left(\begin{array}{c} \sqrt{\pi_{\kappa \rho}} \left(\tilde{A}(H(z_k)) + \tilde{B}(H(z_k)) \right) \\ \times K(H(z_{k-i-\kappa})) \tilde{G}_{i\rho} \end{array} \right) & 0 & \cdots & -(\bar{P}_{i\rho})^{-1} & 0 & 0 \\ I_{n(1+\tau+\rho)} & 0 & \cdots & 0 & -Q_{i\kappa}^{-1} & 0 \\ \left(\begin{array}{c} K(H(z_{k-i-\kappa})) \\ \times \tilde{G}_{i\ell} \end{array} \right) & 0 & \cdots & 0 & 0 & -R_{i\kappa}^{-1} \end{bmatrix} < 0 \tag{20}$$

The left-hand side of the inequality (20) can be pre- and post-multiplied by J^\top

and J , respectively, where $J = \text{blockdiag} \left\{ P_{i\kappa}^{-1}, \underbrace{I_{n(1+\tau+\rho)}, \dots, I_{n(1+\tau+\rho)}}_{\rho+2}, I_m \right\}$

and let $\bar{P}_{i\kappa} \triangleq P_{i\kappa}^{-1}, \bar{X}_{i\ell} \triangleq P_{i\ell}^{-1}$. We obtain the condition (16). So, if the condition

(16), $\bar{X}_{i\ell} > 0, \bar{P}_{i\kappa} > 0$ and $K(H(z_{k-i-\kappa}))$ hold for all $(i, j) \in \mathcal{M}, (\kappa, \ell) \in \mathcal{N}$, then $\tilde{A}V(X_k, k) < 0$ at $X_k \neq 0$. If the condition (16) holds,

$$X_k^\top Q_{i\kappa} X_k + U_k^\top R_{i\kappa} U_k < -\tilde{A}V(X_k, k) \tag{21}$$

Then summing both sides of the inequality (21) from 0 to ∞ and using the established closed-loop stability, the cost (15) will be bounded $X_0^\top P_{i\kappa} X_0$ for any nonzero initial state X_0 :

$$\max_{h_l(z_k) \in \mathcal{H}} \sum_{k=0}^{\infty} \{X_k^\top Q_{i\kappa} X_k + U_k^\top R_{i\kappa} U_k\} < X_0^\top P_{i\kappa} X_0.$$

Since any feasible solutions $\gamma, \bar{X}_{i\ell} > 0, \bar{P}_{i\kappa} > 0$ and $K(H(z_{k-i-\kappa}))$ yielding (17) will also satisfy

$$\max_{h_l(z_k) \in \mathcal{H}} \sum_{k=0}^{\infty} \{X_k^\top Q_{i\kappa} X_k + U_k^\top R_{i\kappa} U_k\} < X_0^\top P_{i\kappa} X_0 \leq \gamma,$$

for any h_l and nonzero X_0 , we can use (17) to minimize $X_0^\top P_{i\kappa} X_0$ for known nonzero initial states. The proof is completed.

The design problem to determine the state feedback gains $K(H(z_{k-i-\kappa}))$ for (16) can be defined as follows: Find $\bar{X}_{i\ell} > 0, \bar{P}_{i\kappa} > 0$ and $K(H(z_{k-i-\kappa}))$ with the constraints (16) such that (17) are satisfied. However, in general cases, the inequalities (16) cannot be transformed equivalently to LMIs and we will utilize the homotopy method [8] to solve it in an iterative manner.

4 Simulation Examples

To illustrate the proposed theoretical results, a numerical example is considered in this section. We consider a nonlinear system which can be represented by the following T-S fuzzy model:

- Rule 1 : IF x_{k1} is M_{l1} THEN $x_{k+1} = A_1 x_k + B_1 u_k$
- Rule 2 : IF x_{k1} is M_{l2} THEN $x_{k+1} = A_2 x_k + B_2 u_k$

where

$$x_k = [x_{k1} \ x_{k2}]^\top$$

$$A_1 = \begin{bmatrix} -0.5 & 0.1 \\ 1 & 0 \end{bmatrix} \quad B_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad A_2 = \begin{bmatrix} -1 & 0.1 \\ 1 & 0 \end{bmatrix} \quad B_2 = \begin{bmatrix} 1 & 0.1 \\ 1 & 0 \end{bmatrix}$$

Define $h_1(x_{k1}) = 1 - \frac{x_{k1}^2}{2.56}$, $h_2(x_{k2}) = 1 - h_1(x_{k1})$, then $h_1(x_{k1})$ and $h_2(x_{k2})$ can be viewed as membership functions of fuzzy sets M_{l1} and M_{l2} . the random delays exist in $\mathcal{M} = \{0, 1\}$ and $\mathcal{N} = \{0, 1\}$, and their transition probability matrices are given by

$$A = \begin{bmatrix} 0.4 & 0.6 \\ 0.1 & 0.9 \end{bmatrix}, \quad \Pi = \begin{bmatrix} 0.2 & 0.8 \\ 0.5 & 0.5 \end{bmatrix}$$

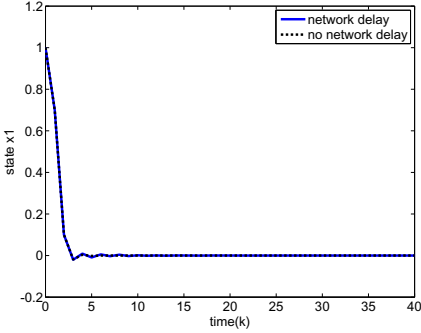


Fig. 1. Complete graph of state x_1

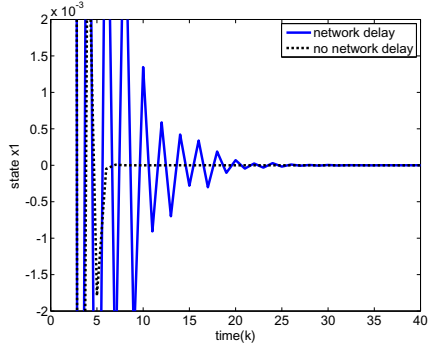


Fig. 2. Local graph of state x_1

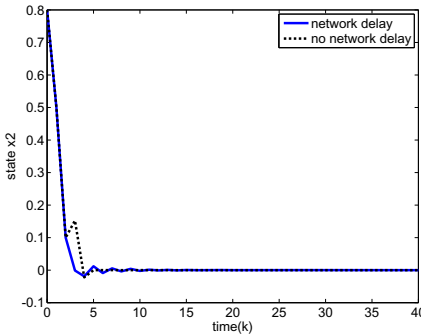


Fig. 3. Complete graph of state x_2

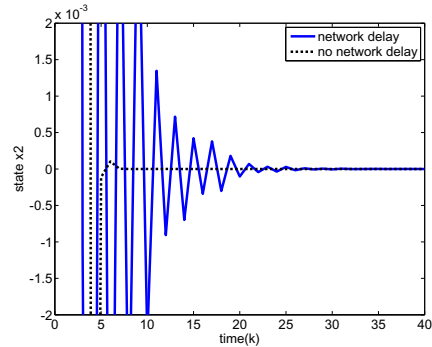


Fig. 4. Local graph of state x_2

The weighting matrices are selected as: $Q = I$, $R = 10I$. Then the guaranteed cost state-feedback controller can be obtained as follows. For the initial condition:

$$x_{01} = [1 \ 0.8]^T, x_{02} = [0.7 \ 0.5]^T, x_{03} = [0.1 \ 0.1]^T$$

Suppose there is no network delay in the system, the initial values of the feedback controllers are obtained.

$$K^1 = \begin{bmatrix} 0.5267 & -0.1064 \\ 1.0585 & -0.0004 \end{bmatrix} \quad K^2 = \begin{bmatrix} 0.0902 & -0.0533 \\ 5.7953 & -0.2577 \end{bmatrix}$$

To design the state feedback controller, we apply the homotopy algorithm, and the initial values are selected as the state-feedback gains given above. So, the state feedback gains for Rule 1 can be obtained as:

$$K_{00}^1 = \begin{bmatrix} 0.0409 & -0.0453 \\ -0.0453 & -0.0008 \end{bmatrix} \quad K_{01}^1 = \begin{bmatrix} 0.1322 & -0.1444 \\ -0.1444 & -0.0011 \end{bmatrix}$$

$$K_{10}^1 = 0.001 \times \begin{bmatrix} -0.1250 & 0.0959 \\ 0.0959 & -0.0320 \end{bmatrix} \quad K_{11}^1 = 0.001 \times \begin{bmatrix} -0.4324 & 0.3321 \\ 0.3321 & -0.1084 \end{bmatrix}$$

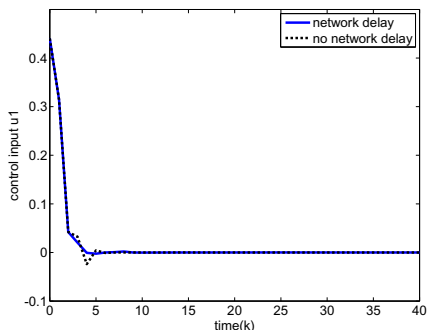


Fig. 5. Complete graph of control input u_1

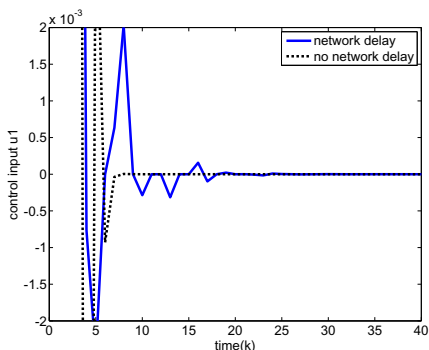


Fig. 6. Local graph of control input u_1

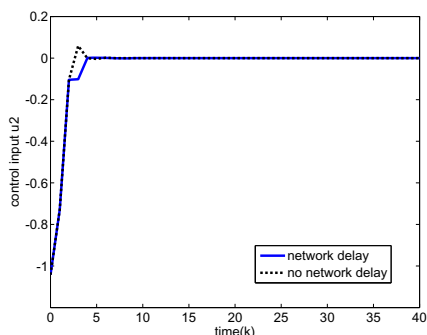


Fig. 7. Complete graph of control input u_2

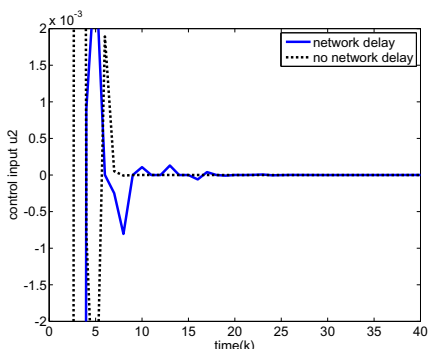


Fig. 8. Local graph of control input u_2

And the state feedback gains for Rule 2 can be obtained as:

$$\begin{aligned}
 K_{00}^2 &= \begin{bmatrix} -0.0019 & -0.0017 \\ -0.0017 & 0.000007 \end{bmatrix} & K_{01}^2 &= \begin{bmatrix} -0.0015 & -0.0047 \\ -0.0047 & -0.0009 \end{bmatrix} \\
 K_{10}^2 &= 0.0001 \times \begin{bmatrix} 0.3481 & 0.0630 \\ 0.0630 & -0.1672 \end{bmatrix} & K_{11}^2 &= 0.001 \times \begin{bmatrix} -0.1426 & -0.0036 \\ -0.0036 & 0.0304 \end{bmatrix}
 \end{aligned}$$

The response behaviors of the closed-loop system are presented in Fig. 14 using the state feedback fuzzy controller where dashed lines denote no network delay condition. Fig. 1 and Fig. 3 are the system state responses, while Fig. 5 and Fig. 7 are control variables curves.

It is shown from these Figures that the variables for no network delay condition converge faster. However it is clear from fig. 14 that the nonlinear system is stochastic stable.

5 Conclusions

In this paper, we studied the problem of modeling and controller design for networked control systems, where a dual-mode discrete-time jump fuzzy system

is developed to model networked control systems with sensors and actuators connected to a controller via two communication networks. On the basis of the assumption that all state variables of an NCS are available, a state feedback controller is developed for the dual-mode discrete-time jump system with sub-optimal guaranteed cost performance based on a piecewise quadratic Lyapunov function. It is shown that the state feedback sub-optimal fuzzy controller can be obtained by solving a set of NMIs using the homotopy approach. The LMI technique is used to effectively minimize the overall cost function and thus achieve the sub-optimal system. Finally, The effectiveness of the proposed approaches are verified by the numerical examples.

Acknowledgments. This work was jointly supported by the National Science Fund for Distinguished Young Scholars under the grant No: 60625304 and the National Natural Science Foundation of China under the grant No: 90716021,60621062 and 60873251.

References

1. Hu, S., Zhu, Q.: Stochastic optimal control and analysis of stability of networked control systems with long delay. *Automatica* 39, 1877–1884 (2003)
2. Walsh, G., Beldiman, O., Bushnell, L.: Asymptotic behavior of nonlinear networked control systems. *IEEE Transactions on Automatic Control* 46(7), 1093–1097 (2001)
3. Wang, L., Feng, G., Hesketh, T.: Piecewise output feedback controller synthesis of discrete time fuzzy systems. In: *Proceedings of the 42nd Conference on Decision and Control*, pp. 4741–4746 (2003)
4. Zhang, L., Shi, Y., Chen, T., Huang, B.: A new method for stabilization of networked control systems with random delays. *IEEE Transactions on Automatic Control* 50(8), 1177–1181 (2005)
5. Huang, D., Nguang, S., Patel, N.: Fuzzy control of uncertain nonlinear networked control systems with random time-delays. In: *Proceedings of American Control Conference*, pp. 299–304 (2007)
6. Sun, F., Wu, F.: A discrete-time jump fuzzy system approach to networked control systems design. *Networked Control Systems: Theory and Applications*, pp. 233–259. Springer, Heidelberg (2008)
7. Boukas, E., Shi, P.: Stochastic stability and guaranteed cost control of discrete-time uncertain systems with Markovian jump parameters. *International Journal of Robust Nonlinear Control* 8(13), 1155–1167 (1998)
8. Liu, H., Sun, F., Hu, Y.: H^∞ control for fuzzy singularly perturbed systems. *Fuzzy Sets and Systems* 155, 272–291 (2005)

A Continuous-Time Recurrent Neurofuzzy Network for Black-Box Modeling of Insulin Dynamics in Diabetic Type-1 Patients*

Marcos Angel González-Olvera¹, Ana Gabriela Gallardo-Hernández¹, Yu Tang^{1,**}, Maria Cristina Revilla-Monsalve², and Sergio Islas-Andrade²

¹ Facultad de Ingeniería, Universidad Nacional Autónoma de México
mangel@verona.fi-p.unam.mx, anagabygh@gmail.com, tang@unam.mx

² Unidad de Investigación de Enfermedades Metabólicas, IMSS

Abstract. Diabetic Type-1 patients have no pancreatic insulin secretion, and an insulin therapy is prescribed for them to regulate glucose absorption. There are several self monitoring devices for glucose, but not for insulin, which must be known and kept within certain limits to avoid damages to the body. Currently, it is possible to obtain real-time glucose measurements, so control schemes can be designed to control the glucose level using the insulin rate injected to the patient. In this work we present a black-box modeling of the insulin dynamics in different *in silico* patients using a recurrent neural network that could be used for on-line monitoring of the insulin concentration for a better treatment. The inputs for the identification is the rate of insulin ($\mu U/dl/min$) applied to the patient, and blood glucose concentration. The output is insulin concentration ($\mu U/ml$) present in the blood stream. The model is validated through numerical simulations.

Keywords: Identification, nonlinear systems, Biological Systems, Recurrent Neurofuzzy Networks, Lyapunov stability, Diabetes.

1 Introduction

In insulin dynamics, as in other control problems, a good model of the system to be controlled is the first step in the design of a control scheme. There exists a vast literature that covers the modeling of insulin dynamics [1,2,3,4,5], from simple models like Bergman Minimal Model [1] to very detailed models as the one proposed by Sorensen [3] with 21 states. Common problems such as parametric uncertainty and/or non-modeled dynamics arise when obtaining such a model, making difficult to apply deterministic methods to estimate the insulin based on these models. Also, it is not always possible to obtain a model for every patient, due to complexity of the measurements and the associated expenses.

* This work is partly supported by UNAM-PAPIIT IN120009. The work of Marcos A. González Olvera and Ana G. Gallardo is supported by CONACyT.

** Corresponding author.

Neural networks and fuzzy systems have proven useful for identifying non-linear systems, first as classifiers or controllers [6], and then as static function identifiers [7,8] for smooth functions in closed sets. The interpretation of a fuzzy system as a neural network, called *neurofuzzy network*, has been used to combine the best properties of both: having the space partition via membership functions, and take advantage of the training algorithms designed for neural networks, as in [9].

In the so called *recurrent neurofuzzy networks* (RNFN) internal dynamics are embedded in these networks, . One of the very first approaches with one-bit memory is given by [10]; but rapidly interest grown and then several models with different structures have been proposed [11,12,13,14,15]. The training of recurrent networks began with usual training schemes (as backpropagation or gradient techniques [16,17,18]) that are known to be sluggish and lack of stability analysis, making it necessary to propose both a network structure and a proper stable and convergent training algorithm in later works. For networks used in control or estimation problems, structures with state-space representation are preferred in some applications [14]. However, they require the actual measurement of the physical states of the system, which is not always the case. In [19], a network with a stable training algorithm is proposed, which generates its own states in order to have the same input-output dynamics as a real system, alleviating the need to measure those of the system.

In this paper we propose a *black-box model* in state-space representation that identifies and models the insulin dynamics in diabetic Type-1 patients, by analyzing the response of the blood stream insulin concentration when insulin is directly fed into the body and when the glucose concentration is measured in real-time. The training of the network is done using simulated lab tests in which both glucose and insulin concentrations are measured in the bloodstream. Once the training is completed, the objective is to have the network to reproduce the insulin dynamics when only glucose measurements and the rate of the injected insulin are available.

This work is structured in the following manner: In Section 2 the basic modeling problem of the insulin dynamics is discussed. For this system, a *black-box* structure based on Recurrent Neural Networks is presented in Section 3, jointly with a training algorithm, where also the boundness and convergence of the training algorithm are discussed. Examples for the validation of the performance of the neural identifier are shown in Section 4. Finally, conclusions and future work are drawn in Section 5.

2 Problem Statement

Glucose is the principal energy substrate of a human body, but at least 80% of body cells can not absorb it by their own, so they need insulin, a hormone produced in pancreas by Beta cells (β cells), to do it. This hormone also regulates the glucose storage in different tissues (liver, muscle) to be used in fasting periods. When insulin is higher than its basal level this tissues absorb glucose to store it, and the lack of insulin activates the mechanisms to release it [20].

In Diabetes Type-1 β cells are destroyed by autoimmune mechanisms, therefore insulin can not be produced. In consequence, the glucose concentration is always higher than its basal level. This is treated by three or four daily glucose measurement and the consequent subcutaneous insulin injections [21], which can also be delivered by an insulin pump with real-time glucose monitors and implantable sensors. This represents the sensor and actuator for a closed loop control for glucose. There are several research works about this topic [22,23,24,25].

Insulin therapy can have some complications, as *hyperinsulinemia* that produces hypoglycemia, a life-threatening condition, whose persistence could cause insulin resistance and chronically lead to excessive lipolysis and unrestrained fatty acid oxidation producing metabolic acidosis and dehydration [26]. It is then desirable to design a system that can estimate the concentration of insulin within the bloodstream to help that those limit conditions bounded. This can be considered as a system with two inputs (the glucose concentration in the bloodstream and the injected insulin) and one output (insulin concentration in the bloodstream). The purpose is that, given input-output measurements, to obtain a continuous-time dynamic model based on RNFN that can learn this dynamics and generate an estimate of the output.

Treated as an identification problem, the system is a *unknown* MISO system

$$\begin{aligned}\dot{\mathbf{x}}(t) &= f(\mathbf{x}, \mathbf{u}), \\ y(t) &= h(\mathbf{x}),\end{aligned}\tag{1}$$

where $\mathbf{u}(t) = [u_1(t) \ u_2(t)]^T \in \mathfrak{R}^2$ and $y(t) \in \mathfrak{R}$ are the input and the output, respectively, and $\mathbf{x} \in \mathfrak{R}^n$ is the state of the system, $f : \mathfrak{R}^n \times \mathfrak{R}^2 \rightarrow \mathfrak{R}^n$, $h : \mathfrak{R}^n \rightarrow \mathfrak{R}$, are unknown smooth functions, dependent on the dynamics of the absorption of the glucose and the insulin present in the bloodstream. For this application, It is assumed that (1) is observable in the region of interest. The number of states n varies depending on the precision required from the model (from 3 states, as in [1], to 21 in [3]).

Following [27], we also hypothesize (and then assume) that this system pertains to the class of nonlinear systems transformable to an output feedback form

$$\begin{aligned}\dot{\mathbf{z}} &= A\mathbf{z} + \bar{f}(y, \mathbf{u}), \\ y &= C\mathbf{z},\end{aligned}\tag{2}$$

with (A, C) observable, and $\bar{f}(y, \mathbf{u})$ smooth. Also, it is assumed that (1) fulfills the sufficient and necessary conditions in [27] in order to transform (1) through a local diffeomorphism $\mathbf{z} = \Phi(\mathbf{x})$ into the system (2).

To identify this system, We propose the RNFN to approximate \bar{f} in (2) via $\varphi(y, \mathbf{u})\theta_\eta$ by:

$$\begin{aligned}\dot{\eta} &= A\eta + \varphi(y, \mathbf{u})\theta_\eta, \\ y &= C\eta,\end{aligned}\tag{3}$$

From [28,29], this is a dual identification problem. Under persistent excitation condition for the regressor $\varphi(y, \mathbf{u})$ embedded in the neurofuzzy network, an

adaptive observer can be designed such that jointly estimates $\hat{\eta}$ and θ_η via a decoupled algorithm.

3 Proposed Recurrent Neurofuzzy Network and Training Algorithm

The neurofuzzy system (3) is described by the r fuzzy rules:

$$R_i : \text{ If } (\mathbf{u}, y) \text{ is } F_i \text{ then } \dot{\eta}^1 = A\eta + \boldsymbol{\theta}_{u1,i}u_1 + \boldsymbol{\theta}_{u2,i}u_2 + \boldsymbol{\theta}_{y,i}\rho(y),$$

where the fuzzy sets F_i are characterized by the radial basis functions

$$F_i = \exp(-\sigma_{u1,i}^2(u_1 - \mu_{u1,i})^2 - \sigma_{u2,i}^2(u_2 - \mu_{u2,i})^2 - \sigma_{y,i}^2(y - \mu_{yi})^2), \quad (4)$$

and $u_1(t)$ and $u_2(t)$ are the injected insulin rate and glucose concentration respectively. The consequent parameters are

$$\begin{aligned} \boldsymbol{\theta}_{u1,i} &= [\theta_{u1,i,1} \ \theta_{u1,i,2} \ \dots \ \theta_{u1,i,n}]^T, \\ \boldsymbol{\theta}_{u2,i} &= [\theta_{u2,i,1} \ \theta_{u2,i,2} \ \dots \ \theta_{u2,i,n}]^T, \\ \boldsymbol{\theta}_{y,i} &= [\theta_{y,i,1} \ \theta_{y,i,2} \ \dots \ \theta_{y,i,n}]^T. \end{aligned}$$

Remark 1. The main function that the fuzzy structure has in the neurofuzzy network is to provide with a space partition i , and then each section be modeled by a nonlinear system in the form of R_i .

With the defuzzification defined as the weighted sum $\dot{\eta} = \sum_{i=1}^r F_i \dot{\eta}^i / \sum_i F_i$, φ is defined by

$$\varphi(y, \mathbf{u}) = [\varphi_{u1}(y, \mathbf{u}) \ \varphi_{u2}(y, \mathbf{u}) \ \varphi_y(y, \mathbf{u})] \in \mathfrak{R}^{n \times 3nr}, \quad (5)$$

and $\varphi_{u1}, \varphi_{u2}, \varphi_y \in \mathfrak{R}^4$ can be expressed as $\varphi_{u1}(y, \mathbf{u}) = Fu_1 / \sum_i F_i$, $\varphi_{u2}(y, \mathbf{u}) = Fu_2 / \sum_i F_i$, $\varphi_y(y, \mathbf{u}) = F\rho(y) / \sum_i F_i$, with $F = [F_1 I_{n \times n} \ \dots \ F_r I_{n \times n}]$, $\rho : \mathfrak{R} \rightarrow \mathfrak{R}$, $\rho = \rho(y)$ a smooth sigmoid function, monotonically increasing and $\lim_{y \rightarrow \pm\infty} \rho(y) = \pm 1$, $\rho(0) = 0$.

So,

$$\dot{\eta} = A\eta + [\varphi_{u1}(y, \mathbf{u}) \ \varphi_{u2}(y, \mathbf{u}) \ \varphi_y(y, \mathbf{u})]\boldsymbol{\theta} = A\eta + \varphi(y, \mathbf{u})\boldsymbol{\theta}. \quad (6)$$

Consequently, the parameter vector $\boldsymbol{\theta}$ is

$$\boldsymbol{\theta} = [\boldsymbol{\theta}_{u1}^T \ \boldsymbol{\theta}_{u2}^T \ \boldsymbol{\theta}_y^T]^T \in \mathfrak{R}^{3nr}, \quad (7)$$

with $\boldsymbol{\theta}_{u1} = [\boldsymbol{\theta}_{u1,1}^T \ \dots \ \boldsymbol{\theta}_{u1,r}^T]^T \in \mathfrak{R}^{nr}$, $\boldsymbol{\theta}_{u2} = [\boldsymbol{\theta}_{u2,1}^T \ \dots \ \boldsymbol{\theta}_{u2,r}^T]^T \in \mathfrak{R}^{nr}$, $\boldsymbol{\theta}_y = [\boldsymbol{\theta}_{y,1}^T \ \dots \ \boldsymbol{\theta}_{y,r}^T]^T \in \mathfrak{R}^{nr}$.

Note that the inclusion of the sigmoid function ρ has two objectives: to provide with more nonlinear identification capabilities to the network, and to guarantee

for this structure that $\varphi_y(\mathbf{0}, 0)\theta_y = 0$. Also, The structure of the network is such that the research in adaptive observers made by [30,27,29] can be used, so the regressor can be obtained as a function of the input and output signals from (1). The consequent part in the fuzzy rules is designed in such a way to meet with the persistence of excitation condition. However, other type of fuzzy rules may be considered, as long as they satisfy the persistence of excitation condition. It is noticed that by construction, the regressor φ is always bounded if u is bounded.

3.1 Observation-Training Algorithm

In order to initialize the membership functions, a simple clustering algorithm is proposed to be used. In this case, we propose to use the Fuzzy C-Means [31,32], but other clustering methods may be employed. The number of clusters is to be equal to the number of rules in the fuzzy part of the network.

We now let the network defined by (3) to work as a parallel model for (2), which in turn is equal to the order of the system (3). Following the same treatment as in [19] and [29], the system can be trained by jointly estimating η and θ_η , with the observable pair (A, C) . Designing a vector K such that the eigenvalues of $A - KC$ are in any desired position, the training equations are

$$\dot{\Upsilon} = (A - KC)\Upsilon + \varphi, \tag{8}$$

$$\dot{\hat{\eta}} = A\hat{\eta} + \varphi(y, \mathbf{u})\hat{\theta}_\eta + (K + \Upsilon\Gamma\Upsilon^T C^T)(y - C\hat{\eta}), \tag{9}$$

$$\dot{\hat{\theta}}_\eta = \Gamma\Upsilon^T C^T(y - C\hat{\eta}), \tag{10}$$

where $\Gamma > 0$, the gain matrix, $K \in \mathbb{R}^n$ is such that $A - KC$ is Hurwitz, and (10) is a filter for φ . This training algorithm can be proven to be stable, under a condition of persistent excitation, as indicated in the Theorem 1.

Theorem 1. *Consider the model (2). The neurofuzzy network (3) with the training algorithm (8), (9), (10) and $\Gamma > 0$, $K \in \mathbb{R}^n$ such that $A - KC$ is Hurwitz, can approximate (2) and guarantees that all signals are bounded, and the state estimation error $\tilde{\mathbf{z}} = \hat{\eta} - \mathbf{z}$ and parameter error $\tilde{\theta} = \hat{\theta}_\eta - \theta_\eta$ are ultimately bounded, provided that the regressor matrix is persistently exciting, i.e.,*

$$\int_t^{t+T} \Upsilon^T(\tau)C^T C\Upsilon(\tau)d\tau \in [\alpha I, \beta I]. \tag{11}$$

for some $0 < \alpha < \beta$, and $T > 0$.

Proof. Following the Stone-Weierstrass Theorem, the smooth function $\bar{f}(y, \mathbf{u})$ is approximated by $\varphi(y, \mathbf{u})\theta_\eta$ with arbitrary precision and modeling error ε , so $\bar{f}(y, u) - \varphi(y, \mathbf{u})\theta_\eta = \varepsilon(t)$, assuming that θ_η is the set of parameters that minimize the identification error $y - \hat{y}$ (depending on the minimizing criteria considered). In this sense, the system (2) can be represented as

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \varphi(y, \mathbf{u})\theta_\eta + \varepsilon(t), \quad y = \mathbf{C}\mathbf{z}, \tag{12}$$

Now, considering a network (3) such that η is meant to be an estimate of \mathbf{z} , and defining the error signal $\xi(t) = \tilde{\mathbf{z}} - \Upsilon\tilde{\theta}$, its derivative equation represents the error signal dynamics, which is

$$\dot{\xi}(t) = (A - KC)\xi - \varepsilon, \quad \dot{\tilde{\theta}} = -\Gamma\Upsilon^T C^T C(\xi + \Upsilon\tilde{\theta}), \quad (13)$$

So, while ε is bounded, ξ is bounded, as the matrix $A - KC$ is Hurwitz, and Υ must be bounded as is obtained from a stable filtering of φ , that is also bounded.

Now, it can be proven [29] that under a persistent excitation condition (11) the system $\dot{\chi} = -\Gamma\varphi^T\varphi\chi$ is exponentially stable. So, the homogeneous part of (13) is also exponentially stable. Therefore, (13) is exponentially stable with a bounded perturbation, which implies the boundness and the ultimate boundness of $\tilde{\theta}$.

Remark 2. The persistent excitation condition for the particular case of the identification of the insulin dynamics implies that sufficient measurements must be obtained in the laboratory from the patient under several conditions, like before and after meals, and under different levels of injected insulin.

Remark 3. This algorithm can be used to train a network by epochs, and then the matrix gain Γ can be changed in order to make the training finer as the epoch number increases. In general, we considered this to be a linear relation in the form

$$\Gamma_k = \frac{\Gamma_{final} - \Gamma_{initial}}{Epochs - 1}(k - 1) \quad (14)$$

where k is the current epoch, Γ_{final} is the desired value of Γ in the last epoch, $\Gamma_{initial}$ the gain at $k = 1$ (the first epoch), and $Epochs$ the total number of epochs.

4 Modeling and Identification of Insulin Dynamics in Different Patients

For validation purposes, the network is tested with simulated data in the identification of the responses of an *in silico* patient, where the input $u(t)$ is the rate of insulin injected into the patient in $\mu U/ml/min$, calculated by the Quasi-Continuous algorithm described in [22], and glucose concentration, G (mg/dl) the output is the measured insulin concentration in blood I ($\mu U/ml$). It is true that in a healthy human being, the glucose dynamics has a strong correlation with the amount of insulin present in the blood stream. However, in diabetic patients this relation suffers from a decoupling and relatively cheap and fast sensors exists, the glucose is here used as another input to the system.

The data of the *in silico* patient is obtained using the well known Bergman Minimal Model with parameters published on [23]. This is done in two phases: First, an RNFN is trained assuming that both glucose and insuline data can be sampled 5 times per second, in order to show the approximation capabilities of the network. However, in real experiments this can not be expected to be

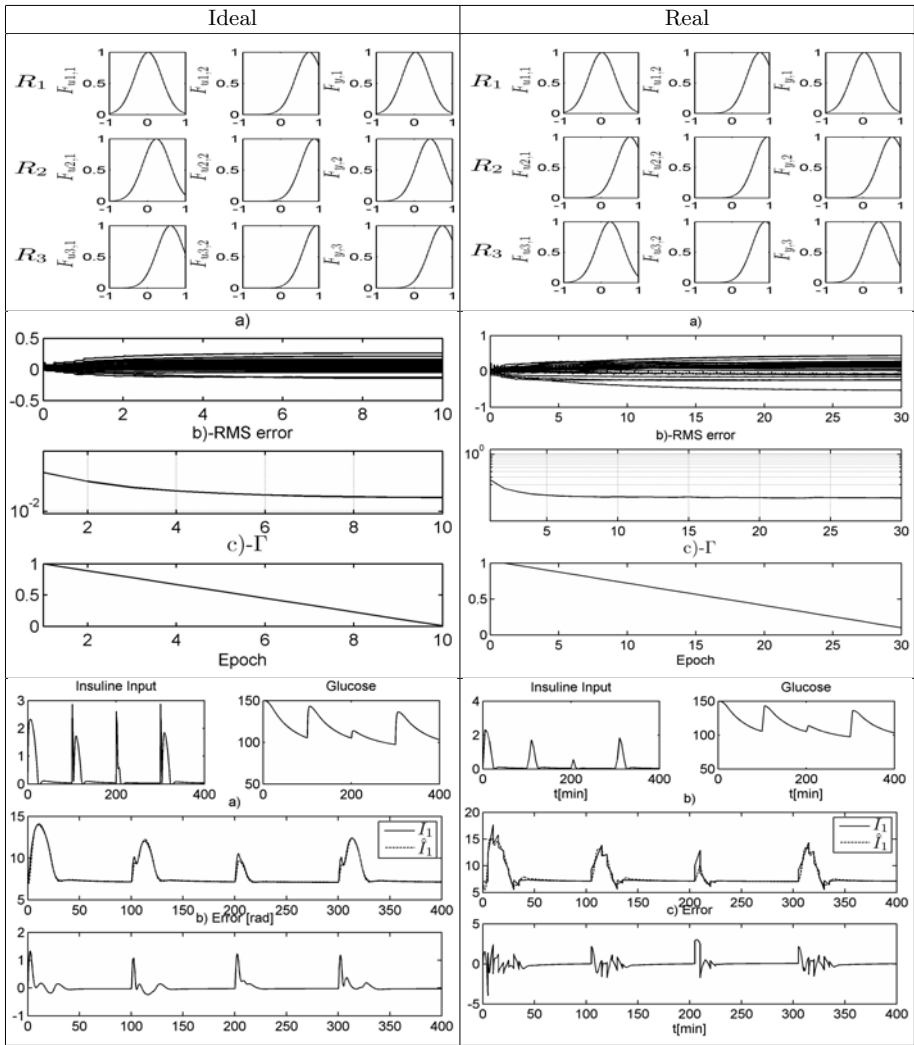


Fig. 1. Training: Ideal and Real case

possible, as experimental data would require the actual blood samples and only a limited number of them can be done in a certain period of time, so results using data sampled each 5 minutes are shown (also, as each measurement requires blood, a limit of blood extracted from a patient must be taken into account). Once the network has been trained, it is tested compared to the actual amount of insulin present in the bloodstream, assuming it can not be measured and only estimated by the RNFN.

The time unit is considered to be one minute, with three rules ($r = 3$) and three states ($n = 3$). The design values are chosen as $\lambda\{\bar{A}\} = [-0.8 \ -0.6 \ -0.4]$,

$C = [1 \ 0 \ 0]$, $\lambda\{\bar{A}-KC\} = [-1.5 \ -1 \ -0.75]$, $Epoch = 30$, $\Gamma_{initial} = 1$, $\Gamma_{final} = 0.1$. So, a total of 27 parameters are trained.

4.1 Simulation Results under Unrestricted Sampling Conditions

As mentioned before, in this case *ideal* conditions are considered, *i.e.* when an undetermined quantity of blood can be obtained from the patient in any desired time length. For this case, a total of 400min and a sampling time $T_s = 0.2[min]$ are taken. The model for a patient is identified, using the input signals (insulin and glucose) depicted in Fig. 1. As indicated before, the Fuzzy C-Means algorithm is run in order to obtain the initial membership functions, which are depicted in the same figure, as well as the convergence of the training algorithms, as well as the progression of the RMS error in normalized data. After training, the network is compared in parallel with the output obtained from the network.

4.2 Simulation Results under Restricted Conditions

Once the RNFN has been tested in unrestricted conditions of sampling time and testing time, more restrictive (and real) conditions must be considered. In this case, the same design matrices, number of rules and states are used, but now the sampling time is increased to $T_s = 5[min]$, and only 80 samples are to be taken (as laboratory resources are limited, as well as the availability of the equipment). As the effect of the sampling is expected to have an influence on the identification results, the data is pre-processed by a First-Order-Hold. The same simulation and training framework previously used are considered: first, the membership functions are obtained using a Fuzzy C-Means algorithm (shown in Fig. 1) using the input and output signals depicted, and then the training algorithm is run and the system and the network are compared in parallel. It can be seen that the parameters are still convergent and that the dynamics is captured by the network although some high-frequency information is lost because of the increased sampling time.

5 Conclusions

In this work we presented a black-box modeling of the insulin dynamics on different *in silico* patients using a continuous-time recurrent neural network with output feedback, where the input signals are considered as the rate of insulin (in $\mu U/dl/min$) applied to the patient, and the blood glucose concentration, and the output is insulin concentration ($\mu U/ml$) present in the blood stream. With a membership initialization based on the Fuzzy C-Means algorithm, and a learning law inspired from adaptive observer theory proven to be convergent in the parameters and stable in the Lyapunov sense even under model mismatch and low sampling rates. Simulation results were shown in order to validate the presented modeling, using only input-output data and practically no knowledge of the models of subsystems involved in the system. Future work includes control schemes based in the identified model, as well as analysis of the obtained model.

References

1. Bergman, R.N., Ider, Y.Z., Bowden, C.R., Cobelli, C.: Quantitative estimation of insulin sensitivity. *Am. J. Physiology, Endocrinology and Metabolism* 235, E667–E677 (1979)
2. Bolie, V.: Coefficients of normal blood glucose regulation. *Journal of Applied Physiology* 16, 783–788 (1961)
3. Sorensen, J.T.: A physiologic model of glucose metabolism in man and its use to design and assess improved insulin therapies for diabetes. Ph.D. dissertation, Massachusetts Institute of Technology. Dept. of Chemical Engineering (1985)
4. Wilinska, M., Chassin, L.J., Schaller, H., Schaupp, L., Pieber, T., Hovorka, R.: Insulin kinetics in type-1 diabetes: continuous and bolus delivery. *IEEE Transactions on Biomedical Engineering* 52(1), 3–12 (2005)
5. Man, C.D., Rizza, R., Cobelli, C.: Meal simulation model of the glucose-insulin system. *IEEE Transaction on Biomedical Engineering* 54, 1740–1749 (2007)
6. Narendra, K., Parthasarathy, K.: Identification and control of dynamical systems using neural networks. *IEEE Tras. Neural Networks* 1, 4–27 (1990)
7. Cybenko, G.: Approximation by superpositions of a sigmoidal function. *Math. Con. Signal Syst.* 2, 303–314 (1989)
8. Jang, J.-S.R.: Anfis: Adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man and Cybernetics* 23, 665–685 (1993)
9. Juang, C.-F.: A tsk-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms. *IEEE Transactions of Fuzzy Systems* 10(2), 155–170 (2002)
10. Hopfield, J.: Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proceedings of the National Academy of Sciences* 79(8), 2554–2558 (1982)
11. Gorrini, V., Bersini, H.: Recurrent fuzzy systems. In: *Proc. IEEE Int. Conference on Fuzzy Systems*, pp. 193–198 (1994)
12. Back, A.D., Tsoi, A.C.: A time series modeling methodology using fir and iir synapses. In: *Proc. Workshop on Neural Networks for Statistical and Economic Data*, pp. 187–194 (1990)
13. Mastorocostas, P.A., Theocharis, J.B.: A recurrent fuzzy-neural model for dynamic system identification. *IEEE Trans. onf Systems, Man and Cybernetics - Part B: Cybernetics* 32(2), 176–190 (2002)
14. Poznyak, A., Sanchez, E., Yu, W.: *Differential Neural Networks for Robust Non-linear Control: Identification, State Estimation and Trajectory Tracking*. World Scientific, Singapore (2001)
15. Gonzalez-Olvera, M.A., Tang, Y.: Identification and Control of Nonlinear Dynamic Systems via a Constrained Input-Output Neurofuzzy Network. In: Tao, G., Sun, J. (eds.) *Advances in Control Systems Theory and Applications* (2008)
16. Nelles, O.: *Nonlinear System Identification*. Springer, Heidelberg (2001)
17. Werbos, P.: Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE* 78(10), 1550–1560 (1990)
18. Williams, R.J., Zipser, D.: A learning algorithm for continually running fully recurrent neural networks. *Neural Computation* 2(1), 270–280 (1989)
19. Gonzalez-Olvera, M.A., Tang, Y.: Identification of a class of nonlinear systems by a continuous-time recurrent neurofuzzy network. In: *American Control Conference* (to be published, 2009)

20. Guyton, A., Hall, J.: *Textbook of Medical Physiology*. W. B. Saunders, Philadelphia (2000)
21. Association, A.D.: *Standards of Medical Care in Diabetes*. ADA (2009)
22. Gallardo-Hernandez, G., Fridman, L., Islas-Andrade, S., Shtessel, Y.: Quasi-continuous high order sliding modes controllers applied to glucose-insulin regulatory system models. In: *Proceedings of Decision and Control Conference*, December 9-11 (2008)
23. Kaveh, P., Shtessel, Y.: Blood glucose regulation using higher-order sliding mode control. *Int. J. of Robust and Nonlinear Control* 18, 557–569 (2007)
24. Hovorka, R., Canonico, V., Chassin, L., Haueter, U., Massi-Benedetti, M., Federici, M., Pieber, T., Schaller, H., Schaupp, L., Vering, T., Wilinska, M.: Nonlinear model predictive control of glucose concentration in subjects with type 1 diabetes. *Physiological Measurements* 25, 905–920 (2004)
25. Parker, R., Doyle III, F., Ward, J., Peppas, N.: Robust h_{∞} glucose control in diabetes using a physiological model. *American Institute of Chemical Engineers (AIChE)* 46(12), 2537–2549 (2004)
26. Reed Larsen, P., Kronenberg, H.M., Melmed, S., Polonsky, K.S.: *Williams Textbook of Endocrinology*. Saunders, USA (2003)
27. Marino, R.: Adaptive observers for single output nonlinear systems. *IEEE Transactions on Automatic Control* 35(9), 1054–1058 (1990)
28. Zhang, G., Patwo, B.E., Hu, M.Y.: Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting* 14, 35–62 (1998)
29. Zhang, Q.: Adaptive observer for multiple-input-multiple-output (MIMO) linear time-varying systems. *IEEE Transactions on Automatic Control* 47(3), 525–529 (2002)
30. Bastin, G., Gevers, M.: Stable adaptive observers for nonlinear time-varying systems. *IEEE Transactions on Automatic Control* 33(7), 650–658 (1988)
31. Dunn, J.: A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics* 3(3), 32–57 (1973)
32. Bezdek, J.: *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers, Norwell (1981)

Vague Query Based on Vague Relational Model

Faxin Zhao¹ and Z.M. Ma²

¹ School of Information Science and Technology, Zhanjiang Normal University
Zhanjiang, Guangdong 524048, China

zfx0405@163.com

² School of Information Science and Engineering, Northeastern University
Shenyang, Liaoning 110004, China

zongmin_ma@yahoo.com

Abstract. Although the relational model for databases provides a great range of advantages over other data models, it lacks a comprehensive way to handle imprecise and uncertain data. Vague set, as a generalized fuzzy set, has more powerful ability to process fuzzy information than fuzzy set. In this paper, based on the vague set, a kind of vague relational model is proposed. Incorporating the notion of vague sets in relations, we describe Vague SQL, which is an extension of SQL for the vague relational model, and show that Vague SQL combines the capabilities of a standard SQL with the power of manipulating vague relations. Although Vague SQL is a minimal extension to illustrate its usages, VSQL allows users to formulate a wide range of queries between vague data and queries.

Keywords: Vague set, Fuzzy set, Vague Relational Model, Vague SQL.

1 Introduction

In the real world, there are many imprecise and uncertain data values. Unfortunately, the relational model cannot represent the inherently imprecise and uncertain nature of the data. In order to handle such vagueness, Zadeh introduced fuzzy set theory in 1965 [1]. Based on the fuzzy set theory, various fuzzy relational database models, such as similarity-based relational model [2], possibility-based relational model [3] and some types of hybrid data models [4], have been proposed to model fuzzy information in relational database. Based on various fuzzy relational database models, many studies have been done for fuzzy query languages [5, 6] and data integrity constraints [3]. For a comprehensive review of what has been done in the development of the fuzzy relational databases, please refer to [7, 8].

In the fuzzy set theory, each object $u_i \in U$ is assigned a single value between 0 and 1, called the grade of membership, where U is a universe of discourse. As pointed out by Gau *et al.* in [9], the drawback of using the single membership value in the fuzzy set theory is that the evidence for $u_i \in U$ and the evidence against $u_i \in U$ are in fact mined together. They also pointed out that the single number reveals nothing about its accuracy. To tackle this problem, Gau *et al.* proposed the notion of vague set. A

vague set, as well as an Intuitionistic Fuzzy Set (IFS) [10], is a further generalization of a fuzzy set. But vague set is not isomorphic to the IFS, there are some interesting features for handling vague data that are unique to vague sets, such as vague sets allow for a more intuitive graphical representation of vague data, which facilitates significantly better analysis in data relationships, incompleteness, and similarity measures [11]. In [11], the notion of vague sets was initially incorporated into relations and a vague SQL (VSQL) was described. The VSQL allows the users to formulate a wide range of queries that occur in different modes of interaction between vague data and queries.

In this paper, we propose a vague relational database model which is an extension of the relational model. Based on the proposed model and the semantic measure of vague sets, we investigate vague querying strategies and give the form of vague querying with SQL.

The rest of this paper is organized as follows. In Section 2, we introduce some basic concepts and definitions about the vague set theory. In Section 3, we propose a vague relation model. In Section 4, vague data redundancies and removal are investigated. In Section 5, vague queries with SQL are discussed. Finally, the conclusions are given in Section 6.

2 Basic Knowledge

Let U be a universe of discourse, where an element of U is denoted by u .

Definition 1. A vague set V in U is characterized by a *truth-membership* function t_V and a *false-membership* function f_V . Here $t_V(u)$ is a lower bound on the grade of membership of u derived from the evidence for u , and $f_V(u)$ is a lower bound on the negation of u derived from the evidence against u . $t_V(u)$ and $f_V(u)$ are both associated with a real number in the interval $[0, 1]$ with each element in U , where $t_V(u) + f_V(u) \leq 1$.

That is

$$t_V: U \rightarrow [0, 1] \text{ and } f_V: U \rightarrow [0, 1]$$

Suppose $U = \{u_1, u_2, \dots, u_n\}$. A vague set V of the universe of discourse U can be represented by

$$V = \sum_{i=1}^n [t_V(u_i), 1 - f_V(u_i)] / u_i, \forall u_i \in U, \text{ where } t_V(u) \leq \mu_V(u) \leq 1 - f_V(u) \text{ and } 1 \leq i \leq n.$$

This approach bounds the grade of membership of u to a subinterval $[t_V(u), 1 - f_V(u)]$ of $[0, 1]$. In other words, the exact grade of membership $\mu_V(u)$ of u may be unknown, but is bounded by $t_V(u) \leq \mu_V(u) \leq 1 - f_V(u)$, where $t_V(u) + f_V(u) \leq 1$. We depict these ideas in Fig. 1.

For a vague value $[t_V(u), 1 - f_V(u)]/u$, the vague value to the object u is the interval $[t_V(u), 1 - f_V(u)]$. For example, if $[t_V(u), 1 - f_V(u)] = [0.5, 0.8]$, then we can say that $t_V(u) = 0.5$, $1 - f_V(u) = 0.8$ and $f_V(u) = 0.2$. It can be interpreted as “the degree that object u belongs to the vague set V is 0.5; the degree that the object u does not belong to the vague set V is 0.2”. In a voting process, the vague value $[0.5, 0.8]$ can be interpreted as “the vote for resolution is 5 in favor, 2 against, and 3 neutral (abstentions).”

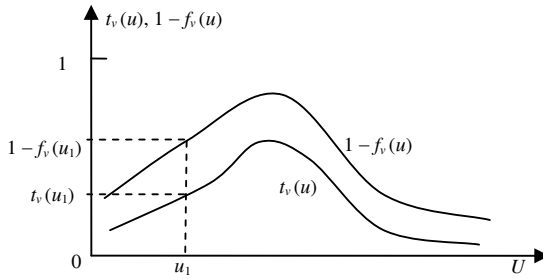


Fig. 1. A vague set

The precision of the knowledge about u is characterized by the difference $(1 - t_V(u) - f_V(u))$. If this is small, the knowledge about u is relatively precise. But if it is large, we know correspondingly little. If $t_V(u)$ is equal to $(1 - f_V(u))$, the knowledge about u is exact, and the vague set theory reverts back to fuzzy set theory. If $t_V(u)$ and $(1 - f_V(u))$ are both equal to 1 or 0, depending on whether u does or does not belong to V , the knowledge about u is very exact and the theory reverts back to ordinary sets. For example, the fuzzy set $0.6/u$ can be presented as the vague set $[0.6, 0.6]/u$, while the ordinary set $\{u\}$ can be presented as the vague set $[1, 1]/u$.

In the following, we present several special vague sets and operations on vague sets.

Definition 2. A vague set V is an empty vague set if and only if its truth-membership function $t_V = 0$ and false-membership function $f_V = 1$ for all u on U . We use \emptyset to denote it.

Definition 3. The complement of a vague set V , denoted V' , is defined by $t_{V'}(u) = f_V(u)$ and $1 - f_{V'}(u) = 1 - t_V(u)$.

Definition 4. A vague set A is contained in another vague set B , written as $A \subseteq B$, if and only if $t_A \leq t_B$ and $1 - f_A \leq 1 - f_B$.

Definition 5. Two vague sets A and B are equal, written as $A = B$, if and only if $A \subseteq B$ and $B \subseteq A$, namely, $t_A = t_B$, and $1 - f_A = 1 - f_B$.

Definition 6. (Union) The union of two vague sets A and B is a vague set C , written as $A \cup B$, whose truth-membership and false-membership functions are related to those of A and B by $t_C = \max(t_A, t_B)$ and $1 - f_C = \max(1 - f_A, 1 - f_B) = 1 - \min(f_A, f_B)$.

Definition 7. (Intersection) The intersection of two vague sets A and B is a vague set C , written as $C = A \cap B$, whose truth-membership and false-membership functions are related to those of A and B by $t_C = \min(t_A, t_B)$ and $1 - f_C = \min(1 - f_A, 1 - f_B) = 1 - \max(f_A, f_B)$.

3 Vague Relational Model

In the relational database model, a relation schema R is denoted by $R(A_1, A_2, \dots, A_n)$ or simply by R , in which each attribute A_i ($1 \leq i \leq n$) is associated with a domain, written as $\text{Dom}(A_i)$. A relation instance on relation schema $R(A_1, A_2, \dots, A_n)$, represented as $r(R)$ or r , is defined as a subset of $\text{Dom}(A_1) \times \text{Dom}(A_2) \times \dots \times \text{Dom}(A_n)$, which is a set of n -tuples and represented as $r = \{t_1, t_2, \dots, t_m\}$. It is clear that $r \subseteq \text{Dom}(A_1) \times \text{Dom}(A_2) \times \dots \times \text{Dom}(A_n)$. A tuple t_j ($1 \leq j \leq m$) can be expressed as $t_j = \langle v_{j1}, v_{j2}, \dots, v_{jn} \rangle$, where $v_{ji} \in \text{Dom}(A_i)$ ($1 \leq i \leq n$). Then $t_j \in \text{Dom}(A_1) \times \text{Dom}(A_2) \times \dots \times \text{Dom}(A_n)$.

In order to incorporate fuzzy information into relational databases, various attempts towards enhancing the relational database model by fuzzy extensions can be found in the literature [2-4]. In this section, we describe an approach of enhancing the relational model by means of vague set theory, which results in the vague relational database (VRDB) model. To this purpose, we should extend $\text{Dom}(A_i)$ as a set of vague sets and a vague relational instance is hereby defined as a subset of the Cartesian product of such attribute domains.

Definition 8. Let A_i for i from 1 to n be attributes defined on the universes of discourse sets U_i , respectively. Then a vague relation r is defined on the relation schema $R(A_1, A_2, \dots, A_n)$ as a subset of the Cartesian product of a collection of vague subsets:

$$r \subseteq V(U_1) \times V(U_2) \times \dots \times V(U_n),$$

where $V(U_i)$ denotes the collection of all vague subsets on a universe of discourse U_i .

Each tuple t of r consists of a Cartesian product of vague subsets on the respective U_i 's, i.e., $t[A_i] = \pi(A_i)$ where $\pi(A_i)$ is a vague subset of the attribute A_i defined on U_i for all i . The relation r can thus be represented by a tableau with n columns as show in Table 1.

Table 1. A vague relation r

A_1	A_2	...	A_n
$\pi_{11}(A_1)$	$\pi_{12}(A_2)$...	$\pi_{1n}(A_n)$
$\pi_{21}(A_1)$	$\pi_{22}(A_2)$...	$\pi_{2n}(A_n)$
...
$\pi_{m1}(A_1)$	$\pi_{m2}(A_2)$...	$\pi_{mn}(A_n)$

It should be noted that the vague relations can be considered as an extension of classical relations (all vague values are $[1, 1]$) and possibilistic relations (all vague values are possibility distributions, i.e. such that each degree is $[a, a]$, $0 \leq a \leq 1$), which can capture more information about vagueness.

4 Vague Data Redundancies and Removal

4.1 Similarity Measure of Vague Data

There have been some studies which discuss the topic concerning how to measure the degree of similarity between vague sets [12-14]. In [14], it was pointed out that the similarity measures proposed in [12, 13] did not fit well in some cases. They proposed a set of modified measures which turned out to be more reasonable in more general cases than the Chen's.

Definition 9. Let x and y be two vague values such that $x = [t_x, 1 - f_x]$, $y = [t_y, 1 - f_y]$, where, $0 \leq t_x \leq 1 - f_x \leq 1$, and $0 \leq t_y \leq 1 - f_y \leq 1$. Let $SE(x, y)$ denote the similarity measure between x and y . Then:

$$SE(x, y) = 1 - \frac{|(t_x - t_y)| + |(f_x - f_y)|}{2}$$

Definition 10. Let $U = \{u_1, u_2, \dots, u_n\}$ be the universe of discourse. Let A and B be two vague sets on U , where:

$$A = \sum_{i=1}^n [t_A(u_i), 1 - f_A(u_i)] / u_i, \forall u_i \in U, \text{ where } t_A(u) \leq \mu_A(u) \leq 1 - f_A(u) \text{ and } 1 \leq i \leq n.$$

$$B = \sum_{i=1}^n [t_B(u_i), 1 - f_B(u_i)] / u_i, \forall u_i \in U, \text{ where } t_B(u) \leq \mu_B(u) \leq 1 - f_B(u) \text{ and } 1 \leq i \leq n.$$

Let $SE(A, B)$ denote the similarity measure between A and B . Then:

$$\begin{aligned} SE(A, B) &= \frac{1}{n} \sum_{i=1}^n SE([t_A(u_i), 1 - f_A(u_i)], [t_B(u_i), 1 - f_B(u_i)]) \\ &= \frac{1}{n} \sum_{i=1}^n \left(1 - \frac{|(t_A(u_i) - t_B(u_i))| + |(f_A(u_i) - f_B(u_i))|}{2} \right) \end{aligned}$$

The notion of similarity measure between the attribute values represented by vague sets can be extended to the similarity measure between two tuples. Let $t_i = \langle a_{i1}, a_{i2}, \dots, a_{in} \rangle$ and $t_j = \langle a_{j1}, a_{j2}, \dots, a_{jn} \rangle$ be two vague tuples in vague relational instance r over schema $R(A_1, A_2, \dots, A_n)$. Let $SE(t_i, t_j)$ denote the similarity measure between t_i and t_j . Then

$$SE(t_i, t_j) = \min \{ SE(t_i[A_1], t_j[A_1]), SE(t_i[A_2], t_j[A_2]), \dots, SE(t_i[A_n], t_j[A_n]) \}$$

4.2 Data Redundancies

Following the similarity measure between vague sets, equivalence redundancy can be evaluated. Being different from the classical set theory, the condition of $SE(A, B) = 1$ is essentially the special case of vague sets due to the vagueness of the data. In general, a threshold which is should be considered when evaluating the similarity measure between two vague sets.

Definition 11. Let A and B be two vague sets and α be a threshold. If $SE(A, B) \geq \alpha$, it is said that A and B are equivalently α -redundant.

If A and B are equivalently redundant, the removal of redundancy can be achieved by merging A and B and producing a new vague set C .

Assume that the vague sets A and B are equivalently α -redundant. The elimination of duplicate could be achieved by merging A and B and producing a new vague data C , where A , B , and C are three vague sets on $U = \{u_1, u_2, \dots, u_n\}$. Then the following three merging operations are defined:

$$\begin{aligned}
 \text{(a) } C = A \cup_v B &= \{[t_C(w), 1 - f_C(w)] / w \mid (\exists([t_A(u_i), 1 - f_A(u_i)] / u_i))(\exists([t_B(u_i), 1 - f_B(u_i)] / u_i))(t_C(w))\} \\
 &= \max(t_A(u_i), t_B(u_i)) \wedge 1 - f_C(w) \\
 &= \max(1 - f_A(u_i), 1 - f_B(u_i)) \wedge (w = u_i) \wedge u_i \in U \wedge 1 \leq i \leq n \\
 \text{(b) } C = A \cap_v B &= \{[t_C(w), 1 - f_C(w)] / w \mid (\exists([t_A(u_i), 1 - f_A(u_i)] / u_i))(\exists([t_B(u_i), 1 - f_B(u_i)] / u_i))(t_C(w))\} \\
 &= \min(t_A(u_i), t_B(u_i)) \wedge 1 - f_C(w) \\
 &= \min(1 - f_A(u_i), 1 - f_B(u_i)) \wedge (w = u_i) \wedge u_i \in U \wedge 1 \leq i \leq n \\
 \text{(c) } C = A -_v B &= \{[t_C(w), 1 - f_C(w)] / w \mid (\exists([t_A(u_i), 1 - f_A(u_i)] / u_i))(\exists([t_B(u_i), 1 - f_B(u_i)] / u_i))t_C(w)\} \\
 &= \max(0, t_A(u_i) - (1 - f_B(u_i))) \vee 1 - f_C(w) \\
 &= \max(0, 1 - f_A(u_i) - t_B(u_i)) \wedge u_i \in U \wedge 1 \leq i \leq n
 \end{aligned}$$

The processing of vague set redundancy can be extended to that of vague tuple redundancy.

Definition 12. Let r be a vague relation on the relational schema $R(A_1, A_2, \dots, A_n)$. Let $t = \langle \pi(A_1), \pi(A_2), \dots, \pi(A_n) \rangle$ and $t' = \langle \pi'(A_1), \pi'(A_2), \dots, \pi'(A_n) \rangle$ be two vague tuples in r . let $\alpha \in [0, 1]$ be a threshold. The vague tuples t and t' are equivalently α -redundant if and only if $\min(SE(t[A_i], t'[A_i]) \geq \alpha$ holds ($1 \leq i \leq n$).

5 Vague Querying with SQL

In relational databases, query processing refers to such procedure that the tuples satisfying a given selection condition are selected, and then the query results with the required formats are delivered to the user. These format requirements include which attributes appear in the result relation, and if the result relation is ordered and grouped over the given attribute(s). Therefore, a query can be seen as comprising two components, namely, a Boolean selection condition and some format requirements. For the sake of simple illustration, some format requirements are ignored in the following discussion. Then utilizing a standard relational query language, i.e., SQL, a basic query expression in SQL is given as follows:

```
SELECT <attribute list> FROM <relation name> WHERE <selection condition>
```

Here the select-clause specifies the target (output) attributes to be returned, which are separated by commas: Attribute₁, Attribute₂, ..., Attribute_n. At least one attribute name must be specified in <attribute list>. The from-clause specifies the relations involved in the query and the where-clause specifies the conditions to be satisfied by the result of the query.

Classical relation databases have been widely used, but it is still suffer from a lack of query flexibility. The given selection condition and the contents of the relation are all crisp. In this context, a tuple will either definitely or definitely not satisfy the condition. In imperfect relational databases, however, a tuple may satisfy the selection condition with a certain possibility and a certain necessity degree even if the condition is crisp due to the fact that tuples are imperfect. On the other hand, imprecise information may exist in the selection condition, and the situation that a tuple may more or less satisfy the selection condition may also occur. For classical relational databases, however, a query with an imprecise selection condition is also useful in order to satisfy the requirement of decision making. Classical query processing is obviously too rigid for two cases. In general, a flexible query should satisfy the following two conditions [15]:

(a) A qualitative distinction between the selected tuples is allowed.

(b) Imprecise conditions inside queries are introduced when the user cannot define his/her needs in a definite way, or when a pre-specified number of responses are desired and therefore a margin is allowed to interpret the query.

Typically, the case in (a) occurs when the queried relational databases contain imperfect information and the query conditions are crisp. For example, in a relation of student design, it is supposed that there are four students John, Jack, Mary, and Rose whose age are respectively interval values [13, 15], [14, 16], [17, 19], and [18, 20]. Now let us query this relation with the condition “age = 14.” Then John and Jack may satisfy the condition in some degrees, whereas Mary and Rose must not satisfy the condition. The case (b) typically occurs when the query conditions are imprecise even if the queried relational databases do not contain incomplete information. For example, there are four students Hans, Mary, John, and Rose whose age are respectively crisp values 15, 16, 18, and 19. Now let us query this relation with the condition with the condition “age is around 16.” Then the order that the four students satisfy the condition will be Mary, Hans, John, and Rose. It can be seen that flexible queries permit users to provide incomplete query conditions and the query results include the tuples satisfying the conditions definitely as well as the tuples satisfying the conditions partially.

For vague relational databases, a basic query condition with form of $A \theta B$ may consist of vague comparison operations \succ_{α} , \prec_{α} , \succeq_{α} , \preceq_{α} , \approx_{α} , and \neq_{α} for “ θ ” and vague constants for “ B ”, where α is a threshold. Then under a given threshold, a tuple either satisfies the condition if the condition is true or does not satisfy the condition if the condition is false. Since the basic conditions are only evaluated true or false, complex query conditions comprised of basic conditions and logical operations can be evaluated true or false by means of 2VL. So based on the given threshold, all tuples in vague relation are divided into two parts for a query, namely, “answer part” and “no answer part”. Compared with the queries for relational databases containing partial values and null values [16], the query answers for vague relational databases are only sure answers. There are no possible answers to be included. The reason that this situation occurs is because of the usage of threshold. The basic conditions with which evaluations of similarity measure are less than the threshold are regarded as false. Therefore, depending on the different thresholds that are values in [0, 1], the same query for the same vague relation may have different query answers. The queries for

vague relational databases are concerned with the number choices of threshold. Therefore, the following syntax of SQL is used to represent the select sentence:

```
SELECT <attribute list> FROM <relation name>
WHERE <selection condition> [WITH SIMILARITY <threshold>]
```

Where select-clause and from-clause are the same as the SQL in classical relational, <selection condition> is a vague select condition, and <threshold> is a crisp threshold in [0, 1]. Utilizing such SQL, one can get such tuples that satisfy the given select condition and the given threshold. Note that the item WITH SIMILARITY <threshold> can be omitted. The default of <threshold> is exactly 1 then.

Here, a problem exists in vague queries, i.e., the strength of answers to the queries is unknown. Such information is useful sometimes.

Table 2. Vague relation s

Product ID	Price
9801	[0.3,0.4]/240+[0.8,0.9]/260+[0.9,0.9]/280
9802	[0.7,0.8]/260+[0.9,1.0]/280+[0.4,0.5]/300

For example, a vague relation is shown in Table 2. Consider the following query:

```
SELECT ID FROM r
WHERE Price ≈ [0.8, 0.8]/260 + [0.9, 1.0]/280 + [0.2, 0.3]/300 WITH
SIMILARITY 0.8.
```

Then

$$SE ([0.3, 0.4]/240 + [0.8, 0.9]/260 + [0.9, 0.9]/280, [0.8, 0.8]/260+[0.9, 1.0]/280+[0.2, 0.3]/300)=0.83>0.8$$

$$SE ({[0.7, 0.8]/260 + [0.9, 1.0]/280 + [0.4, 0.5]/300}, {[0.8, 0.8]/260+[0.9, 1.0]/280+[0.2, 0.3]/300})=0.9>0.8$$

So products “9801” and “9802” all satisfy the condition and become the query answers. It is clear that product “9802” satisfies the condition better than product “9801”. However, in query answers, they cannot be identified with respect to the satisfaction degrees.

In order to estimate the strength of query answers, their restrictions are relaxed so that the basic query conditions are not only evaluated true or false according to the threshold. Here a multivalued logic (MVL) system is introduced, in which each logical value is number in [0, 1]. Based on the MVL, two basic query conditions $X = Y$ and $X > Y$ are defined as follows:

(a) $X = Y$ is evaluated to be a logical value $\mu (X = Y) = SE (X, Y)$

(b) $X > Y$ is evaluated to be a logical value $\mu (X > Y)$, where

$$\mu (X, Y) = 0, \text{ if } \min (\text{supp} (X)) \leq \min (\text{supp} (Y)) \text{ and } \max (\text{supp} (X)) \leq \max (\text{supp} (Y))$$

$$\mu (X, Y) = 1 - SE (X, Y), \text{ otherwise}$$

The logical operations AND, OR, and NOT under MVL are defined as $\min ()$, $\max ()$, and complementation, respectively. Let μ_1 and μ_2 be logical values under MVL. It is clear that μ_1 and μ_2 are all in [0, 1]. Then

- (a) $\mu_1 \text{ AND } \mu_2 = \min(\mu_1, \mu_2)$
- (b) $\mu_1 \text{ OR } \mu_2 = \max(\mu_1, \mu_2)$
- (c) $\text{NOT } \mu_1 = 1 - \mu_1$.

For a query with complex query condition, the logical value that each tuple in the vague relation satisfies the condition can be evaluated according to the comparison operations and logical operations under MVL. This logical value can be seen as the strength that the tuple matches the condition. In general, the tuples whose strength is zero are ignored. Only the tuples whose strengths are larger than zero are permitted in query answers, and these tuples can be ranked by the descent of the strengths. When a threshold is given, the tuples that strengths are less than the threshold can be cut. Since the tuples in query answers are connected with the strengths matching the query, one can choose the tuple with the maximum strength from the query answers. After introducing MVL, the query answers can be measured through the matching strengths.

In the example of Table 2, the matching strengths of product “9801” and “9802” to the given condition are 0.83 and 0.9, respectively. Such results demonstrate that product “9802” satisfies the condition better than product “9801”.

6 Conclusions

This paper concentrated on vague query in the vague relational databases. For this purpose, we proposed an extended relational database model, with an objective to provide a generalized approach for coping with imprecise data. In order to retrieve vague data by the well-established database query language, we incorporate the notion of vagueness into the relational data model and demonstrate how VSQL can be employed to formulate a wide range of queries.

Acknowledgments. The work is supported by the National Research Foundation for the Doctoral Program of Higher Education of China under Grant No.20050145024.

References

1. Zadeh, L.A.: Fuzzy sets. *Information and Control* 8, 338–353 (1965)
2. Buckles, B.P., Petry, F.E.: A Fuzzy Representation of Data for Relational Databases. *Fuzzy Sets and Systems* 7, 213–226 (1982)
3. Raju, K.V.S.V.N., Majumdar, A.K.: Fuzzy Functional Dependencies and Lossless Join Decomposition of Fuzzy Relational Database Systems. *ACM Transactions on Database Systems* 13, 129–166 (1988)
4. Ma, Z.M., Mili, F.: Handling Fuzzy Information in Extended Possibility-Based Fuzzy Relational Databases. *International Journal of Intelligent Systems* 17, 925–942 (2002)
5. Bosc, P., Pivert, O.: SQLf: A Relational Database Language for Fuzzy Querying. *IEEE Transactions on Fuzzy Systems* 3, 1–17 (1995)
6. Takahashi, Y.: Fuzzy Database Query Languages and Their Relational Completeness Theorem. *IEEE Transactions on Knowledge and Data Engineering* 5, 122–125 (1993)
7. Petry, F.E.: *Fuzzy Databases: Principles and Applications*. Kluwer Academic Publisher, Dordrecht (1996)

8. Ma, Z.M.: *Fuzzy Database Modeling with XML*. Springer, Heidelberg (2005)
9. Gau, W.L., Buehrer, D.J.: Vague Sets. *IEEE trans. Syst., Man, Cybern.* 23, 610–614 (1993)
10. Atanassov, K.: Intuitionistic fuzzy sets. *Fuzzy Sets and Systems* 20, 87–96 (1986)
11. Lu, A., Ng, W.: Vague sets or intuitionistic fuzzy sets for handling vague data: Which one is better? In: Delcambre, L.M.L., Kop, C., Mayr, H.C., Mylopoulos, J., Pastor, Ó. (eds.) *ER 2005. LNCS*, vol. 3716, pp. 401–416. Springer, Heidelberg (2005)
12. Chen, S.M.: Measures of Similarity between Vague Sets. *Fuzzy Sets and Systems* 74, 217–223 (1995)
13. Chen, S.M.: Similarity Measure between Vague Sets and between Elements. *IEEE Trans. Syst. Man, Cybern.* 27, 153–158 (1997)
14. Hong, D.H., Kim, C.: A Note on Similarity Measures between Vague Sets and between Elements. *Information Sciences* 115, 83–96 (1999)
15. Bosc, P., Pivert, O.: Some Approaches for Relational Databases Flexible Querying. *Journal of Intelligent Information Systems* 1, 323–354 (1992)
16. Lipski, W.: On Semantic Issues Connected with Incomplete Information Databases. *ACM Transactions on Database Systems* 4, 262–296 (1979)

Identification and Speed Control of a DC Motor Using an Input-Output Recurrent Neurofuzzy Network

Edgar Berrospe*, Marcos Angel González-Olvera**, and Yu Tang

Universidad Nacional Autónoma de México (UNAM). Facultad de Ingeniería
Tel.: (52)-55-56223013

Abstract. In this work an input-output recurrent neurofuzzy network is used to identify and control DC motor. The known data are input-output signals obtained directly from measurements of the system. The structure of the network is linear in the consequent parameters and nonlinear in the antecedent ones. The linearization of the antecedent parameters around a suboptimal value is used to get a linear parametrization, and then a Kalman filter is applied. Nonlinear constraints built into the structure are proposed, and a new parameter initialization algorithm is presented. A certainty equivalence control scheme with online adaptation is presented, based on this model.

Keywords: Neurofuzzy Network, System Identification, Fuzzy Systems, Certain Equivalence Control, Neural Networks Model.

1 Introduction

The neurofuzzy networks use the ability of fuzzy systems to emulate the human fuzzy way of thinking and the learning capabilities of the neural networks [4]. Neural networks and fuzzy systems are, as noted by Cybenko [11], universal approximators. This means that they can approximate any continuous smooth function with an arbitrary degree of accuracy in a compact set. For this reason, they have been used modeling and designing control systems. Recurrent neurofuzzy networks are networks that contain feedback connections in some or all layers within the network. These networks are useful when modeling and identifying dynamic systems.

Training a neurofuzzy network is also a challenging task. The lumpiness in convergence and the curse of dimensionality are challenges to be faced when training processes. Simpler structures, designed for specific task, have been proposed in order to deal with these problems [4].

The network proposed in [2] and [3] has proven to be useful identifying a typical benchmark non-linear system [12]. In this work we carry out the experimental

* The work of Edgar Berrospe is supported by UNAM-PAPITT IN112908 and UNAM-PAPITT IN20009.

** The work of Marcos A. Olvera-González is supported by CONACYT.

validation of the *input-output recurrent neurofuzzy network* by identifying and controlling an experimental DC motor with associated nonlinearities. This paper is organized as following: in Section 2 the problem of the identification of a SISO system is stated. In Section 3 the structure and training of the network is presented, as well as the control of the system. The experimental results are shown in Section 4. Finally, the conclusions are in Section 5.

2 System Identification

Take the following SISO system to be identified with input u_k and output y_k , modeled by the difference equation

$$y_{k+1} = f_d(y_k, \dots, y_{k-n+1}, u_k, \dots, u_{k-m+1}). \quad (1)$$

The unknown function f_d is continuous. The following assumptions are made for this system:

- The system is stable in the sense that any bounded input will produce a bounded output.
- The orders n and m of time-delays of the output and input, respectively, are assumed known a priori.
- y_k and u_k , for $k = 0, \dots, N$ are available.

The goal is to find an input-output recurrent neurofuzzy network with parameters θ

$$\hat{y}_{k+1} = f(\hat{y}_k, \dots, \hat{y}_{k-n+1}, u_k, \dots, u_{k-m+1}, \theta), \quad (2)$$

so that

$$\sup_k |\hat{y}_k(\cdot) - y_k(\cdot)| < \varepsilon, \quad (3)$$

where $\varepsilon > 0$ is as small as possible [1].

3 Input-Output Recurrent Neurofuzzy Network

3.1 Structure

The Input-Output Recurrent Neurofuzzy Network (IORNN) is a nonlinear model of the system. Each fuzzy rule contains a linear model of the system which identifies the system in some region. The IORNN with n_R rules has the following structure

R_i : If \hat{y}_k is A_i then

$$\begin{aligned} \hat{y}_{k+1} &= [c_{i1}, \dots, c_{in}] \begin{bmatrix} \hat{y}_k \\ \vdots \\ \hat{y}_{k-n+1} \end{bmatrix} + [h_{i1}, \dots, h_{im}] \begin{bmatrix} u_k \\ \vdots \\ u_{k-m+1} \end{bmatrix}, \\ &= c_i^T z_k + h_i^T \xi_k, \end{aligned} \quad (4)$$

where $c_i = [c_{i1}, \dots, c_{in}]$, $z_k = [\hat{y}_k, \dots, \hat{y}_{k-n+1}]$, $h_i = [h_{i1}, \dots, h_{in}]$ and $\xi_k = [u_k, \dots, u_{k-m+1}]$.

The membership functions are $A_i(x) = \mu_i = e^{-\sigma_i^2(x-\varsigma_i)^2}$. The defuzzification is made by a weighted average, so

$$\varphi_k^i = \frac{A_i(\hat{y}_k)}{\sum_{j=i}^{n_R} A_j(\hat{y}_k)}, \quad (5)$$

and the network can be described by the following equation

$$\begin{aligned} \hat{y}_{k+1} &= (\varphi_k^1 c_1^T + \dots + \varphi_k^{n_R} c_{n_R}^T) z_k + (\varphi_k^1 h_1^T + \dots + \varphi_k^{n_R} h_{n_R}^T) \xi_k, \\ &= (\varphi_k^T \otimes z_k^T) \text{vect}(C^T) + (\varphi_k^T \otimes \xi_k^T) \text{vect}(H^T), \end{aligned} \quad (6)$$

where \otimes is the Kronecker product, $\text{vect}(\cdot)$ is the row vectorization into a single column of the argument matrix, $C = [c_1^T, \dots, c_{n_R}^T]^T$, $H = [h_1^T, \dots, h_{n_R}^T]^T$ and $\varphi_k = [\varphi_k^1, \dots, \varphi_k^{n_R}]^T$. The parameters to be identified are: $\sigma = [\sigma_1, \dots, \sigma_{n_R}]^T$, $\varsigma = [\varsigma_1, \dots, \varsigma_{n_R}]^T$, C and H .

The approximation to the real system with output signal y_{k+1} is given by

$$\begin{aligned} y_{k+1} &= f_1(C, \sigma, \varsigma, z_k, \xi_k) + f_2(H, \sigma, \varsigma, z_k, \xi_k) + \bar{\epsilon}_k, \\ &= f(C, H, \sigma, \varsigma, z_k, \xi_k) + \bar{\epsilon}_k, \end{aligned} \quad (7)$$

The approximation error $\bar{\epsilon}_k$ will be minimum ($\epsilon_k = \min_{\theta} \bar{\epsilon}_k$) if the optimal values for C , H , σ y ς are found, then

$$y_{k+1} = f(C^*, H^*, \sigma^*, \varsigma^*, z_k, \xi_k) + \epsilon_k. \quad (8)$$

3.2 Linearization of the Antecedent Parameters

In order to train the antecedent parameters of the network σ and ς , we propose to obtain a linearization, so linear identification methods can be used. If the network is near a local optimum for the antecedent parameters, then a linearization will be taken place around the local optimum. For this purpose, an initialization algorithm will be discussed in Section 3.4. We propose to expand the optimal-valued network (that is, with the parameters $\theta = \theta^*$), around a suboptimal value θ . So, the function $f = f(\theta^*)$ defining the optimal network will be expressed, around its parameters θ , as.

$$f(\theta^*) = f(\theta) + (\theta^* - \theta)^T \left. \frac{\partial f}{\partial \theta^*} \right|_{\theta^* = \theta} + R(\theta, \theta^*), \quad (9)$$

where $\theta = [\text{vect}(C^T), \text{vect}(H^T), \sigma, \varsigma]^T$ and $R(\theta, \theta^*)$ represents the higher-order terms in the expansion. Given that direct measurements of the output, it can be

considered for training that $z_k = [y_k, \dots, y_{k-n+1}]^T$. Given that f is linear in the consequent parameters C, H , the expansion can be written as

$$f(\theta^*, z_k, \xi_k) = f(C, H, \sigma, \varsigma, z_k, \xi_k) + f(C^* - C, H^* - H, \sigma, \varsigma, z_k, \xi_k) + (\sigma^* - \sigma)^T \left. \frac{\partial f}{\partial \sigma^*} \right|_{\theta} + (\varsigma^* - \varsigma)^T \left. \frac{\partial f}{\partial \varsigma^*} \right|_{\theta} + R(\theta, \theta^*). \tag{10}$$

Given that $y_{k+1} = f(C^*, H^*, \sigma^*, \varsigma^*, z_k, \xi_k) + e_k$ and $\hat{y}_{k+1} = f(C, H, \sigma, \varsigma, z_k, \xi_k)$, subtracting both equations $\tilde{y}_{k+1} = y_{k+1} - \hat{y}_{k+1}$

$$\begin{aligned} \tilde{y}_{k+1} &= (\sigma^* - \sigma)^T g_k + (\varsigma^* - \varsigma)^T h_k + (\varphi_k^T \otimes z_k^T) \text{vect}((C^* - C)^T) \\ &\quad + (\varphi_k^T \otimes \xi_k^T) \text{vect}((H^* - H)^T) + \vartheta_k, \\ &= \tilde{\sigma}^T g_k + \tilde{\varsigma}^T h_k + (\varphi_k^T \otimes z_k^T) \text{vect}(\tilde{C}^T) + (\varphi_k^T \otimes \xi_k^T) \text{vect}(\tilde{H}^T) + \vartheta_k, \end{aligned} \tag{11}$$

where $\vartheta_k = R + \epsilon_k$, $\rho_k = \left. \frac{\partial f}{\partial \sigma^*} \right|_{\theta}$, $\varrho_k = \left. \frac{\partial f}{\partial \varsigma^*} \right|_{\theta}$, $\tilde{C} = C^* - C$, $\tilde{H} = H^* - H$, $\tilde{\sigma} = \sigma^* - \sigma$ and $\tilde{\varsigma} = \varsigma^* - \varsigma$.

The equations (12) and (13) express the calculus of the partial derivatives of f_1 with respect to the parameters σ . The partial derivatives with respect to the parameters ς can be found in the same way.

$$\left. \frac{\partial f_1}{\partial \sigma^*} \right|_{\sigma} = \frac{\partial \varphi^T}{\partial \sigma} C z(k) \tag{12}$$

$$\frac{\partial \varphi^T}{\partial \sigma} = 2 \begin{bmatrix} \frac{\sigma_1(y_k - \varsigma_1)^2}{\sum_j \mu_j} \mu_1 \\ \vdots \\ \frac{\sigma_n(y_k - \varsigma_n)^2}{\sum_j \mu_j} \mu_n \end{bmatrix} \otimes [\mu_1 \cdots \mu_n] - 2 \text{diag} \left(\left[\frac{\sigma_1(y_k - \varsigma_1)^2}{\sum_l \mu_l} \mu_1 \cdots \frac{\sigma_n(y_k - \varsigma_n)^2}{\sum_l \mu_l} \mu_n \right] \right) \tag{13}$$

Rewriting the equations, the following linearized model respect to the parameters is obtained

$$\tilde{y}_{k+1} = G_k \tilde{\theta} + \vartheta_k = G_k \tilde{\theta}_k + \vartheta_k, \tag{14}$$

where $\tilde{\theta} = \theta^* - \theta$ and $G_k = [\varphi_k^T \otimes z_k, \varphi_k^T \otimes \xi_k^T, \rho_k, \varrho_k]^T$.

3.3 Nonlinear Constraints for the Parameters

It is desirable to avoid that the parameters ς take values such that the fuzzy sets would be out of the operation region of the system. Also, for reasons that will be discussed in Section 3.7, the parameters of the first column of the matrix H are to be kept with the same sign. We propose to implement nonlinear constraints to the parameters which may take undesired values. The constrains proposed here are similar to those of proposed in [3].

The constrains are implemented using nonlinear sigmoid functions ϕ . The constrained parameter x is defines as $x = \phi(\check{x})$. The functions ϕ have the following characteristics:

- $\phi(\tilde{x}) \in (a, b), \quad a \in \mathfrak{R}, b \in \mathfrak{R}$
- $\phi(\hat{x}) = \hat{x}$
- $\lim_{\tilde{x} \rightarrow \infty} \phi(\tilde{x}) = a$
- $\lim_{\tilde{x} \rightarrow -\infty} \phi(\tilde{x}) = b$
- $0 \leq \frac{\partial \phi}{\partial \tilde{x}} \leq 1$

where a is the superior bound, b the inferior one, and \hat{x} is the initial condition of the constrained parameter

Once the constrains are implemented the network equations become:

$$\begin{aligned}
 y_{k+1} &= f_1(C, \sigma, \zeta, z_k) + f_2(\bar{H}, \check{h}_{c1}, \sigma, \zeta, \bar{\xi}_k, u_k) + \bar{\epsilon}_k, \\
 &= f(C, \bar{H}, \check{h}_{c1}, \sigma, \zeta, z_k, \bar{\xi}_k, u_k) + \bar{\epsilon}_k,
 \end{aligned}
 \tag{15}$$

where \bar{H} is H without its first column $\bar{H} = [h_{c2}, h_{c3}, \dots, h_{cn_R}]$, $\bar{\xi}$ is ξ without the its first element $\bar{\xi} = [u_{k-1}, \dots, u_{k-m+1}]^T$, $h_{c1} = \phi_h(\check{h}_{c1})$ is the first column of H and $\varsigma = \phi_\varsigma(\zeta)$.

By expanding in Taylor series we get

$$\begin{aligned}
 f(\theta^*, z_k, \xi_k) &= f(C, H, \sigma, \zeta, z_k, \bar{\xi}_k, u_k) + f(\bar{C}, \bar{H}^* - \bar{H}, \sigma, \zeta, z_k, \bar{\xi}_k, u_k) \\
 &\quad + (\check{h}_{c1}^* - \check{h}_{c1})^T \frac{\partial f}{\partial \check{h}_{c1}^*} \Big|_{\theta} + (\sigma^* - \sigma)^T \frac{\partial f}{\partial \sigma^*} \Big|_{\theta} \\
 &\quad + (\zeta^* - \zeta)^T \frac{\partial f}{\partial \zeta^*} \Big|_{\theta} + R(\theta, \theta^*).
 \end{aligned}
 \tag{16}$$

Finally

$$\frac{\partial f}{\partial \check{h}_{c1}^*} \Big|_{\theta} = \frac{\partial \phi_h}{\partial \check{h}_{c1}^*} \varphi^T u_k, \quad \frac{\partial f_1}{\partial \zeta^*} \Big|_{\zeta} = \frac{\partial \varphi^T}{\partial \zeta} C z(k),
 \tag{17}$$

$$\begin{aligned}
 \frac{\partial \varphi^T}{\partial \zeta} = & -2 \begin{bmatrix} \frac{\partial \phi_{\varsigma_1}}{\partial \zeta_1} \frac{\sigma_1^2(y_k - \zeta_1)}{\sum_j \mu_j} \mu_1 \\ \vdots \\ \frac{\partial \phi_{\varsigma_n}}{\partial \zeta_n} \frac{\sigma_n^2(y_k - \zeta_n)}{\sum_j \mu_j} \mu_n \end{bmatrix} \otimes [\mu_1 \cdots \mu_n] \\
 & + 2diag \left(\left[\frac{\partial \phi_{\varsigma_1}}{\partial \zeta_1} \frac{\sigma_1^2(y_k - \zeta_1)}{\sum_l \mu_l} \mu_1 \cdots \frac{\partial \phi_{\varsigma_n}}{\partial \zeta_n} \frac{\sigma_n^2(y_k - \zeta_n)}{\sum_l \mu_l} \mu_n \right] \right).
 \end{aligned}
 \tag{18}$$

Which is similar to (13).

3.4 Parameter Initialization Algorithm

The parameter initialization algorithm is proposed in order to begin the training of the network with some θ such that the loss function were near some local minimum. So $(\theta^* - \theta)$ and R were also small. In [2] and [3] an algorithm is proposed. Here we propose the next algorithm:

1. Excite the system small amplitude signals and different offset values, such as the system outputs were inside some small bounded operation region. There are as many signals as fuzzy rules.

2. Using the least squares method find lineal models corresponding to each rule, which identify the system inside the small bounded operation region and whose structure is that of the linear models of the consistent part of the fuzzy rules of the network (see equation (4)).
3. Initialize C and H with the parameters c_i and h_i of the previous step.
4. Initialize ς_i with values inside the operation regions.
5. Initialize σ_i with values so that the operation regions are cover with the fuzzy sets A_i .

3.5 Training Using Kalman Filter

Once the network has been initialized near a local minimum the difference $(\theta^* - \theta)$ and R can be assumed to be also small. With this in mind, the optimized linearized model (14) can be described by

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \omega_k, \quad \hat{y}_{k+1} = G_k \hat{\theta}_k - \bar{\epsilon}_k, \tag{19}$$

As there is no direct knowledge about $\hat{\theta}$, and there exist some perturbations such as ω_k and $\bar{\epsilon}_k$ whose effect must be taken into account. A Kalman filter-based algorithm is used to train the network and then find the estimates $\hat{\theta}$ as in [9]. The filter is defined by the following equations

$$\hat{\theta}_k = \hat{\theta}_{k-1} + K_k (y^k - G_k \hat{\theta}_{k-1}), \tag{20}$$

$$K_k = \frac{P_{k-1} G_k^T}{R_2 + G_k P_{k-1} G_k^T}, \tag{21}$$

$$P_k = P_{k-1} - \frac{P_{k-1} G_k^T G_k P_{k-1}}{R_2 + G_k P_{k-1} G_k^T} + R_1. \tag{22}$$

where $E[\omega\omega^T] = R_1$, $E[\bar{\epsilon}\bar{\epsilon}^T] = R_2$ and $E[x_0 x_0^T] = P_0$

3.6 Training Using Steepest Descent Algorithm

In order to estimate the parameter on-line the steepest descent algorithm is used, that is a gradient-based algorithm. We use an algorithm to estimate the parameter on-line because we want the system to adapt to the changes in the operation regime when applying a certain equivalence control.

If the loss function is chosen as $V_{k+1}(\theta) = \frac{1}{2}(\hat{y}_{k+1} - y_{k+1})^2$, the gradient can be obtained by

$$g = \frac{\partial V(\theta)}{\partial \theta} = (\hat{y}_{k+1} - y_{k+1}) G_k. \tag{23}$$

Then θ is adjusted as follow

$$\theta_{k+1} = \theta_k - \eta g_k, \tag{24}$$

where η is a positive constant.

3.7 Certain Equivalence Control

Once the system has been identified we apply a certain equivalence control based in the network. For this, we rewrite (6) as

$$\hat{y}_{k+1} = (\varphi_k^T \otimes z_k^T) \text{vect}(C^T) + \varphi_k^T \bar{H} \bar{\xi}_k + \varphi_k^T h_{c1} u_k, \tag{25}$$

If $\varphi_k^T h_{c1} \neq 0$, we can find the next certain equivalence control

$$u_{ck} = \frac{1}{\varphi_k^T h_{c1}} (-(\varphi_k^T \otimes z_k^T) \text{vect}(C^T) - \varphi_k^T \bar{H} \bar{\xi}_k + \nu). \tag{26}$$

If the constraints to h_{c1} were not considered the elements of h_{c1} may have values with different signs and it is not possible to guarantee that $\varphi_k^T h_{c1} \neq 0$ for all φ_k^T . ν is defined as

$$\nu = y_{(k+1)d} - k_1 e_k - k_2 e_{k-1} - \dots - k_{p+1} e_{k-p}, \tag{27}$$

where $y_{(k+1)d}$ is the desired output in the time $k + 1$ and $e_k = y_k - y_{(k)d}$ is the control error in the time k . In order to have $e_k = 0$ is necessary for the roots of $z^{p+1} + k_1 z^p + k_2 z^{p-1} + \dots + k_{p+1}$ to have all its roots inside the unitary circle.

In the ideal case where ϵ_k of (15) is zero, then we would have $\lim_{k \rightarrow \infty} e_k = 0$. If $\epsilon_k \neq 0$, but is bounded, it can be assumed that the system has the following structure $y_{k+1} = f_y + f_y u_k + \zeta_k$, and is identified by $\hat{y}_{k+1} = \hat{f}_y + \hat{f}_y u_k$.

We define $f_y = \hat{f}_y - f_y$ and $\tilde{f}_u = \hat{f}_u - f_u$, by applying the control to the original system we get

$$\begin{aligned} y_{k+1} &= f_y + f_y \left(\frac{1}{\hat{f}_u} (-\hat{f}_y + \nu) \right) + \zeta_k \\ &= f_y - \hat{f}_y + \hat{f}_y + \left(\frac{f_u - \hat{f}_u + \hat{f}_u}{\hat{f}_u} (-\hat{f}_y + \nu) \right) + \zeta_k \\ &= \hat{f}_y - \tilde{f}_y + \left(\frac{\hat{f}_y - \tilde{f}_y}{\hat{f}_u} (-\hat{f}_y + \nu) \right) + \zeta_k \\ &= \hat{f}_y + \frac{\hat{f}_y}{\hat{f}_y} (-\hat{f}_y + \nu) - \tilde{f}_y + \frac{\tilde{f}_y}{\hat{f}_y} (+\hat{f}_y - \nu) + \zeta_k \\ &= \nu - (f_y + \tilde{f}_u u_k) + \zeta_k = \nu + \tilde{y}_{k+1} + 2\zeta_k \end{aligned} \tag{28}$$

so the tracking error depends on the identification error as proved in [10].

4 Experimental Results

The MT150F is a DC motor, part of the system MS150, developed by Feedback.inc. The system MS150 has several parts, we use some of them in order to build a physical system whose input is a voltage and whose output is the voltage generated in a taco-generator proportional to the angular speed of the motor. The highest output is 13 V. Just for practicity the voltage of the taco-generator is divided by 13 V, so a normalized and adimensional output is found.

The control and the on-line estimation of parameters are implemented using a personal computer. A data acquisition board NI USB-6009 is used to communicate the physical system with de computer. The analog outputs of the NI



Fig. 1. Physical system (MS150)

USB-6009 have a resolution of 12 bits and a range from 0 to 5 V. So to generate bipolar outputs it is necessary to add an offset, in software, to the control signal. This offset is eliminated using analog adders (parts of the system MS150). The analog inputs of the NI USB-6009 have a resolution of 14 bits and a range from -20 to 20 V.

The output of the taco-generator is noisy, so a digital second order low pass filter is implemented in software. The filter has a cut-off frequency of 53 HZ, this frequency is chosen because do not eliminates the frequency at which the system operates and rejects the high-frequency noise and the 60 HZ noise produced by the electric installation. The difference equation of this filter is $y_i = 0.5236x_i + 0.5236x_{i-1} + 0.0472y_{i-1}$.

In order to get input-output data, information to train the network, the system was excited with the signal shown in Fig. 2

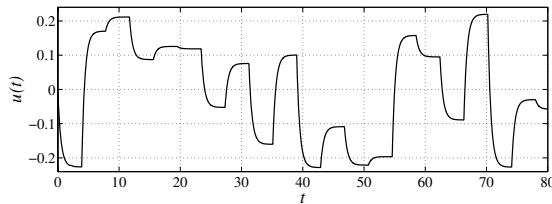


Fig. 2. Input

The number of fuzzy rules used to identify the system was $n_R = 3$. The initial value of P was $P = 108I$. R_1 and R_2 were chosen as $R1 = 10^8I$ and $R2 = 0.995$. The initialization algorithm of the Section 3.4 was used. The parameters h_{c1} and ς were forced to be near its initial values using constrains with the following form $x = r \tanh(\tilde{x}_1 + s) - t$. So, it is guaranteed that these parameters will never take undesired values.

In the Fig. 3 the solid line is the output of the physical system and the dashed line is output of the trained network. The approximation error has a RMS value of 0.0693.

The first figure of Fig. 4 shows the response of the system when certain equivalence control based on the network is applied to the system. The reference signal

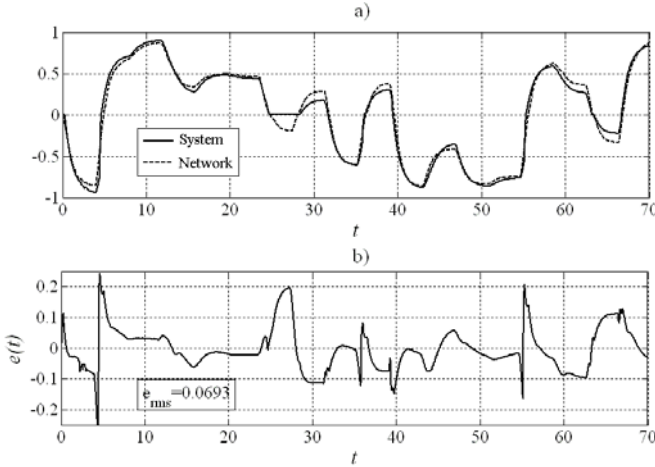


Fig. 3. a) Outputs. b) Approximation errors.

is an step with an amplitude of 0.5. ν is defined as $\nu = y_{(k+1)d} + 0.99e_k - 0.08e_{k-1}$. In order to test the control algorithm, a perturbation via a magnetic brake is applied. The system tracks the reference signal, but when the perturbation is applied the tracking error becomes greater. To overcome this problem, the proposed on-line adaptation algorithm (24) is used with $\eta = 0.000001$. In the second figure of Fig. 4, it can be seen how the system which uses the on-line adaptation algorithm tracks the reference when the brake is applied.

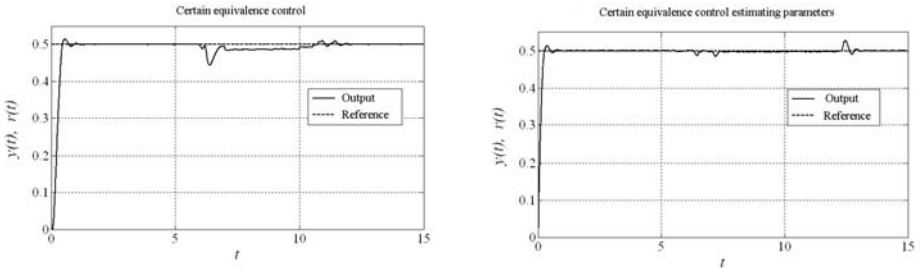


Fig. 4. Step response of the system without and with on-line estimation of parameters

5 Conclusions

The network proposed has proven to be useful identifying a physical system. The ability of the network to identify the system is good enough as to make it unnecessary to add a PID control, as proposed in [3], to compensate the error in the identification. The network can be used as an accurate model of the system for other applications. As well other control schemes based in a model can be used.

When the perturbation is applied the tracking error becomes greater, problem that was solved by using an on-line adaptation of parameters, which may also be useful when there exists some time-variant parameters.

The non-linear constrains first proposed in [3] just for the antecedent parameters has been extended to the h_{c1} parameters. This constrains are very important when a certain equivalence control is applied because they do not allow the indetermination of the control signal. In the other hand the constrains to the antecedent parameters are useful to preserve the previous knowledge about the network.

References

1. Nguyen, H.T., Prasad, N.R., Walker, C.L., Walker, E.A.: A First Course in Fuzzy and Neural Control. Chapman & Hall/CRC, New Mexico (2003)
2. Gonzalez Olvera, M., Tang, Y.: A New Recurrent Neurofuzzy Network for Identification of Dynamic Systems. *Fuzzy sets and systems* 158(10), 1023–1035 (2007)
3. Gonzalez Olvera, M., Tang, Y.: Identification and Control of Nonlinear Systems Via a Constrained Input - Output Neurofuzzy Network. In: Sun, J., Tao, G. (eds.) *Advances in Control Systems Theory and Applications*. University of Science and Technology of China (2009) ISBN 978-7-312-02238-8
4. Nelles, O.: *Nonlinear System Identification*. Springer, Berlin (2001)
5. Werbos, P.J.: *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Harvard University (1974)
6. Zadeh, L.A.: *Fuzzy sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers*. World Scientific Pub. Co. Inc., Singapore (1996)
7. Passino, K.M.: *Fuzzy control*. Addison Wesley Longman, Menlo (1998)
8. Wang, L.-X.: *Adaptative Fuzzy Systemes and Control*. Prentice Hall, Englewood Cliffs (1994)
9. Johanssons, R.: *System Modeling and Identification*. Prentice Hall, Englewood Cliffs (1993)
10. Khalil, H.: *Nonlinear Systems*. Prentice Hall, Englewood Cliffs (2002)
11. Cybenko, G.: Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals, and Systems (MCSS)*, vol. 2(4), pp. 303–314. Springer, Heidelberg (1989)
12. Narendra, K.S., Parthasarathy, K.: Identification and CControl of Dynamical Systems Using Neural Networks. *IEEE Trans. Neural Networks* 2(4), 303–314 (1990)

Hybrid Intelligent Control Scheme for an Anaerobic Wastewater Treatment Process

Rubén Belmonte-Izquierdo¹, Salvador Carlos-Hernández²,
and Edgar Nelson Sanchez¹

¹ Department of Automatic Control, CINVESTAV del IPN, Unidad Gdl.,
Av. Científica 1145, Col. El Bajío, CP. 45015,
Zapopan, Jalisco, Mexico

{rbelmont, sánchez}@gdl.cinvestav.mx

² Department of Energy and Natural Resources, CINVESTAV del IPN, Unidad Saltillo,
Carr. Saltillo-Mty Km. 13, CP. 25900,
Ramos Arizpe, Coahuila, Mexico
salvador.carlos@cinvestav.edu.mx

Abstract. A control strategy, composed by a neural observer and a fuzzy supervisor, for an anaerobic process is proposed in this paper. A recurrent high order neural observer (RHONO) is developed to estimate variables difficult to measure (biomass and substrate) in a completely stirred tank reactor (CSTR). The recurrent high order neural network (RHONN) structure is trained by an extended Kalman filter. The fuzzy supervisor uses estimations of biomass and methane production to detect biological activity inside the reactor and to apply an L/A (logarithm/anti-logarithm) control action if required in order to avoid washout. The applicability of the proposed scheme is illustrated via simulation.

Keywords: Recurrent high order neural observer, fuzzy control, anaerobic digestion.

1 Introduction

Anaerobic digestion is an encouraging method for wastewater treatment. The products (biogas and treated water) have value and can be used to offset treatment costs. However, this bioprocess is sensitive to variations on the operating conditions, such as pH, temperature, overloads, etc. In addition, some variables and parameters are hard to measure due to economical or technical constraints. Then, estimation and control strategies are required in order to guarantee adequate performance.

In biological processes there exist variables hard to measure, which can be necessary for process control [1]. The last two decades, the interest to improve the operation of bioprocesses, by applying advanced control schemes [2], [3], [4], has increased significantly. Control objectives focus on output pollution regulation, biogas production and process stability. On the other hand, observer synthesis is a priority to estimate variables, which are key for control strategies but difficult to be measured. Different observers have been proposed [5], [6]; recently, fuzzy and neural

networks algorithms have been used to design observers and controllers for bioprocesses [1], [7], [8], [9], [10], [11]. Complete knowledge of the system model is usually assumed in order to design nonlinear estimators; nonetheless this is not always possible. In some cases especial nonlinear transformations are proposed, which are not often robust in presence of uncertainties. An interesting approach for avoiding the associated problem of model-based observers is neural networks. Neural observers require feasible outputs and inputs measures and a training algorithm in order to learn the process dynamics, and the model knowledge is not strictly necessary [12], [13], [14].

In this paper, a hybrid intelligent control scheme is proposed for an anaerobic wastewater treatment process, which takes place in a CSTR. First, a RHONO is designed to estimate biomass and substrate. The observer structure uses the hyperbolic tangent as activation function and is trained using an extended Kalman filter (EKF). The main advantage of this observer is high performance and reduced complexity. After that, a fuzzy supervisor is implemented in order to apply a control action (dilution rate) as a function of the operating conditions: if a small input disturbance arrives the supervisor allows the process to operate in open loop, if a large disturbance arrives the supervisor applies the control action avoiding washout. The supervisor is based on the Takagi-Sugeno algorithm and requires only one variable: organic daily load per biomass unit (ODL/ X_2).

2 Anaerobic Digestion Process

2.1 Process Description

Anaerobic digestion (AD) is a biological process in which organic matter (substrate) is degraded by micro-organisms (biomass), in absence of oxygen. Such degradation produces biogas, consisting primarily of methane (CH_4) and carbon dioxide (CO_2). AD is developed in four successive stages: hydrolysis, acidogenesis, acetogenesis and methanogenesis. The last stage is the slowest one; it imposes the global dynamics and is considered as the limiting step. Then, special attention is focused on this stage.

A functional diagram proposed in [15] is shown in Fig. 1. Biomass is classified as: X_1 , corresponding to hydrolytic, acidogenic and acetogenic bacteria and X_2 , corresponding to methanogenic bacteria. The organic load is classified in S_1 , the components *equivalent glucose*, which model complex molecules and S_2 , the components *equivalent acetic acid*, which represent the molecules directly transformed in acetic acid.

Thus, a mathematical model is deduced as follows. On one side, the physical-chemical phenomena are modeled by a set of five algebraic equations (1), which represent the chemical acid-base equilibrium and the conservation of materials. On the other side, the biological phenomena are modeled by a set of six ordinary differential equations (2), which represent the dynamical part of the process. Finally the gaseous phase (CH_4 and CO_2) is considered as the process outputs (3). For more details, the reader is referred to [16].

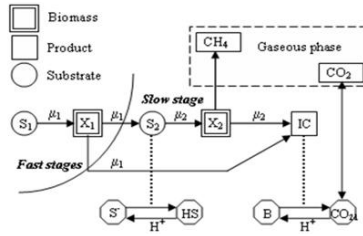


Fig. 1. Functional diagram of the anaerobic digestion

$$0 = g(x_a, x_d) \quad (1)$$

$$\dot{x}_d = f(x_a, x_d, u) \quad (2)$$

$$y = h(x_a, x_d) \quad (3)$$

with:

$$\begin{aligned} x_a &= [HS \quad H^+ \quad S^- \quad CO_{2,d} \quad B]^T, \\ x_d &= [X_1 \quad S_1 \quad X_2 \quad S_2 \quad IC \quad Z]^T, \\ u &= [S_{1in} \quad S_{2in} \quad IC_{in} \quad Z_{in} \quad D_{in}]^T, \end{aligned} \quad (4)$$

where HS is non ionized acetic acid (mol/l), H^+ ionized hydrogen (mol/l), S^- ionized acetic acid (mol/l), $CO_{2,d}$ dissolved carbon dioxide (mol/l), B measured bicarbonate (mol/l), IC inorganic carbon (mol/l), Z the total of cations (mol/l), S_{1in} the fast degradable substrate input (mol/l), S_{2in} the slow degradable substrate input (mol/l), IC_{in} the inorganic carbon input (mol/l), Z_{in} the input cations (mol/l) and D_{in} the dilution rate (h^{-1}).

2.2 Problem Statement

Methanogenesis is very sensitive to variations on substrate concentration and biomass increase can be stopped by an excessive substrate production in the previous stages [1]. Depending on the amplitude and duration of these variations, the environment can be acidified avoiding biomass growth and even producing bacteria death; besides, the hydraulic overloads can lead the process to washout (absence of active bacteria inside the reactor). From these situations, the substrate degradation and the methane production can be blocked and the process is stopped. Then, monitoring the process behavior and implementation of control strategies are important tasks in view to guarantee an adequate operation. The main idea is to develop efficient control actions, easy to implement.

Methane production, biomass growth and substrate degradation are good indicators of biological activity inside the reactor. These variables can be used for monitoring the process and to define control strategies. Nowadays, there exist commercial biogas sensors, which allow methane and carbon dioxide to be measured on-line [17].

However, substrate and biomass measures are more restrictive. The existing biomass sensors are quite expensive, are designed from biological viewpoint and then, they are not reliable for control purposes. Furthermore, substrate measure is done off-line by chemical analysis, which requires at least two hours. Then, state observers are an interesting alternative in order to deal with this situation.

3 Neural Networks Observer for Biomass and Substrate Estimation

An artificial neural network (NN) is a massively parallel distributed processor and consists of a finite number of neurons, which are interconnected to each other. One very useful NN structure is the recurrent NN. This recurrent structure has a profound impact on the learning capability of the NN and on its performance [18]. This structure also offers a better suited tool to model and control nonlinear systems [12]. Using neural networks, control algorithms can be developed to be robust to uncertainties and modeling errors. Recurrent high order neural networks (RHONN) [19] are a generalization of the first-order Hopfield networks.

3.1 Observer Development

The considered RHONO structure is displayed on Fig. 2 and is based on that proposed in [20].

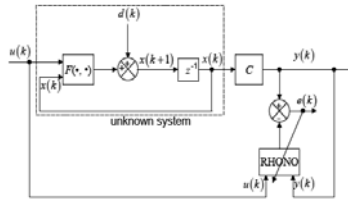


Fig. 2. Observer scheme

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ the input vector, $y \in \mathbb{R}^p$ the output vector, $h(x(k))$ a nonlinear function of the system states, $d(k) \in \mathbb{R}^n$ a disturbance vector, $e(k)$ the output error and $F(\bullet, \bullet)$ a smooth vector field.

In this paper, model (1)-(3) is transformed to discrete time. After that, a neural observer is proposed for X_2 and S_2 as follows:

$$\begin{aligned}
 \hat{X}_2(k+1) &= w_{11}S(\hat{X}_2(k)) + w_{12}S^2(\hat{X}_2(k)) + w_{13}S(\hat{C}(k)) \\
 &\quad + w_{14}S^2(\hat{X}_2(k))D_{in}(k) + g_1e(k), \\
 \hat{S}_2(k+1) &= w_{21}S(\hat{S}_2(k)) + w_{22}S^2(\hat{S}_2(k)) + w_{23}S(\hat{C}(k)) \\
 &\quad + w_{24}S^2(\hat{S}_2(k))D_{in}(k) + w_{25}S^2(\hat{S}_2(k))S_{2in}(k) + g_2e(k),
 \end{aligned}
 \tag{5}$$

with the output given as:

$$\hat{Y}_{CH_4} = R_1 R_2 \hat{\mu}_2 \hat{X}_2 \quad (6)$$

$$\hat{Y}_{CO_2} = \hat{\lambda} R_2 R_3 \hat{\mu}_2 \hat{X}_2 \quad (7)$$

The neural network is trained with an EKF [21]. For EKF-based neural network training, the network weights become the states to be estimated. The error between the neural network output and the measured plant output can be considered as additive white noise. The training goal is to find the optimal weight values, which minimize the predictions error. Besides, it is assumed that pH, Y_{CH_4}, Y_{CO_2} measures are available, as well as the system inputs. The proposed observer has a parallel configuration and the EKF training is performed on-line. The hyperbolic tangent is used as the activation function:

$$S(x) = \alpha \tanh(\beta x) \quad (8)$$

with $\alpha = \beta = 1$. Equation (8) is used because the antisymmetric functions allow the NN to learn the respective dynamics in a faster way in comparison with other functions, as explained in [22]. In addition, the hyperbolic tangent derivative is easily obtained.

The proposed observer is tuned as follows. The covariance matrices are initialized as diagonals and verifying the next inequality:

$$P_i(0) > R_i(0) > Q_i(0) \quad (9)$$

This condition implies that *a priori* knowledge is not required to initialize the vector weights [23]. In fact, higher entries in $P_i(0)$ correspond to a higher uncertainty in the *a priori* knowledge. It is advisable to set $P_i(0)$ in a number of the order of 100-1000 and so on for the other covariance matrices (9). In this way, the covariance matrices for the Kalman filter are initialized as diagonals, with nonzero elements:

$$\begin{aligned} P_1(0) = P_2(0) = P_3(0) &= 1000, \\ R_1(0) = R_2(0) &= 10, R_3(0) = 1, \\ Q_1(0) = Q_2(0) &= 1, Q_3(0) = 0.1. \end{aligned} \quad (10)$$

An arbitrary scaling can be applied to $P_i(0)$, $R_i(0)$ and $Q_i(0)$ without altering the evolution of the weight vector. Since the NN outputs do not depend directly on the weight vector, the matrix H is initialized as $H_i(0) = 0$.

It is assumed that weights values are initialized to small random values with zero mean value and normal distribution. The learning rate (η) determines the magnitude of the correction term applied to each neuron weight; it usually requires small values to achieve good training performance. To this end, it is bounded as $0 < \eta < 1$. More details concerning the parameters settings are discussed in [23].

The Luenberger-like observer gain (g) is set by trial and error; unfortunately there is a shortage of clear scientific rationale to define it. However, it is bounded to $0 < g < 0.1$ for a good performance on the basis of training experience.

3.2 Validation

The process model and the observer are implemented using Matlab/Simulink™. The observer is initialized at random values to verify the estimation convergence. In order to test the observer sensitivity to change on inputs, a disturbance (100% S_{2in} increase) on the input substrate is incepted at $t = 200$ hours and eliminated at $t = 500$ hours. Also, a parameters variation on the biomasses growth rates is considered: a 30% positive variation in μ_{2max} and 30% negative variation in μ_{1max} .

The performance of the proposed RHONO is illustrated in Fig. 3. Even though the observer is initialized at a random value, its convergence is ensured. It is clear that the biomass and substrate are well estimated since beginning. Thus, the proposed neural observer is a good alternative to estimate those important states of the considered anaerobic process.

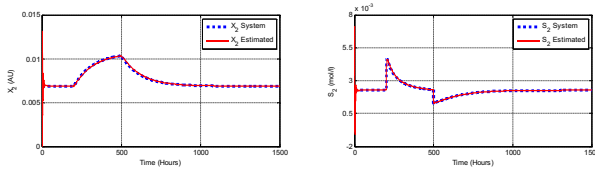


Fig. 3. Neural observer performance with variation in μ_{1max} , μ_{2max} and a disturbance in S_{2in}

4 Hybrid Intelligent Control Scheme

Anaerobic digestion is able to work adequately even in presence of small disturbances. However, in presence of large disturbances, a control law is required in order to maintain the process stability. Then, supervision of key variables is a very important task.

4.1 Design of a Control Strategy

The ODL/ X_2 is the quantity of organic load that a unit of biomass can treat in a working day and is important regarding process stability. It is defined as:

$$ODL_{\hat{X}_2} = \frac{D_k A_2 S_{2-0}}{\hat{X}_2} \tag{11}$$

where D_k is the dilution rate (h^{-1}), A_2 a disturbance amplitude on the substrate input S_{2in} (mol/l), S_{2-0} the initial value of the substrate S_2 (mol/l) and \hat{X}_2 is the estimated biomass X_2 (AU).

In presence of a disturbance on S_{2in} , ODL/X_2 can abruptly increase up to a value, which exceeds the conditions of stability limits (critical value); therefore the process tends to washout or system instability. If ODL/X_2 is above its critical value then a control law must be applied in order to allow biomass growth, and hence, diminishing ODL/X_2 and stabilizing the system. In contrast, if the ODL/X_2 is under its critical value then the system can work in open loop. Depending on the ODL/X_2 value, commutation between operating modes (open loop, closed loop) is done by a Takagi-Sugeno fuzzy supervisor. This commutation takes place progressively in order to avoid abrupt switching. The fuzzy sets are defined as in Fig. 4.

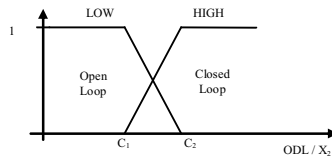


Fig. 4. Fuzzy sets for the supervisor

For closed loop, the control action is generated by an L/A controller, which is the best suited approach for anaerobic processes as concluded in [24]. It considers dilution rate changes as:

$$D_k = D_{k-1} \left(\frac{B_{k-1}}{B_k} \right)^{K_p} \left(\frac{B^*}{B_k} \right)^{K_i} \tag{12}$$

with:

$$K_p = \alpha K_i, \tag{13}$$

$$K_i = \frac{1}{1 + \alpha} \frac{\ln \left(\frac{D_{max}}{D^*} \right)}{\ln \left(\frac{B^*}{B^* + \Delta} \right)},$$

where α , Δ and D_{max} are regulation parameters, B^* is the bicarbonate setpoint (mol/l) and D^* is the dilution rate at equilibrium (h^{-1}).

The control law (12) offers advantages such as: a) It takes into account physical process constraints (positivity), and actuator saturation, and b) it does not require knowing the mathematical model of the process.

4.2 Validation

The proposed intelligent strategy is implemented using Matlab/Simulink™. A typical disturbance is considered: a S_{2in} step with amplitude $A_2 = 0.5$ (50% S_{2in} increase)

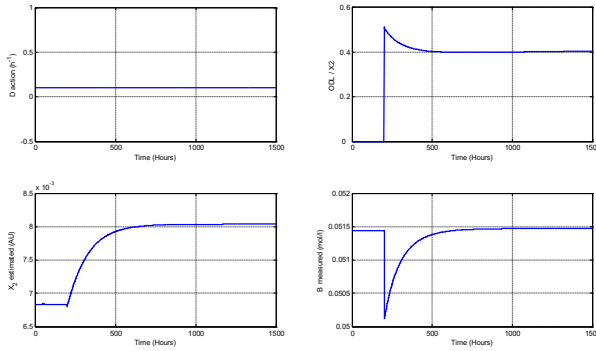


Fig. 5. Hybrid intelligent control scheme performance considering a small disturbance on S_{2in}

is incepted at $t = 200$ hours. The proposed scheme performance is illustrated in Fig. 5.

As can be seen in Fig. 5, the proposed scheme detects that A_2 is a small disturbance; hence, a control action is not needed. Due to system capability to reject this small perturbation, then it is operated in open loop.

On the other side, a S_{2in} step, with amplitude $A_2 = 2.2$ (220% S_{2in} increase) is incepted at $t = 200$. The proposed scheme performance is displayed on Fig. 6. In this case, the proposed scheme detects that A_2 is a large disturbance; hence, a control action is applied. System operating is ensured due to the control strategy applied, even though a large perturbation is incepted.

From a qualitative comparison, it can be concluded that the proposed strategy is easier to design with respect to other approaches, such as those described in [10], [11]; neither a system identification nor adaptive control schemes are required. Moreover, high performances are obtained.

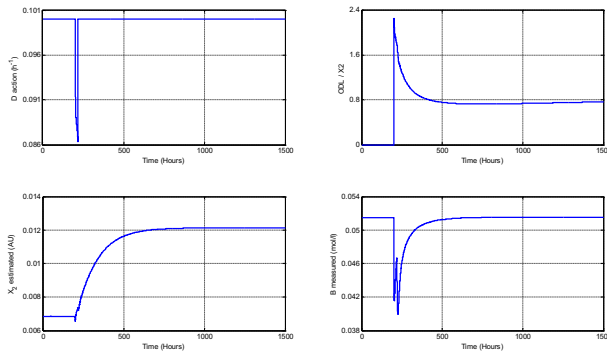


Fig. 6. Hybrid intelligent control scheme performance considering a large disturbance on S_{2in}

5 Conclusions

A RHONO, which is trained with an EFK, is proposed in this paper. The goal is to estimate the biomass concentration and substrate degradation in an anaerobic digestion process considering a CSTR with biomass filter, which is operated in continuous mode. The training of the RHONO is performed on-line. The variables are estimated from methane and carbon dioxide flow rates which are commonly measured in this process. Also, pH and system inputs measurements are assumed. Simulation results illustrate the effectiveness and robustness of the proposed RHONO.

Furthermore, a hybrid intelligent control scheme for an anaerobic wastewater treatment process is proposed in order to avoid washout. The fuzzy supervisor detects biological activity inside the tank reactor, on the basis of estimated biomass, and applies a control action. Simulations results are very encouraging.

Since one of the limiting factors for the implementation of the control strategies is the lack of on-line sensors, these neural observer outcomes are an interesting alternative to be applied. Thus research efforts are proceeding in order to implement the neural observer and the hybrid intelligent control scheme in real-time.

Acknowledgments. This work was supported by CONACyT project 57801Y.

References

1. Carlos-Hernández, S., et al.: Fuzzy Observer for the anaerobic digestion process. In: Proceedings of IFAC Symposium on Structures Systems and Control, Oaxaca, Mexico (2004)
2. Yamuna Rani, K., Ramachandra Rao, V.S.: Control of fermenters. *Bioprocess Engineering*, vol. 21, pp. 77–78. Springer, Heidelberg (1999)
3. Van Lier, J.B., et al.: New perspectives in anaerobic digestion. *Water Science and Technology* 43(1), 1–18 (2001)
4. Pind, P.F., Angelidaki, I., Ahring, B.K., Stamatelatou, K., Lyberatos, G.: Monitoring and Control of Anaerobic Reactors. In: *Advances in Biochemical Engineering/Biotechnology*, vol. 82, pp. 135–182. Springer, Heidelberg (2003)
5. Alcaraz-González, V., González-Álvarez, V.: Robust Nonlinear Observers for Bioprocesses: Application to Wastewater Treatment. Book Chapter in *Selected Topics in Dynamics and Control of Chemical and Biological Processes*, pp. 119–164. Springer Berlin, Germany (2007)
6. Dochain, D.: State and parameter estimation in chemical and biochemical processes: a tutorial. *Journal of Process Control* 13, 801–818 (2003)
7. Ascencio, P., Sbarbaro, D., Feyo de Azevedo, S.: An adaptive fuzzy hybrid state observer for bioprocesses. *IEEE Transactions on Fuzzy Systems* 12(5), 641–651 (2004)
8. Polit, M., Genovesi, A., Claudet, B.: Fuzzy logic observers for a biological wastewater treatment process. *Applied Numerical Mathematics* 39(2), 173–180 (2001)
9. Bernard, O., et al.: Advanced monitoring and control of anaerobic wastewater treatment plants: software sensors and controllers for an anaerobic digester. *Water Science and Technology* 43(7), 175–182 (2001)
10. Baruch, I.S., Galvan-Guerra, R., Nenkova, B.: Centralized Indirect Control of an Anaerobic Digestion Bioprocess Using Recurrent Neural Identifier. LNCS, pp. 297–310. Springer, Berlin (2008)

11. Holubar, P., Zani, L., Hager, M., Froschl, W., Radak, Z., Braun, R.: Advanced controlling of anaerobic digestion by means of hierarchical neural networks. *Water Research* 36, 2582–2588 (2002)
12. Poznyak, A.S., et al.: *Differential Neural Networks for Robust Nonlinear Control*. World Scientific Publishing Co., Singapore (2001)
13. Ricalde, L.J., Sanchez, E.N.: Inverse optimal nonlinear high order recurrent neural observer. In: *International Joint Conference on Neural Networks*, Montreal, Canada (2005)
14. Urrego-Patarroyo, D., et al.: Recurrent neural networks observer for anaerobic processes. In: *XIII Latin-American Congress of Automatic Control*, Mérida, Venezuela (2008)
15. Rozzi, A.: Modelling and control of anaerobic digestion process. *Transactions Instrumentation, Modelling and Control* 6(3), 153–159 (1984)
16. Carlos-Hernández, S.: *Integrated intelligent control strategy for wastewater treatment plants by anaerobic digestion*. Ph. D. Thesis, INPG, France (2005) (in French)
17. <http://www.bluesens.de> (visited on April 13, 2009)
18. Haykin, S.: *Neural Networks: A Comprehensive Foundation*, 2nd edn. Prentice Hall, New Jersey (1999)
19. Rovithakis, G.A., Chistodoulou, M.A.: *Adaptive Control with Recurrent High-Order Neural Networks*. Springer, Berlin (2000)
20. Sanchez, E.N., et al.: *Discrete Time High Order Neural Control Trained with Kalman Filtering*. Springer, Germany (2008)
21. Song, Y., Grizzle, J.W.: The extended Kalman filter as a local asymptotic observer for discrete-time nonlinear systems. *Journal of Mathematical Systems, Estimation and Control* 5(1), 59–78 (1995)
22. Sanchez, E.N., Alanís, A.Y.: *Neural Networks: Fundamental Concepts and Applications to Automatic Control*. Pearson Education, Madrid (2006) (in Spanish)
23. Haykin, S.: *Kalman Filtering and Neural Networks*. John Wiley & Sons, New York (2001)
24. Lakrori, M.: *Commande L/A applications*. Ph. D. Thesis, INPG, France (1989) (in French)

Workability of a Genetic Algorithm Driven Sequential Search for Eigenvalues and Eigenvectors of a Hamiltonian with or without Basis Optimization

Subhajit Nandy^{1,2}, Pinaki Chaudhury³, and Shankar Prasad Bhattacharyya^{1,*}

¹ Department of Physical Chemistry, Indian Association for the Cultivation of Science, Jadavpur, Kolkata 700 032, India
Fax: +(91) (33) 2473 2805

pcspb@iacs.res.in

² Permanent Address: Andrew's High (H.S.) School, Kolkata-700031
subho29@hotmail.com

³ Department of Chemistry, University of Calcutta, Kolkata - 700009, India
pinakicu@gmail.com

Abstract. A sequential strategy for searching out the eigenvalues and eigenvectors of a real symmetric Hamiltonian matrix by Genetic Algorithm (GA) is proposed and its workability demonstrated. The fitness landscape is generated from the gradient of the Rayleigh quotient (RQ) for the state concerned in such a way that the achievement of fitness maximum condition isolates the corresponding eigenvalue and eigenvector. An avenue for computing the excited eigenvalues is suggested and an inverse iteration scheme is invoked to accelerate the convergence. The non-linear problem of diagonalization of the Hamiltonian with simultaneous optimization of the basis set is explored.

Keywords: Stochastic diagonalization, diagonalization by GA, diagonalization with basis optimization, sequential search for eigenvalues, Rayleigh quotient minimization, etc.

1 Introduction

Eigenvalue problems constitute an important branch of linear algebra and often arise naturally while studying linearized motion in a physical system [1,2,3,4,5,6,7,8,9,10]. A full solution of the eigenvalue problem yields important information concerning the level-structure and stability of the system, wave propagation, decay, etc. In quantum mechanics, observables of interest are represented by hermitian operators. The Hamiltonian operator (\hat{H}) is one such operator representing energy.

In calculating the spectrum of H one generally proceeds by introducing a fixed orthonormal basis set (ϕ_i) for expanding ψ ($\psi = \sum_{i=1}^N c_i \phi_i$) and performs a linear variational calculation leading to a N - dimensional matrix eigen value problem

* Corresponding author.

$HC = EC$. The diagonalization of H yields the eigenvalues and eigenvectors. This general problem continues to attract attention. Ahlrichs and Tsereteli [11] have recently proposed an excellent technique for extracting all the eigenvalues of H . The problems relating to extraction of few eigenvalues have been recently reviewed by Killingbeck and Jolicard [12].

One way to handle the matrix eigenvalue problem could be to reduce it to an optimization problem and then invoke an efficient search strategy to find a scalar a_i and vector X_i such that the equation $HX_i = a_iX_i$ is satisfied. In this context, methods based on extremization of Rayleigh quotient could be explored [3]. It is indeed a paradigm shift when we talk about a stochastic route to isolating a few eigenvalues of a hermitian hamiltonian. Although there have been some attempts in this direction [13, 14, 15], a lot remains to be done to make the stochastic methods competitive with the currently available deterministic methods. We demonstrate in this communication that the Genetic Algorithm (GA) can be invoked to handle the problem concerned by taking the norm of the gradient of the Rayleigh quotient ($\nabla\rho$) as the objective function and an appropriate function of $\nabla\rho$ as the generator of a fitness landscape [10]. The resulting algorithm may be used for extracting the eigenvalues sequentially, starting from the lowest one. More importantly, one can easily enlarge the scope of the algorithm by using a variable basis and use GA for the simultaneous optimization of the linear and non-linear parameters in the basis functions while diagonalizing H . We may mention here that like the modified GA-Jacobi method developed earlier [14], the GA driven Rayleigh quotient minimization method too, is inherently parallelizable, a feature which has, however, not been exploited in the present dispensation.

2 The Method

2.1 Diagonalization in a Fixed Basis : Lowest Eigenvalue and Vector

GAs [13, 14] simulate evolution of a population of probable solutions on a fitness landscape. The evolution takes place under the action of genetic operators (GO) like selection, crossover and mutation so as to maximize the average fitness of the population. The Genetic operators act on the individual members of the evolving population, each member being represented in the form of either a binary string, or an integer string or a string of floating point numbers. In our formulation, we will be working with strings of floating point numbers, each of which represents some variable to be optimized. In the present context, they stand for the amplitudes of the evolving wavefunction projected on a given basis set. Thus, taking ψ_i to be the i th plausible wavefunction string (solution) present in the evolving population we write [Φ_p is an orthonormal basis containing, let us say, ' m ' adjustable non-linear parameters $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m)$]

$$\psi_i = \sum_{p=1}^n c_{pi} \phi_p^\alpha \quad i = 1, 2, \dots, n_p \quad (1)$$

the corresponding string s_i would be given by

$$s_i \equiv (c_{1,i}, \dots, c_{l,i}, \dots, c_{n,i}, \alpha_{1i}, \alpha_{2i}, \dots, \alpha_{mi}) \tag{2}$$

while another string s_k would be similarly represented by

$$s_k \equiv (c_{1,k}, \dots, c_{l,k}, \dots, c_{n,k}, \alpha_{1k}, \alpha_{2k}, \dots, \alpha_{mk}) \tag{3}$$

Each of the strings (s_i) would have a unique value of fitness (f_i) associated with it, f_i being defined as follows. Assuming the basis $\{\Phi_p^\alpha\}$ to be orthonormal, we may transform the Hamiltonian operator (\hat{H}) into an appropriate Hamiltonian matrix $H(\alpha)$, and set out for defining the Rayleigh Quotient ρ_i for the i th wavefunction string $\psi_i \equiv s_i$, for the ground state of the system defined by the Hamiltonian matrix $H(\alpha)$ for the particular set of non-linear basis parameters α as follows:

$$\rho_i = \frac{C_i^\dagger H(\alpha_i) C_i}{C_i^\dagger C_i} \tag{4}$$

If C_i happens to be the ground eigenvector of H (i.e. ψ_i is the ground state eigenfunction of $\hat{H}(\alpha_i)$)

$$\nabla \rho_i = \frac{2[H - \rho_i \cdot \mathbb{1}]C_i}{C_i^\dagger C_i} = 0, \tag{5}$$

a condition that will not be generally satisfied by any one of the wavefunction strings in the starting population, except by an accident. Therefore, we can make use of $\nabla \rho_i$ as the generator of fitness landscape by taking

$$f_i = e^{-\lambda(\nabla \rho_i)^\dagger \nabla \rho_i} \tag{6}$$

where λ is a real number chosen to take care of numerical overflow or underflow.

Clearly, $f_i \rightarrow 1$, as $\nabla \rho_i \rightarrow 0$, a situation that signals that we have hit the true ground state eigenvector in C_i . For selection, roulette wheel with slot widths proportional to fitness values is spun n_p times ($n_p =$ population size) and the position of the pointer is noted. It is expected that the scheme will allow us to copy a larger number of strings with higher fitness onto the mating pool since higher fitness means wider slots on the wheel. Crossover operation is now carried out on a pair of strings allowing partial information exchange between them with a probability p_c . Suppose, the basis set is a fixed one and the pair of strings (s_k, s_l) have been randomly selected for crossover at the p th site chosen randomly with probability p_c . The crossover operator now creates a pair of new strings (s'_k, s'_l) where

$$s'_k = (c_{1k}, c_{2k}, \dots, c'_{pk}, c_{p+1,k} \dots c_{nk}) \tag{7}$$

$$s'_l = (c_{1l}, c_{2l}, \dots, c'_{pl}, c_{p+1,l} \dots c_{nl}) \tag{8}$$

and

$$c'_{pk} = f c_{pk} + (1 - f) c_{pl} \tag{9}$$

$$c'_{pl} = (1 - f) c_{pk} + f c_{pl} \tag{10}$$

with $f[0, 1]$ being drawn from a randomly distributed set of values and the non-linear parameter set does not change. f plays the role of a mixing parameter responsible for creation of new strings [9-11]. Usually, only a certain percentage of strings (75%) in the population are made to undergo crossover. Once the operation crossover is complete, all the strings in the post selection and post crossover phases are made to suffer a low intensity and low probability event called mutation with probability p_m . In the case of a binary coded string mutation randomly flips one bit with probability p_m ($0 \rightarrow 1, 1 \rightarrow 0$). In a floating point string, however, mutation is a user-defined operation and in our case, it has been operationally defined as follows:

Let q be the site chosen for mutation on the i th string with a pre-set probability p_m . Then, the amplitude c'_{qi} is going to suffer mutation [9] as follows

$$c''_{qi} = c'_{qi} + (-1)^L r.s \tag{11}$$

Here L is a random integer, r a random number (0,1) and s is the intensity of mutation specified by the user. s can be static (held fixed in every generation) or a dynamic one (value changes as a function of the number of generations elapsed). For screening, we have adopted the strictly elitist strategy.

2.2 Finding Higher Eigenvalues and Eigenvectors in a Fixed Basis

For the eigenvector (eigenvalue) of the next higher state a slightly different strategy has to be adopted. This is based on the principle that the Rayleigh quotient (ρ) has a minimum at the vector corresponding to the lowest eigenvalue and a saddle point for every other eigenvector of H . However, locating a saddle point is a much more difficult job than finding a minimum. We would therefore recast the problem of locating an excited eigenvalue in the form of a minimization problem. Since we have already found the ground (lowest eigenvalue) state ($|\psi_0\rangle$) we may project out the ground state and construct a new projected Hamiltonian [9,12-13]

$$H_p = (1 - P_0)^\dagger H (1 - P_0) \quad , \quad P_0 = |\psi_0\rangle\langle\psi_0|$$

and apply the strategy already described to extract the lowest eigenvalue of H_p with or without level shifts.

2.3 Diagonalization with Basis Optimization : Lowest Eigenvalue

An advantage of a GA-driven stochastic diagonalization method lies in its ability to handle the eigenvalue problem most generally. That means, it can treat the linear and non-linear parameters in the wavefunction on equal footing and achieve optimization of both sets of parameters simultaneously without any change in the structure of the algorithm. The only change that needs to be introduced is in the structure of the strings. The modified strings S_k^m will now have the amplitudes c_i s as well as the non-linear parameters in the basis function ($\alpha_{1k}, \alpha_{2k}, \dots, \alpha_{mk}$): Thus $S_k^m = (c_{1k}, \dots, c_{ik}, \dots, c_{nk}, \alpha_{1k}, \alpha_{2k}, \dots, \alpha_{nk})$.

The crossover and mutation operators remain the same. The basis does not remain orthonormal when non-linear parameters are allowed to vary during the evolution. We have therefore a generalised eigenvalue problem to deal with

$$H\bar{C} = S\bar{C}E \quad (12)$$

Here S is the metric of the basis.

The GA searches through the (C, α) space democratically. However, it is a better strategy to deal with variations in linear and the non-linear parameters slightly differently. Let us assume that the fixed basis $(\Phi(\alpha))$ is not orthonormal. Here the H matrix is constructed in the $\Phi(\alpha)$ basis. It is transformed into \bar{H} matrix by symmetrical orthogonalization, where

$$\bar{H} = S^{-\frac{1}{2}}HS^{-\frac{1}{2}} \quad (13)$$

The eigenvalue equation now becomes

$$\bar{H}C = CE \quad (14)$$

The Rayleigh Quotient is then defined using \bar{H} and GA is used to find the eigenvectors of \bar{H} as already outlined. GA, of course, changes the nonlinear parameters (α) embedded in Φ . Such changes are automatically included in \bar{H} through symmetric orthogonalization indirectly.

3 Results and Discussion

We have applied the GA-RQ method to several problems to demonstrate its workability, its strong features, drawbacks and point out possible directions for improvements. In what follows, we present the results pertaining to applications of the method to three different problems.

3.1 Ground and Excited Eigenvalues in a Fixed Basis

GA for the Ground Eigenvalue. The test calculations are performed with Morse Hamiltonian matrices of size 101X101, 250X250, 500X500, respectively. We have made use of the Fourier Grid Hamiltonian (FGH) method [17] for constructing the Hamiltonian matrix. In the first example 101 grid points uniformly distributed over a grid of length 21 a.u. have been used. A population size $n_p = 20$ has been used with the crossover probability fixed at $p_c = 0.75$ and the mutation probability p_m fixed at 0.5, for all generations. The initial population was generated randomly from a set of nodeless amplitude distributions (the highest fitness in the starting population ~ 0.1). Fig. 1(a) shows the profile of evolution of fitness of the best string in the population through the generations. One can see the steep rise in fitness value to ~ 0.9 in about 5000 generations. Fig. 1(b) shows the profile of energy of the best evolving wavefunction string in the course of evolution. The sharp fall of energy to $E_0 = 0.0045$ a.u. in the first 5000 generations mirrors the rapid rise of fitness value of the same string

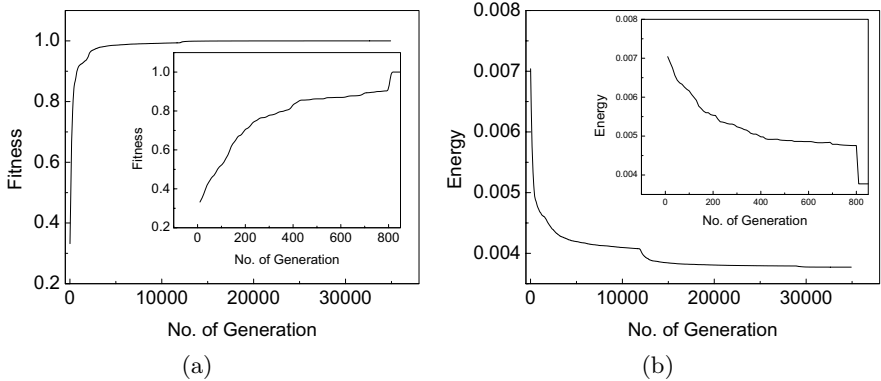


Fig. 1. (a): Fitness evolution of the best string during the calculation of the lowest eigenvalue of a Morse Oscillator(101 grid point FGH calculation). Inset: The same when inverse iteration technique is applied after 800 generations. **(b):** Evolution of the lowest eigenvalue of the same Morse Oscillator. Inset: Profile with inverse iteration.

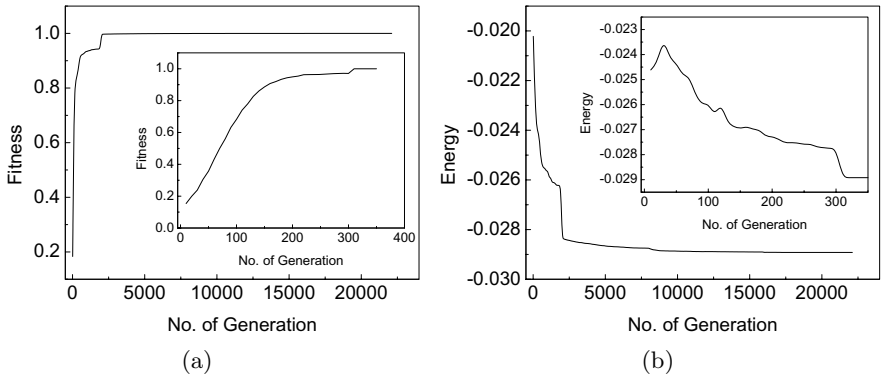


Fig. 2. (a): Fitness evolution of the best string while calculating the first excited state of a Morse Oscillator. Inset: Same fitness profile when inverse iteration is done after 300 generations. **(b):** Energy evolution of the first excited state of a Morse Oscillator during a GA- RQ run. Inset: Profile with application of inverse iteration technique.

during the same period. The initial evolution is controlled mainly by crossover which introduces gross changes in the evolving wavefunction strings. The flat portion of the fitness profile ranging from $n_{gen} = 5000 - 30000$ is dominated by tiny corrections in the amplitude distribution mainly brought in by mutation. It is possible to invoke ‘inverse iteration’ using the best string in the population at this stage as the initial guess to force convergence to the eigenvector and eigenvalue desired. If the strategy works it can cut down the computational cost substantially. Inset in figures 1(a) and 1(b) are the fitness and energy

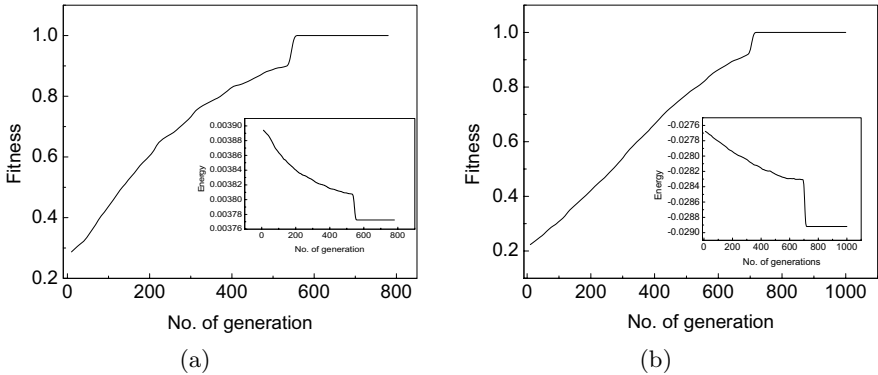


Fig. 3. (a): Fitness profile of the best string of a Morse Oscillator (250 grid point FGH calculation) when inverse iteration technique is applied after 550 generations. Inset: The rapid convergence of the lowest eigenvalue for the same. **(b):** Fitness evolution of the best string while calculating the eigenvalue of the first excited state of a Morse Oscillator (250 grid point FGH calculation) by applying inverse iteration technique after 700 generations. Inset: Energy evolution of the first excited state of the same.

profiles with inverse iterations switched on as fitness reaches values in the range 0.8-0.85 during the search for ground state eigenfunction. The convergence to the ground eigenvalue occurs in only one additional iteration, suggesting that it would be computationally profitable to resort to inverse iteration technique at the appropriate stage, rather than persisting with the GA based search right to the end. We have presented one specific realization of the algorithm which represents almost the worst case scenario.

Sequential Search for Higher Eigenvalues. The evolution of the fitness profile for the first excited eigenvalue of the Morse Oscillator has been displayed in Fig. 2(a) while the corresponding profile for energy evolution has been depicted in Fig. 2(b). Here also, the appearance of the plateau in the fitness profile signals that the inverse iteration can be invoked at this point reducing for computational labour (see the inset figures in 2(a) and (b)). Figures 3(a) and 3(b) exhibit the profiles for fitness and energy evolution for the ground and excited states of the oscillator. In each case , inverse iteration was invoked at the appropriate stage for accelerating convergence. Figure 4(a) shows the corresponding profiles for the ground state of Morse Oscillator when 500 grid points are used (the dimension of the H matrix is 500X500). It turns out that there is no deterioration in the performance index of the method proposed as the basis space increases from 100 to 500 , and that the inverse iteration scheme may be reliably invoked with great facility once the fitness has increased to 0.8 - 0.9 in the GA based search for eigenfunctions.

3.2 Diagonalization of Hamiltonian with Simultaneous Optimization of Basis Parameters

The workability of the GA-based strategy of computing eigenvalues along with basis optimization is demonstrated in the context of finding the ground state energy of Helium atom in a restricted (s-type) basis.

The trial wavefunction strings $\hat{\psi}_k(r_1, r_2)$ for the ground state of He has been taken in the form

$$\hat{\psi}_k(r_1, r_2) = \sum_{i,j} C_{ij}^{(k)} \phi_{ij}^{(k)}(r_1, r_2; n_i, n_j, l_i = l_j; \alpha_i^{(k)}; \alpha_j^{(k)}) \Phi_{spin}(s_1, s_2) \quad (15)$$

where

$$\phi_{ij}^{(k)} = N \left[\chi(r_1, \alpha_{n_i}^{(k)}) \chi(r_2, \alpha_{n_j}^{(k)}) + \chi(r_1, \alpha_{n_j}^{(k)}) \chi(r_2, \alpha_{n_i}^{(k)}) \right] \quad (16)$$

and $\phi_{spin}(s_1, s_2)$ is the singlet eigenfunction of S^2 operator. In equation (19), we have used $n_i = n_j = 0, 2, 4, 6, 8, 10$. The string s_k would therefore comprise in the N amplitudes $\{C_{ij}^k\}$ and n number of non-linear parameters $\{\alpha_{n_i}\}$, all of which are allowed to evolve simultaneously to maximize the fitness function defined already. Although only the linear variational parameters $C_{ij}^{(n)}$ appear explicitly in the fitness function, there is implicit dependence of fitness f_k on $\alpha_{n_i}^{(k)}$ through the dependence of matrix elements of H and $C_{ij}^{(k)}$'s on $\alpha_{n_i}^{(k)}$'s. χ 's represent eigenfunctions of a 3-d spherical oscillator. We have used functions with $l=0$ only as we have targeted the S-limit energy of the ion. Figure 4(b) shows the growth of fitness of the best fit string in the evolving population, representing the ground state of He atom. Figure 4(b Inset), on the other hand, displays the complementary profile of energy, the converged ground state energy being

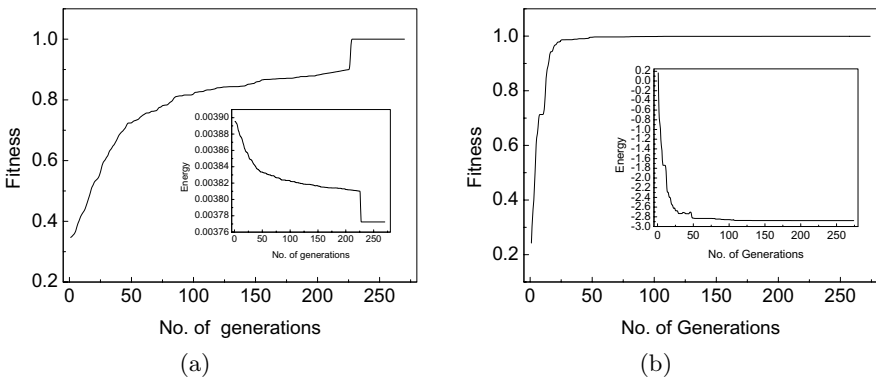


Fig. 4. (a): Fitness evolution of the best string for the evaluation of the lowest eigenvalue of a Morse Oscillator when 500 grid points are taken. Inset: Energy profile for the same Morse Oscillator. **(b):** Fitness growth of the best string while calculating the ground state eigenvalue of He atom with simultaneous optimization of basis parameters. Inset: Evolution of the eigenvalue of the same system.

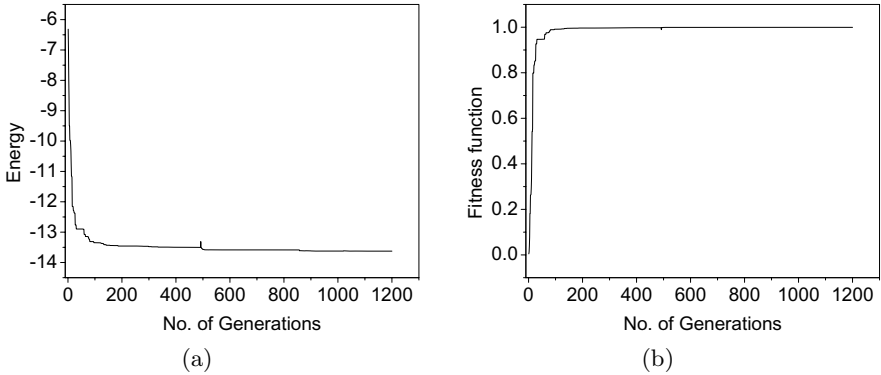


Fig. 5. (a): Evolution of energy of the ground state of Be^{+2} ion in GA-RQ method based on a simultaneous optimization of basis parameters. **(b):** The fitness profile of the same problem.

$-2.87846a.u$ (virtually the s-limit energy). The profiles are quite smooth, indicating that the simultaneous optimization of linear and non-linear parameters take place without any difficulty in our GA-RQ minimization method. As another example, we consider the the ground state of Be^{+2} ion. Here the coupling among the linear and non-linear parameters appears to be rather strong and the diagonalization with simultaneous optimization of the non-linear parameters of the basis set proceeds in a relatively checkered manner as opposed to diagonalization in a fixed basis (fig. 5a, 5b) implying that the fitness landscape is more rugged. Nevertheless the s-limit energy is reached in six hundred generations (Actual CI- -13.62685 a.u., obtained -13.62521 a.u.). As to the computational labour, diagonalization in a fixed basis and diagonalization with simultaneous basis optimization do not appear to differ much in their demands on computational time for low dimensional basis spaces. For higher dimensional basis the repeated construction of the H matrix as the basis parameters are changed by genetic operators consumes substantial cpu although optimization of non-linear parameters enables one to use basis space of lower dimensions. The GA-based method proposed here is perfectly workable although it may not competitive in its present form with deterministic routes to the eigenvalues and eigenvectors, except when used with simultaneous optimization of non-linear parameters in the basis. However, the present algorithm is parallelizable [17] and its overall efficiency can improve many fold once a parallel code is developed for implementing the GA-RQ technique [18]. We may also mention that it is possible to redesign the GA-based strategy to extract more than one eigenvalue and vector simultaneously. We hope to return to this problem in the near future.

4 Conclusion

Genetic Algorithm based on minimization of Rayleigh Quotient leads to affordable realization of sequential stochastic diagonalization of a Hamiltonian matrix.

The scheme is quite different from the modified GA-Jacobi method proposed earlier [9] and can accommodate simultaneous optimization of the basis set as well as search for excited state eigenvalues. It would be interesting to parallelize the algorithm and extend it to finding eigenvalues of much larger matrices, specially in the context of designing optimal basis sets, targeting more than one eigenvalue at a time.

Acknowledgments. SPB thanks the DST, Government of India, New Delhi, for a generous research grant. PC wishes to thank the centre for research on Nano Science and Nano Technology, University of Calcutta for a research grant, Conv/002/Nano RAC (2008).

References

1. Parlett, B.N.: *The symmetric Eigenvalue Problem*. Prentice Hall, Englewood Cliffs (1981)
2. Wilkinson, J.H.: *The Algebraic Eigenvalue problem*. Clarendon, Oxford (1965)
3. Kerner, W.J.: *J. Comput. Phys.* 85, 1 (1989)
4. Cullum, J.K., Willoughby, R.A.: *Lanczos Algorithm for Large Symmetric Eigenvalue Computations*. Birk häuss, Boston (1985)
5. Davidon, E.R.: *J. Comput. Phys.* 17, 87 (1975)
6. Shavitt, I., Bender, C.F., Pipand, A., Hosteny, R.P.: *J. Comput. Phys.* 11, 90 (1973)
7. Coope, J.A.R., Sabo, D.W.: *J. Comput. Phys.* 44, 20 (1981)
8. Raedt, H.D., Frick, M.: *Phys. Reports* 231, 107 (1993)
9. Nandy, S., Chaudhury, P., Bhattacharyya, S.P.: *Intern. J. Quantum Chem.* 90, 188 (2002)
10. Nandy, S., Chaudhury, P., Bhattacharyya, S.P.: *J. Chem. Sci.* 116, 285 (2004)
11. Ahlrichs, R., Tsereteli, K.: *J. Comput. Chem.* 23, 306 (2002)
12. Killingbeck, J.P., Jolicard, G.: *J. Phys. A* 36, R105–R180 (2003)
13. Holland, J.H.: *Adaptation in Natural and Artificial System*. University of Michigan Press, Ann Arbor (1975)
14. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine learning*. Addison Wesley, Reading (1989)
15. Lowdin, P.O.: *Intern. J. Quantum Chem.* 2, 167 (1968)
16. Lowdin, P.O.: *Phys. Rev. A* 139, 357 (1965)
17. Marston, C.C., Balint-Kurti, G.G.: *J. Chem. Phys.* 91, 357 (1989)
18. Saha, R., Bhattacharyya, S.P., Taylor, C.D., Zhao, Y., Cundari, T.R.: *Intern. J. Quantum Chem.* 94, 243 (2003)

An Improved Quantum Evolutionary Algorithm Based on Artificial Bee Colony Optimization

Haibin Duan, Zihui Xing, and Chunfang Xu

School of Automation Science and Electrical Engineering, Beihang University, Beijing,
100191, P.R. China
hbduan@buaa.edu.cn

Abstract. Quantum evolutionary algorithm (QEA) is proposed on the basis of the concept and principles of quantum computing, which is a classical meta-heuristic algorithm for the approximate solution of combinatorial optimization problems that has been inspired by the principles of evolution of living organisms in nature. QEA has strong robustness and is easy to combine with other methods in optimization, but it has the shortcomings of stagnation that limits the wide application to the various areas. In this paper, a hybrid QEA with Artificial Bee Colony (ABC) optimization was proposed to overcome the above-mentioned limitations. ABC is adopted to increase the local search capacity as well as the randomness of the populations, which can help QEA jump out of the premature convergence and find the optimal value. The proposed algorithm is tested with the Benchmark optimization problem, and the experimental results demonstrate that the proposed QEA is feasible and effective in solving complex optimization problems.

Keywords: Quantum evolutionary algorithm (QEA), Artificial Bee Colony (ABC), Q-bit chromosome, Crossover, Premature.

1 Introduction

Since the concept of quantum was put forward, there was a revolution coming in the field of computing, and it was coming from quantum—the smallest of all places: the subatomic particles that form the basis of all matters.

Quantum computing has promised prodigious powers in the past years [1]. Its basic currency, the Q-bit, exists in an ON or OFF verge, which you will never know until it's read out. Therefore, if you could operate on K Q-bits, a potentially vast space of 2^K values opens up for computation which means that we can solve many computing problems at the same time, which saves you a lot of time. The fundamental operation on Q-bits is a rotation. We have logic gates to combine the rotations. The algorithm is based on these logic gates. In principle, these algorithms can perform calculations far beyond classical computation's conceivable reach.

Genetic algorithm (GA) was firstly put forward by J. Holland in 1970s to study the self adaptation behavior of natural system [2]. It's a classical meta-heuristic algorithm

for the approximate solution of combinatorial optimization problems that has been inspired by the principles of evolution of living organisms in nature. The application of GA needs no initiating knowledge of the system, and it isn't limited by the form and property of the problem. Guided by fitness function and principle of probability, it can search in global according to self adaptation by using selection, crossover and mutation. Therefore, it's a comprehensive optimization method with extensive application in terms of processing complex non-linear problems.

GA has strong robustness and is easy to combine with other methods in optimization, but it has limited population size and the problems of premature convergence and stagnation that limit the wide application to the various areas often exist. Q-bit chromosomes enjoy a rapidly growing population and strong randomization.

To overcome the above-mentioned shortcomings of GA, quantum evolutionary algorithm (QEA) is proposed on the basis of the concept and principles of quantum computing. In QEA, Q-bit chromosomes, which can represent a linear superposition of solutions, are adopted to maintain solution diversity and overcome premature convergence. At the same time, quantum rotation gate, which make full use of the information of the current best individual, is used to update individual and avoid stagnation [3].

The common QEA uses Q-bit gate rotation in mutation and whole interference in crossover [4]. By using rotation operation, we can make full use of the information of the currently best individual to perform the next searching process, and the whole interference can avoid prematurity. In this way, the global search capacity can be greatly improved, while the convergence speed is slowed down.

Artificial Bee Colony Algorithm (ABC) is an optimization algorithm based on the intelligent foraging behavior of honey bee swarm, proposed by Karaboga in 2005[5]. It is as simple as Particle Swarm Optimization(PSO) and Differential Evolution(DE) algorithms, and uses only common control parameters such as colony size and maximum cycle number. ABC as an optimization tool, provides a population-based search procedure in which individuals called foods positions are modified by the artificial bees with time and the bee's aim is to discover the places of food sources with high nectar amount and finally the one with the highest nectar. The most prominent feature of ABC algorithm is that this system combines local search methods, carried out by employed and onlooker bees, with global search methods, managed by onlookers and scouts, attempting to balance exploration and exploitation process. Thus, ABC algorithm can have a very good search capacity.

In order to further improve the whole performance of QEA, we proposed a new hybrid strategy combined with the ABC algorithm in this paper. And the experiment results prove the feasibility and effectiveness of the algorithm in solving complex optimization problem.

The remainder of this paper is organized as follows. The next section introduces the main process of common QEA. Section 3 proposes a hybrid QEA model based on ABC. Then, in Section 4, the experimental results are given to verify the feasibility of the proposed approach. Our concluding remarks are contained in the final section.

2 Main Process of Basic QEA

2.1 Qubit Chromosome

In QEA, a qubit chromosome as a string of n qubits can be defined as follows [6]:

$$q = \begin{bmatrix} \alpha_1 & | & \alpha_2 & | & \dots & | & \alpha_m \\ \beta_1 & & \beta_2 & & \dots & & \beta_m \end{bmatrix} \tag{1}$$

Where $|\alpha_i|^2 + |\beta_i|^2 = 1, i=1, \dots, m, m$ is the number of qubits as well as the string length of the qubit individual. $|\alpha_i|^2$ provides the probability that the qubit will be found in the state of ‘0’, while $|\beta_i|^2$ gives the probability that the qubit will be found in the ‘1’ state. A qubit chromosome is able to represent a linear superposition of all possible solutions. It has a better characteristic of diversity than classical chromosome [7]. The process to get classical chromosome is: bring a random number between 0 and 1, if it’s bigger than $|\alpha_i|^2$, this bit in classical chromosome is ‘1’, else ‘0’ is chosen.

2.2 Quantum Mutation

The standard mutation operation is totally random without any directions, so the speed of convergence is slowed down. But in QEA, the qubit representation can be used as a mutation operator. Directed by the current best individual, quantum mutation is completed through the quantum rotation gate $U(\theta)$, then the $[\alpha_i \ \beta_i]^T$ is updated as:

$$\begin{bmatrix} \alpha'_i \\ \beta'_i \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} \tag{2}$$

Look up the table (1) below to find out the right θ_i . it’s determined by both quantum and classical chromosome:

Table 1. Rotation angle

x_i	$best_i$	$f(x) > f(\text{best})$	θ_i			
			$\alpha_i \beta_i > 0$	$\alpha_i \beta_i < 0$	$\alpha_i = 0$	$\beta_i = 0$
0	0	False	0	0	0	0
0	0	True	0	0	0	0
0	1	False	0	0	0	0
0	1	True	-0.05π	0.05π	$\pm 0.05\pi$	0
1	0	False	-0.05π	0.05π	$\pm 0.05\pi$	0
1	0	True	0.05π	-0.05π	0	$\pm 0.05\pi$
1	1	False	0.05π	-0.05π	0	$\pm 0.05\pi$
1	1	True	0.05π	-0.05π	0	$\pm 0.05\pi$

x_i is the i -th bit of the current classical chromosome, $best_i$ is the i th bit of the current best classical chromosome, $f(x)$ is the adaptation function [8].

The Figure 1 below describes the polar plot of the rotation operation on qubit. It tells us the reason why the rotation gate can increase the speed of convergence obviously [9].

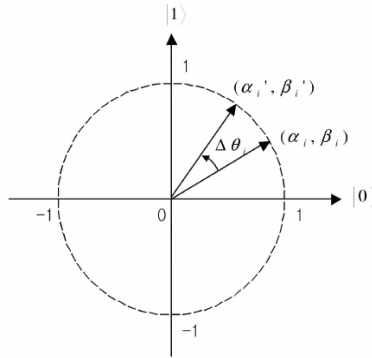


Fig. 1. Polar plot of the rotation gate for qubit chromosome

2.3 Quantum Whole Interference Crossover

This kind of crossover operation is constructed by the interference characteristic of qubit. All the quantum chromosomes are involved in. For example, when the population number is 5 and the length of chromosome is 6, the table 2 below introduces a kind of operation:

Table 2. The whole interference crossover operation

1	A(1)	E(2)	D(3)	C(4)	B(5)	A(6)	E(7)
2	B(1)	A(2)	E(3)	D(4)	C(5)	B(6)	A(7)
3	C(1)	B(2)	A(3)	E(4)	D(5)	C(6)	B(7)
4	D(1)	C(2)	B(3)	A(4)	E(5)	D(6)	C(7)
5	E(1)	D(2)	C(3)	B(4)	A(5)	E(6)	D(7)

The whole interference crossover operation can make full use of the information in the chromosome, improve the unilateralism of classical crossover and avoid premature convergence and stagnation problems.

3 The Proposed Hybrid QEA Based on ABC

3.1 Artificial Bee Colony(ABC) Optimization

Artificial Bee Colony (ABC) is one of the most recently defined algorithms by Dervis Karaboga in 2005, motivated by the intelligent behavior of honey bees [5]. In ABC system, artificial bees fly around in the search space, and some (employed and onlooker bees) choose food sources depending on the experience of themselves and their nest mates, and adjust their positions. Some (scouts) fly and choose the food sources randomly without using experience. If the nectar amount of a new source is

higher than that of the previous one in their memory, they memorize the new position and forget the previous one. Thus, ABC system combines local search methods, carried out by employed and onlooker bees, with global search methods, managed by onlookers and scouts, attempting to balance exploration and exploitation process.

In order to introduce the model of forage selection that leads to the emergence of collective intelligence of honey bee swarms, first, we need to define three essential components: food sources, unemployed foragers and employed foragers.

- **Food Sources (A and B in Fig.3):** For the sake of simplicity, the “profitability” of a food source can be represented with a single quantity. In our function optimization problem, the position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution.
- **Unemployed foragers:** If it is assumed that a bee have no knowledge about the food sources in the search field, bee initializes its search as an unemployed forager. Unemployed foragers are continually at look out for a food source to exploit. There are two types of unemployed foragers: scouts and onlookers.
 - **Scouts (S in Fig.3):** If the bee starts searching spontaneously for new food sources without any knowledge, it will be a scout bee. The percentage of scout bees varies from 5% to 30% according to the information into the nest.
 - **Onlookers(R in Fig.3):** The onlookers wait in the nest and search the food source through sharing information of the employed foragers, and there is a greater probability of onlookers choosing more profitable sources.
- **Employed foragers:** They are associated with a particular food source which they are currently exploiting. They carry with them information about this particular source, the profitability of the source and share this information with a certain probability. After the employed foraging bee loads a portion of nectar from the food source, it returns to the hive and unloads the nectar to the food area in the hive. There are three possible options related to residual amount of nectar for the foraging bee.
 - If the nectar amount decreased to a low level or exhausted, foraging bee abandons the food source and become an unemployed bee.(UF in Fig. 3)
 - If there are still sufficient amount of nectar in the food source, it can continue to forage without sharing the food source information with the nest mates.(EF2 in Fig. 3)
 - Or it can go to the dance area to perform waggle dance for informing the nest mates about the food source.(EF1 in Fig. 3), as is shown in Fig.2

In this way, the bees finally can construct a relative good solution of the multimodal optimization problems.

3.2 The Proposed Hybrid QEA Based on ABC

The QEA provide new ideas to improve the traditional GA. Firstly, the information in a quantum chromosome is more than that in a classical chromosome, the number of population is decreased and the diversity is improved. Secondly, the mutation

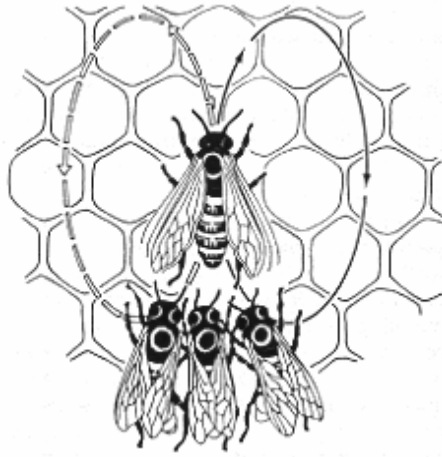


Fig. 2. Waggle dance of honey bees

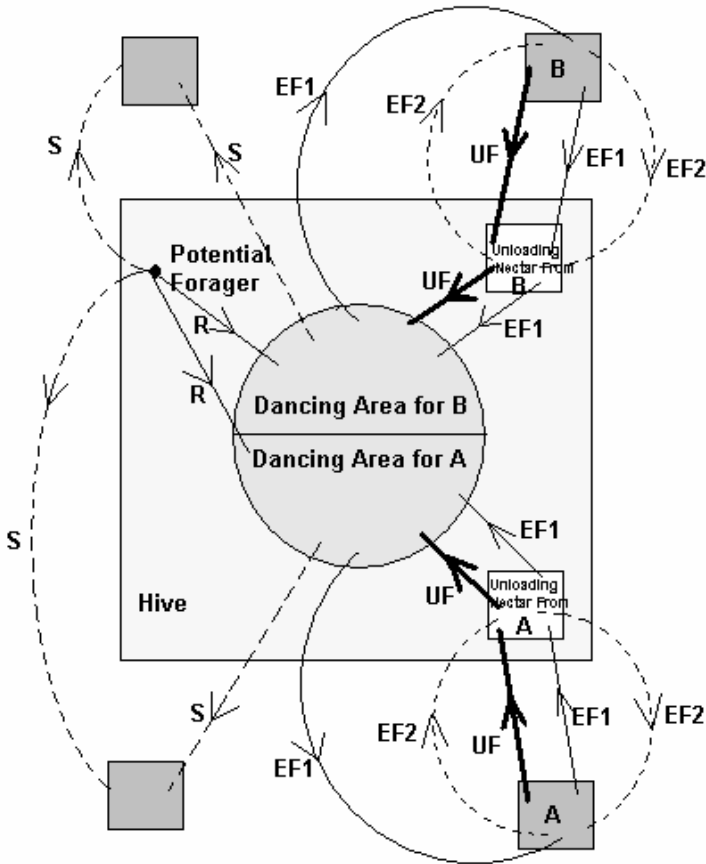


Fig. 3. The behavior of honey bee foraging for nectar

operation is no longer totally random but directed by some rules to make the next generation better and increase the speed of convergence. Thirdly, whole interference crossover operation can avoid premature convergence and stagnation problems.

Although the QEA has many advantages, it still has space to be improved. In our test, we can easily find out that the QEA could not always reach the best solution of the problem, which means that the algorithm still has a considerable probability of premature convergence. To solve this problem, we proposed a hybrid QEA based on ABC algorithm. The result is extraordinary both in theory and experiments.

In our proposed hybrid QEA, we introduced the ABC idea into the traditional QEA to enhance the ability of global search. The parameters above are just the same with basic QEA.

At the first step, a randomly distributed initial population (food source positions) is generated.

After initialization, the population is subjected to repeat the cycles of the search processes of the employed, onlooker, and scout bees, respectively.

An employed bee produces a modification on the source position in her memory and discovers a new food source position. Provided that the nectar amount of the new one is higher than that of the previous source, the bee memorizes the new source position and forgets the old one. Otherwise she keeps the position of the one in her memory.

In this process, the modification strategy is achieved by the classical single point crossover as well as the quantum rotation. In the single point crossover process, Roulette selection operation is used to choose two quantum chromosomes from the parent generations, and then the child generation is produced by crossover. After this, two better individuals can be chosen into the next generation by evaluating their fitness. This operation is mainly to improve the convergence speed and preserve the instructive information. Usually, we choose 0.6 to 0.9 as the crossover probabilities. After the crossover process, we choose the best one as the mutation director, and implement the quantum mutation operation using the rules shown in Table 1. This operation is also to improve the convergence speed as well as to increase the diversity of the population. We choose 0.01 to 0.2 as the mutation probabilities. After all the operations, finally the employed bee selects the better population as the new source position to remember.

After all employed bees complete the search process, they share the position information of the sources with the onlookers on the dance area. Each onlooker evaluates the nectar information taken from all employed bees and then chooses a food source depending on the nectar amounts of sources. As in the case of the employed bee, she produces a modification on the source position in her memory and checks its nectar amount. Providing that its nectar is higher than that of the previous one, the bee memorizes the new position and forgets the old one.

The sources abandoned are determined and new sources are randomly produced to be replaced with the abandoned ones by artificial scouts.

4 Experimental Results

In order to investigate the feasibility and effectiveness of the proposed hybrid QEA algorithm, experiment is conducted on the Benchmark problem: to find the maximum value of function

$$f(x) = 10 + \frac{\sin(1/x)}{(x-0.16)^2 + 0.1} \quad x \in (0,1) \tag{3}$$

The experiment has been encoded in Matlab language and implemented on PC-compatible with 512 Mb of RAM under the Windows XP. The parameters were set to the following values: n=20, N_employed=10, N_unemployed=10, L=22, Pcc=0.9, Pm=0.2, germax =100, Limit=30. Figure 4 shows the final position of the chromosomes in this experiment, and the maximum and the average evolution curves are presented in Figure 5. Table 3 shows the final specific results that contains both the maximum value and the average value of the chromosomes in the experiment.

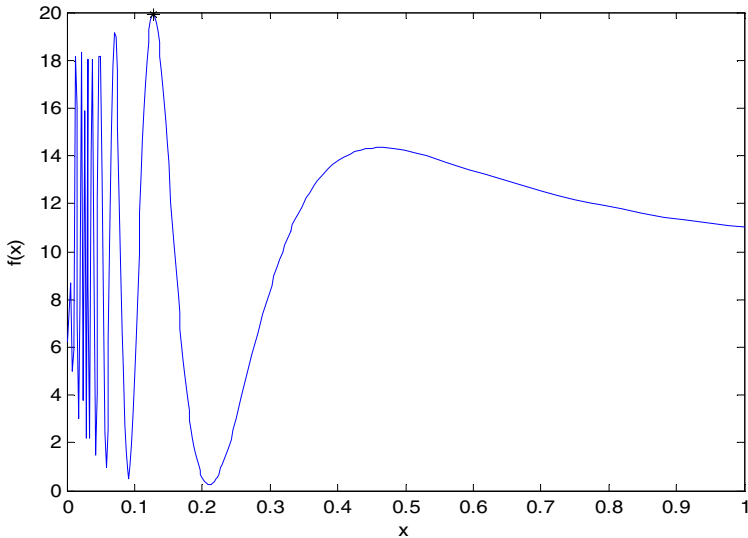


Fig. 4. Final positions of the chromosomes in experiment 3

Table 3. The maximum value and the average value of the chromosome

Times	1	2	3	4	5	6	7	8	9	10
Max	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949
Average	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949
Times	11	12	13	14	15	16	17	18	19	20
Max	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949
Average	19.8949	19.8949	19.8949	19.5014	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949

Average(max)	19.8949
Average(average)	19.8736

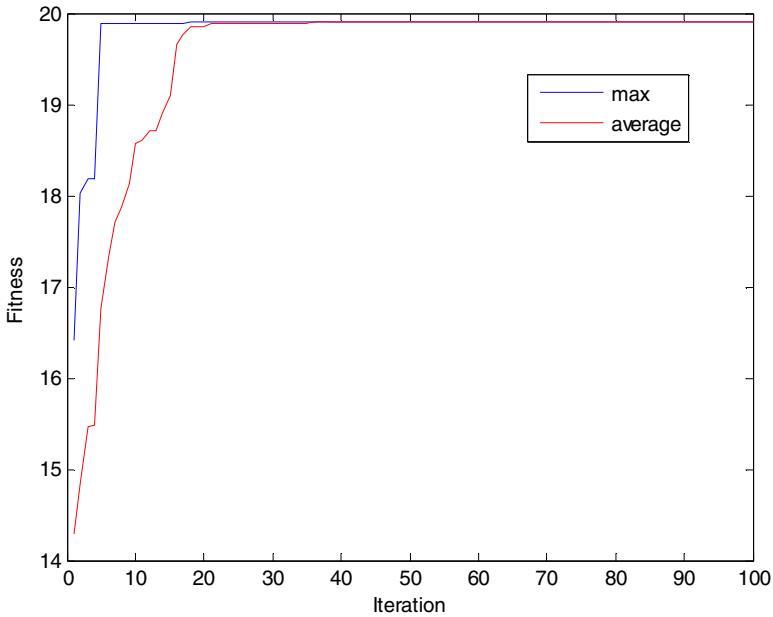


Fig. 5. The maximum and the average evolution curve

It is obvious that our proposed QEA model can find the satisfactory solutions in solving continuous optimization problems.

5 Conclusions

This paper has presented an improved QEA based on ABC algorithm for solving the continuous optimization problems. The experimental results verify that the proposed hybrid QEA and ABC model is a practical and effective algorithm in solving complex optimization problems, and also a feasible method for other complex real-world optimization problems. Our future work will focus on applying the newly proposed hybrid QEA and ABC approach in this paper to other more complicated optimization problems. Furthermore, we are also interested in the theoretical analysis on the proposed hybrid QEA and ABC model.

Acknowledgments. This work was partially supported by the Natural Science Foundation of China under grant #60604009, Aeronautical Science Foundation of China under grant #2006ZC51039, Beijing NOVA Program Foundation of China under grant #2007A0017, and New Scientific Star in Blue Sky Talent Program of Beihang University of China.

References

- [1] Al-Rabadi, A.N.: Circuits for m-valued classical, reversible and quantum optical computing with application to regular logic design. *International Journal of Intelligent Computing and Cybernetics* 2(1), 52–101 (2009)

- [2] Holland, J.: *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor (1975)
- [3] Zhang, R., Gao, H.: Improved quantum evolutionary algorithm for combinatorial optimization problem. In: 6th International Conference on Machine Learning and Cybernetics, Hong Kong, pp. 3501–3505 (2007)
- [4] Yang, S.Y., Liu, F., Jiao, L.C.: The Quantum Evolutionary Strategies. *Acta Electronica Sinica* 29(12A), 1873–1877 (2001)
- [5] Dervis, K.: An idea based on honey bee swarm for numerical optimization. Technical Report-TR06 (2005)
- [6] Zhang, W.F., Shi, Z.K., Luo, Z.Y.: Prediction of Urban Passenger Transport Based-on Wavelet SVM with Quantum-Inspired Evolutionary Algorithm. In: Proceedings of the 2008 International Joint Conference on Neural Networks, Hong Kong, pp. 1510–1514 (2008)
- [7] Tayarayi, M.H., Akbarzadeh-T, M.R.: A Cellular Structure and Diversity Preserving operator in Quantum Evolutionary Algorithms. In: Proceedings of the 2008 IEEE Congress on Evolutionary Computation, Hong Kong, pp. 2670–2675 (2007)
- [8] Xiao, J., Yan, Y.P., Lin, Y., Yuan, L., Zhang, J.: A Quantum-inspired Genetic Algorithm for Data Clustering. In: Proceedings of the IEEE Congress on Evolutionary Computation, Hong Kong, pp. 1513–1518 (2008)
- [9] Wei, M., Li, Y.X., Jiang, D.Z., He, Y.F., Huang, X.Y., Xu, X.: A New Evolutionary Algorithm based on Quantum Statistical Mechanics. In: Proceedings of the IEEE Congress on Evolutionary Computation, Hong Kong, pp. 1722–1726 (2008)

Rough Approximation Operators with Hedges

Xueyou Chen

School of Mathematics, Shandong University of Technology,
Zibo, Shandong 255049, P.R. China
chenxueyou0@yahoo.com.cn

Abstract. The theory of formal concept analysis and the theory of rough sets are related. In the paper, from the point of view of graded truth approach, two \mathbf{L} -rough operators with hedges are defined, some of their properties are investigated, which correspond to the fuzzy concept lattice with hedges induced by R. Bělohlávek, et. al.

Keywords: rough set, \mathbf{L} -set, hedge, rough approximation operator.

1 Introduction

Z. Pawlak introduced rough set theory to study incomplete and insufficient information([13]). In rough set theory, the approximation of an arbitrary subset of a universe by two definable subsets are called lower and upper approximations, which correspond to two rough operators. The two rough operators were first defined by means of a given indiscernibility relation in [13]. Usually indiscernibility relations are supposed to be equivalences. As generalizations, they were defined by means of an arbitrary binary relation ([20]) and a fuzzy relation in [4, 5, 6, 10, 11, 12, 14, 15, 16, 18].

On the other hand, in [1], from the point of view of graded truth approach, R. Bělohlávek studied \mathbf{L} -orders, \mathbf{L} -closure operators, \mathbf{L} -concept lattice. In [2, 3], they studied the \mathbf{L} -concept lattice with hedges to control the size of the concept lattice.

The theory of rough sets and the theory of formal concept analysis are related, see [4, 17, 19]. Corresponding to the fuzzy concept lattice with hedges in [2, 3], in the paper, we introduce the rough approximation operators with hedges, and investigate their algebraic properties.

Our paper is organized as follows: we begin with an overview of rough sets, \mathbf{L} -sets, \mathbf{L} -concept lattice in Section 2, which surveys Preliminaries. Then, in Section 3, we define two \mathbf{L} -rough approximation operators with hedges by means of an \mathbf{L} -binary relation I , and investigate some of their properties. In Section 4, we extend the two rough approximation operators on two universes. In Section 5, we investigate the relation between the rough approximation operators with hedges and the fuzzy concept lattice with hedges.

2 Preliminaries

We introduce some main notions for each area, i.e., rough sets ([13, 20]), **L**-sets ([1, 7]), **L**-concept lattice ([1, 2, 3]).

2.1 Rough Sets

Pawlak initiated rough set theory in [13]. Let (X, R) be an approximation space, and $R \subseteq X \times X$ be an equivalence relation, then for $A \subseteq X$, two subsets $\underline{R}(A)$ and $\overline{R}(A)$ of X are defined:

$$\underline{R}(A) = \{x \in X \mid [x]_R \subseteq A\} \quad \text{and} \quad \overline{R}(A) = \{x \in X \mid [x]_R \cap A \neq \emptyset\},$$

where $[x]_R = \{y \in X \mid xRy\}$.

If $\underline{R}(A) = \overline{R}(A)$, A is called a definable set; if $\underline{R}(A) \neq \overline{R}(A)$, A is called an undefinable set, and $(\underline{R}(A), \overline{R}(A))$ is referred to as a pair of rough set. \underline{R} and \overline{R} are called two rough operators.

The two rough operators, were defined by means of a partition, or an equivalence relation in [13]. Yao generalized rough set theory by generalizing the equivalence relation to a more general relation, see [20].

The theory of rough sets has also been constructed on fuzzy framework, see [5, 10, 11, 12, 14, 18]. A fuzzy subset in X is a mapping $A : X \rightarrow [0, 1]$, $F(X)$ is the set of all fuzzy subsets of X , R is a fuzzy relation, $\forall A \in F(X)$, $\overline{R}(A)$ and $\underline{R}(A)$ are fuzzy sets defined as following: $\forall x \in X$,

$$\overline{R}(A)(x) = \bigvee_{y \in X} R(x, y) \wedge A(y), \quad \underline{R}(A)(x) = \bigwedge_{y \in X} (1 - R(x, y)) \vee A(y),$$

In [15, 16], A. M. Radzikowska and E. E. Kerre investigated fuzzy rough sets based on a residuated lattice, see Definition 4 and [4].

2.2 L-Sets

As a generalization of Zadeh’s (classical) notion of a fuzzy set, the notion of an **L**-set was introduced in [7]. An overview of the theory of **L**-sets and **L**-relations (i.e., fuzzy sets and relations in the framework of complete residuated lattices) can be found in [1]. Let us recall some main definitions.

Definition 1. Suppose $\mathbf{L} = \langle L, \vee, \wedge, \otimes, \rightarrow, 0, 1 \rangle$ is a residuated lattice, a hedge is a function \sim on L which fulfils these properties:

$$1^\sim = 1, \quad a^\sim \leq a, \quad (a \rightarrow b)^\sim \leq a^\sim \rightarrow b^\sim, \quad a^{\sim\sim} = a^\sim.$$

Hedge \sim is a (truth function of) logic connective “very true”, see [2, 8, 9]. $a^\sim \leq a$ may be read: if a is vert true, then a is true; $(a \rightarrow b)^\sim \leq a^\sim \rightarrow b^\sim$ may be read: if $a \rightarrow b$ is very true, and a is very true, then b is also very true; etc.

The following lemma will be used in this paper, see [2].

Lemma 1. A hedge \sim satisfies $(\bigvee_{i \in I} a_i^\sim)^\sim = \bigvee_{i \in I} a_i^\sim$.

Residuated lattice \mathbf{L} is called complete if $\langle L, \vee, \wedge \rangle$ is a complete lattice. In this paper, we assume that \mathbf{L} is complete.

Let \mathbf{L} be a residuated lattice. \mathbf{L} satisfies the law of double negation if it satisfies: $a = (a \rightarrow 0) \rightarrow 0$ for every $a \in L$. \mathbf{L} satisfies idempotency if it satisfies $a \otimes a = a$ for every $a \in L$, see [1] p.32. \mathbf{L} satisfies the prelinearity axiom if for $a, b, c \in L$, $(a \wedge b) \rightarrow c = (a \rightarrow c) \vee (b \rightarrow c)$, or $a \rightarrow (b \vee c) = (a \rightarrow c) \vee (a \rightarrow c)$ holds, see [1] p.42.

For a universe set X , an \mathbf{L} -set in X is a mapping $A : X \rightarrow L$. $A(x)$ indicates the truth degree of “ x belongs to A ”. We use the symbol L^X to denote the set of all \mathbf{L} -sets in X . The negation operator is defined: for $A \in L^X$, $A^*(x) = A(x) \rightarrow 0$ for every $x \in X$. For $A \in L^X$, A^\sim is defined as: for every $x \in X$, $A^\sim(x) = (A(x))^\sim$.

$I \in L^{X \times X}$ is called an \mathbf{L} -binary relation. The truth degree to which elements x and y are related by an \mathbf{L} -relation I is denoted by $I(x, y)$ or (xIy) .

For $A, B \in L^X$, the subsethood degree $S(A, B)$ is defined as:

$$(A \preceq B) = S(A, B) = \bigwedge_{x \in X} A(x) \rightarrow B(x), \quad (A \approx B) = S(A, B) \wedge S(B, A).$$

Thus $\langle\langle L^X, \approx \rangle, \preceq \rangle$ is an \mathbf{L} -ordered set. We write $A \subseteq B$, if $S(A, B) = 1$.

Definition 2. (1) Suppose $\{A_i \mid i \in I\} \subseteq L^X$, $\bigvee_{i \in I} A_i$, $\bigwedge_{i \in I} A_i$ are two \mathbf{L} -sets defined: for every $x \in X$, $(\bigvee_{i \in I} A_i)(x) = \bigvee_{i \in I} A_i(x)$, $(\bigwedge_{i \in I} A_i)(x) = \bigwedge_{i \in I} A_i(x)$

(2) For $A, B \in L^X$, $A \otimes B$ is an \mathbf{L} -set in X defined: for every $x \in X$, $(A \otimes B)(x) = A(x) \otimes B(x)$.

Definition 3. A mapping $C : L^X \rightarrow L^X$ is called an \mathbf{L} -closure operator, if for $A, B \in L^X$, we have

(1) $A \subseteq C(A)$, (2) $S(A, B) \leq S(C(A), C(B))$, and (3) $C(C(A)) = C(A)$.

2.3 \mathbf{L} -Concept Lattice Introduced by R. Bělohlávek

In [1], suppose X and Y are two sets with \mathbf{L} -equalities \approx_X and \approx_Y , respectively; I an \mathbf{L} -relation between X and Y which is compatible with respect to \approx_X and \approx_Y . A pair $\langle \uparrow_I, \downarrow_I \rangle$ of mappings was defined as:

$$\uparrow_I : L^X \rightarrow L^Y, \text{ for } A \in L^X, \quad A^{\uparrow_I}(y) = \bigwedge_{x \in X} A(x) \rightarrow I(x, y)$$

$$\downarrow_I : L^Y \rightarrow L^X, \text{ for } B \in L^Y, \quad B^{\downarrow_I}(x) = \bigwedge_{y \in Y} B(y) \rightarrow I(x, y).$$

Then $\langle X, Y, I \rangle$ is called a formal \mathbf{L} -context.

Clearly $A^{\uparrow_I}(y)$ is the truth degree to which each object of A has the attribute y , and $B^{\downarrow_I}(x)$ is the truth degree to which each attribute of B is shared by the object x .

$\langle A, B \rangle$ is called a concept in $\langle X, Y, I \rangle$, if $A^{\uparrow_I} = B$, $B^{\downarrow_I} = A$. Then A and B are called an extent and an intent of $\langle A, B \rangle$, respectively.

$\beta(X, Y, I) = \{ \langle A, B \rangle \mid \langle A, B \rangle \text{ is a concept} \}$ is called a formal concept lattice in $\langle X, Y, I \rangle$. A is also called a concept of objects, B is called a concept of attributes. For $\langle A_1, B_1 \rangle, \langle A_2, B_2 \rangle \in \beta(X, Y, I)$, R. Bělohlávek defined:

$$S(\langle A_1, B_1 \rangle, \langle A_2, B_2 \rangle) = S(A_1, A_2) = S(B_2, B_1)$$

$$(\langle A_1, B_1 \rangle \approx \langle A_2, B_2 \rangle) = (A_1 \approx A_2).$$

Suppose $\{\langle A_i, B_i \rangle\} \subseteq \beta(X, Y, I)$, the meet and the join were defined,

$$\bigwedge_i \langle A_i, B_i \rangle = \langle \bigwedge_i A_i, (\bigwedge_i A_i)^\uparrow \rangle = \langle \bigwedge_i A_i, (\bigvee_i B_i)^{\downarrow\uparrow} \rangle,$$

$$\bigvee_i \langle A_i, B_i \rangle = \langle (\bigwedge_i B_i)^\downarrow, \bigwedge_i B_i \rangle = \langle (\bigvee_i A_i)^{\uparrow\downarrow}, \bigwedge_i B_i \rangle.$$

In [2], R. Bělohlávek, T. Funioková and V. Vychodil introduced the notion of Galois connections with hedges. In [3], R. Bělohlávek and V. Vychodil defined the fuzzy concept lattice with hedges.

Suppose X and Y are two sets with \mathbf{L} -equalities \approx_X and \approx_Y , respectively; let \sim_X, \sim_Y be hedges. I an \mathbf{L} -relation between X and Y which is compatible with respect to \approx_X and \approx_Y . A pair $\langle \uparrow_I, \downarrow_I \rangle$ of mappings was defined as:

$$\uparrow_I : L^X \rightarrow L^Y, \text{ for } A \in L^X, \quad A^{\uparrow_I}(y) = \bigwedge_{x \in X} A^{\sim_X}(x) \rightarrow I(x, y).$$

$$\downarrow_I : L^Y \rightarrow L^X, \text{ for } B \in L^Y, \quad B^{\downarrow_I}(x) = \bigwedge_{y \in Y} B^{\sim_Y}(y) \rightarrow I(x, y).$$

Then $\langle X^{\sim_X}, Y^{\sim_Y}, I \rangle$ is called a formal \mathbf{L} -context.

Clearly $A^{\uparrow_I}(y)$ is the truth degree to which each object of A^{\sim_X} has the attribute y , and $B^{\downarrow_I}(x)$ is the truth degree to which each attribute of B^{\sim_Y} is shared by the object x .

$\langle A, B \rangle$ is called a concept in $\langle X^{\sim_X}, Y^{\sim_Y}, I \rangle$, if $A^{\uparrow_I} = B, B^{\downarrow_I} = A$. Then A and B are called an extent and an intent of $\langle A, B \rangle$, respectively.

$\beta(X^{\sim_X}, Y^{\sim_Y}, I) = \{\langle A, B \rangle \mid \langle A, B \rangle \text{ is a concept}\}$ is called a formal concept lattice in $\langle X^{\sim_X}, Y^{\sim_Y}, I \rangle$. A is also called a concept of objects, B is called a concept of attributes.

In [3], R. Bělohlávek et. al. studied many choices of \sim_X, \sim_Y , and \sim_X, \sim_Y can be seen as parameters controlling the size of the resulting $\beta(X^{\sim_X}, Y^{\sim_Y}, I)$.

3 Rough Operators with Hedges

In [4], we defined the two rough approximation operators as follows:

Suppose $I \in L^{X \times X}$ is an \mathbf{L} -binary relation, for every $A \in L^X$, we defined two \mathbf{L} -sets $\mathbf{N}(A)$ and $\mathbf{H}(A)$ in X , for every $x \in X$,

$$\mathbf{N}(A)(x) = \bigwedge_{y \in X} A(y) \vee I^*(y, x), \quad \mathbf{H}(A)(x) = \bigvee_{y \in X} I(x, y) \otimes A(y)$$

$\mathbf{N}(A)$ and $\mathbf{H}(A)$ are called an \mathbf{L} -lower approximation and an \mathbf{L} -upper approximation of A , respectively. \mathbf{N}, \mathbf{H} are called an \mathbf{L} -lower approximation operator and an \mathbf{L} -upper approximation operator, respectively, or \mathbf{N} and \mathbf{H} are called \mathbf{L} -rough operators.

In the section, we are going to define two \mathbf{L} -rough operators with hedge by means of an \mathbf{L} -binary relation I , and investigate some of their properties.

Definition 4. Suppose $I \in L^{X \times X}$ is an \mathbf{L} -binary relation, $\sim (\sim_X)$ is a hedge, for every $A \in L^X$, we define two \mathbf{L} -sets $N(A)$ and $H(A)$ in X , for every $x \in X$,

$$N(A)(x) = \bigwedge_{y \in X} A^\sim(y) \vee I^*(y, x), \quad H(A)(x) = \bigvee_{y \in X} I(x, y) \otimes A^\sim(y)$$

$N(A)$ and $H(A)$ are called an **L**-lower approximation and an **L**-upper approximation of A , respectively. N, H are called an **L**-lower approximation operator with hedge and an **L**-upper approximation operator with hedge, respectively, or N and H are called **L**-rough operators with hedge.

For every $A \in L^X$, $(N(A), H(A))$ is referred to as a pair of **L**-rough set. Clearly we have $N(A) \subseteq \mathbf{N}(A), H(A) \subseteq \mathbf{H}(A)$.

When $\mathbf{L}=\mathbf{2}$, and I is an equivalence relation, the above definition coincides with Pawlak’s definition, see Section 2.1.

Let $P = \{N(A) \mid A \in L^X\}, Q = \{H(A) \mid A \in L^X\}$. We define the relation **L**-order \preceq , and **L**-equivalence relation \approx on P and Q . For example: for $N(A), N(B) \in P$,

$$(N(A) \preceq N(B))S(N(A), N(B)) = \bigwedge_{x \in X} N(A)(x) \rightarrow N(B)(x),$$

$$(N(A) \approx N(B)) = S(N(A), N(B)) \wedge S(N(B), N(A)).$$

By [1], we obtain that $\langle\langle P, \approx \rangle, \preceq\rangle$ and $\langle\langle Q, \approx \rangle, \preceq\rangle$ are two **L**-ordered sets.

We investigate some properties of the **L**-rough operators N and H .

- Proposition 1.** (1) $N(X) = X, \quad H(\emptyset) = \emptyset,$
 (2) $S(A^\sim, B^\sim) \leq S(N(A), N(B)), \quad (3) S(A^\sim, B^\sim) \leq S(H(A), H(B))$
 (4) $N(A^\sim) = N(A), \quad (5) H(\bigvee_{i \in I} A_i^\sim) = \bigvee_{i \in I} H(A_i).$

Proof. (1) Since for every $x \in X$, we have

$$N(X)(x) = \bigwedge_{y \in X} X^\sim(x) \vee I^*(y, x) = \bigwedge_{y \in X} 1^\sim \vee I^*(y, x) = \bigwedge_{y \in X} 1 \vee I^*(y, x) = 1.$$

$$H(\emptyset)(x) = \bigvee_{y \in X} I(x, y) \otimes \emptyset^\sim(x) = \bigvee_{y \in X} I(x, y) \otimes 0^\sim = \bigvee_{y \in X} I(x, y) \otimes 0 = 0.$$

Therefore the two equalities $N(X) = X, H(\emptyset) = \emptyset$ hold.

(2) For $A, B \in L^X$, by the definition of $S(N(A), N(B)) = \bigwedge_{x \in X} N(A)(x) \rightarrow N(B)(x)$, we have to prove $S(A^\sim, B^\sim) \leq S(N(A), N(B))$, which holds if and only if $S(A^\sim, B^\sim) \leq N(A)(x) \rightarrow N(B)(x)$ holds for every $x \in X$. That is to say, $S(A^\sim, B^\sim) \otimes N(A)(x) \leq N(B)(x)$ holds for every $x \in X$. It is true, since

$$\begin{aligned} S(A^\sim, B^\sim) \otimes N(A)(x) &= S(A^\sim, B^\sim) \otimes \bigwedge_{y \in X} A^\sim(y) \vee I^*(y, x) \\ &\leq \bigwedge_{y \in X} S(A^\sim, B^\sim) \otimes [A^\sim(y) \vee I^*(y, x)] \\ &= \bigwedge_{y \in X} [S(A^\sim, B^\sim) \otimes A^\sim(y)] \vee [S(A^\sim, B^\sim) \otimes I^*(y, x)] \\ &\leq \bigwedge_{y \in X} B^\sim(y) \vee I^*(y, x) = N(B)(x). \end{aligned}$$

(3) For $A, B \in L^X$, by the definition of $S(H(A), H(B)) = \bigwedge_{x \in X} H(A)(x) \rightarrow H(B)(x)$, in order to prove $S(A, B) \leq S(H(A), H(B))$, we have to show for every $x \in X, S(A^\sim, B^\sim) \leq H(A)(x) \rightarrow H(B)(x)$ holds. That is, $S(A^\sim, B^\sim) \otimes H(A)(x) \leq H(B)(x)$ holds for every $x \in X$. It is valid, since

$$\begin{aligned}
 S(A^\sim, B^\sim) \otimes H(A)(x) &= S(A^\sim, B^\sim) \otimes \bigvee_{y \in X} I(x, y) \otimes A^\sim(y) \\
 &= \bigvee_{y \in X} S(A^\sim, B^\sim) \otimes I(x, y) \otimes A^\sim(y) \leq \bigvee_{y \in X} I(x, y) \otimes B^\sim(y) = H(B)(x).
 \end{aligned}$$

The proofs of (4) and (5) are trivial. □

Obviously, if $A^\sim \subseteq B^\sim$, we have $N(A) \subseteq N(B), H(A) \subseteq H(B)$.

If $A \subseteq B$, then for every $x \in X$, we have $A(x) \leq B(x)$, i.e., $A(x) \rightarrow B(x) = 1$. By this, we obtain $(A(x) \rightarrow B(x))^\sim = 1^\sim = 1$, thus $A^\sim(x) \rightarrow B^\sim(x) = 1$, so $S(A^\sim, B^\sim) = 1$, that is, $A^\sim \subseteq B^\sim$.

By the above analysis, if $A \subseteq B$, we also have $N(A) \subseteq N(B), H(A) \subseteq H(B)$.

Proposition 2. Suppose $\{A_i \mid i \in I\} \subseteq L^X$ and $A, B \in L^X$, we have

$$(1) \bigvee_{i \in I} N(A_i) \subseteq N(\bigvee_{i \in I} A_i), \quad (2) H(\bigwedge_{i \in I} A_i) \subseteq \bigwedge_{i \in I} H(A_i).$$

Proof. (1) Suppose $\{A_i \mid i \in I\} \subseteq L^X$, since $\forall i \in I, A_i \subseteq \bigvee_{i \in I} A_i$, by Proposition 1, we have $N(A_i) \subseteq N(\bigvee_{i \in I} A_i)$, thus $\bigvee_{i \in I} N(A_i) \subseteq N(\bigvee_{i \in I} A_i)$.

(2) Similarity. □

Proposition 3. Suppose \mathbf{L} is distributive, and $(\bigwedge_{i \in I} a_i)^\sim = \bigwedge_{i \in I} a_i^\sim$, then P is closed for \bigwedge , that is, suppose $\{N(A_i)\} \subseteq P$, then $N(\bigwedge_i A_i) = \bigwedge_i N(A_i)$.

Proof. Since for every $x \in X$, we have

$$\begin{aligned}
 N(\bigwedge_i A_i)(x) &= \bigwedge_{y \in X} (\bigwedge_i A_i)^\sim(y) \vee I^*(y, x) = \bigwedge_i [\bigwedge_{y \in X} A_i^\sim(y) \vee I^*(y, x)] \\
 &= \bigwedge_i N(A_i)(x).
 \end{aligned}$$
□

Proposition 4. If $(\bigvee_{i \in I} a_i)^\sim = \bigvee_{i \in I} a_i^\sim$, then Q is closed for \bigvee , that is, suppose $\{H(A_i)\} \subseteq Q$, then $H(\bigvee_i A_i) = \bigvee_i H(A_i)$.

Proof. Since for every $x \in X$, we have

$$H(\bigvee_i A_i)(x) = \bigvee_{y \in X} (\bigvee_i A_i)^\sim(y) \otimes I(y, x) = \bigvee_i [\bigvee_{y \in X} A_i^\sim(y) \otimes I(y, x)] = \bigvee_i H(A_i)(x).$$
□

Proposition 5. Suppose I is reflexive, then we have $N(A) \subseteq A^\sim \subseteq A, A^\sim \subseteq H(A)$ for every $A \in L^X$.

Proof. For every $A \in L^X$, we have to show $N(A) \subseteq A^\sim$, and $A^\sim \subseteq H(A)$. It is enough to prove $N(A)(x) \leq A^\sim(x)$, and $A^\sim(x) \leq H(A)(x)$ holds for every $x \in X$.

The first inequality holds, since

$$N(A)(x) = \bigwedge_{y \in X} A^\sim(y) \vee I^*(y, x) \leq A^\sim(x) \vee I^*(x, x) \leq A^\sim(x) \leq A(x).$$

The second inequality holds, since

$$H(A)(x) = \bigvee_{y \in X} I(x, y) \otimes A^\sim(y) \geq I(x, x) \otimes A^\sim(x) = A^\sim(x).$$
□

Proposition 6. *If $a^\sim = a$, and I is reflexive and transitive, then H is an \mathbf{L} -closure operator.*

Proof. For $A, B \in L^X$, by Propositions 1, 5, we obtain $A = A^\sim \subseteq H(A)$, $S(A, B) \leq S(H(A), H(B))$. Thus we have $H(A) \subseteq HH(A)$. We show H is an \mathbf{L} -closure operator, by Definition 3, it is enough to prove $HH(A) \subseteq H(A)$, which is equivalent to $HH(A)(x) \leq H(A)(x)$ which holds for every $x \in X$. It is valid, since

$$\begin{aligned} HH(A)(x) &= \bigvee_{y \in X} I(x, y) \otimes H(A)(y) = \bigvee_{y \in X} I(x, y) \otimes \bigvee_{z \in X} I(y, z) \otimes A^\sim(z) \\ &= \bigvee_{y \in X} \bigvee_{z \in X} I(x, y) \otimes I(y, z) \otimes A^\sim(z) \leq \bigvee_{z \in X} I(x, z) \otimes A^\sim(z) = H(A)(x). \quad \square \end{aligned}$$

Note that since I is not symmetric, H may be viewed as an (right or left) \mathbf{L} -closure operator. When I is an \mathbf{L} -equivalence relation, H is an \mathbf{L} -closure operator defined in [1]. But if $a^\sim \neq a$, H is not an \mathbf{L} -closure operator, since $A \subseteq H(A)$ does not hold in general.

Proposition 7. *Suppose \mathbf{L} is prelinearity, and I is symmetric, then $N(A^*) \subseteq H^*(A)$ holds for every $A \in L^X$. Moreover, if $\wedge = \otimes$, $N(A^*) = H^*(A)$ holds.*

Proof. For every $A \in L^X, x \in X$, we have

$$\begin{aligned} N(A^*)(x) &= \bigwedge_{y \in X} A^{*\sim}(y) \vee I^*(y, x) = \bigwedge_{y \in X} [A(y) \rightarrow 0]^\sim \vee [I(y, x) \rightarrow 0] \\ &\leq \bigwedge_{y \in X} [A^\sim(y) \rightarrow 0^\sim] \vee [I(y, x) \rightarrow 0] = \bigwedge_{y \in X} [A^\sim(y) \rightarrow 0] \vee [I(y, x) \rightarrow 0] \\ &= \bigwedge_{y \in X} [A^\sim(y) \wedge I(y, x)] \rightarrow 0 \quad (\text{since } \mathbf{L} \text{ is prelinearity}) \\ &\leq \bigwedge_{y \in X} [A^\sim(y) \otimes I(y, x)] \rightarrow 0 = [\bigvee_{y \in X} A^\sim(y) \otimes I(x, y)] \rightarrow 0 \\ &= H(A)(x) \rightarrow 0 = H^*(A)(x). \end{aligned}$$

So $N(A^*) \subseteq H^*(A)$ holds for every $A \in L^X$. □

If $\wedge = \otimes$, then $N(A^*) = H^*(A)$ holds by the above proof.

Note that a Heyting algebra is a residuated lattice satisfying: $\wedge = \otimes$, see [1] p.32.

4 Generalized Framework

On the one hand, in formal concept analysis, a binary relation induces a Galois connection between two universes (one set is the collection of objects, the other set is the collection of properties, or attributes). On the other hand, in the study of modal logics, I. Düntsch and G. Gediga defined modal-style operators based on a binary relation [6], and introduced a kind of concept lattice. Yao introduced another kind of concept lattice, and compared the roles of different concept in data analysis. Thus a binary relation serves a common basis for rough sets and formal concept analysis. We generalized the three kinds of concept lattice in \mathbf{L} -rough sets, see [4].

In the section, we will extend the two rough approximation operators with hedges on two universes, and investigate some of their properties.

Given an \mathbf{L} -formal context (X, Y, I) , let \sim_X, \sim_Y be hedges, $I(x, y) \in L^{X \times Y}$, by means of the two rough operators, we define a pair of approximation operators: $L^X \rightarrow L^Y$, for every $A \in L^X$, and $y \in Y$,

$$A^\square(y) = \bigwedge_{x \in X} A^{\sim_X}(x) \vee I^*(x, y), \quad A^\diamond(y) = \bigvee_{x \in X} I(y, x) \otimes A^{\sim_X}(x)$$

and another pair of approximation operators: $L^Y \rightarrow L^X$, for every $B \in L^Y$, and $x \in X$, $B^\square(x) = \bigwedge_{y \in Y} B^{\sim_Y}(y) \vee I^*(y, x)$, $B^\diamond(x) = \bigvee_{y \in Y} I(x, y) \otimes B^{\sim_Y}(y)$

Proposition 8. *Suppose $A_1, A_2 \in L^X, B_1, B_2 \in L^Y$, we have,*

$$S(A_1^{\sim_X}, A_2^{\sim_X}) \leq S(A_1^\square, A_2^\square), S(A_1^{\sim_X}, A_2^{\sim_X}) \leq S(A_1^\diamond, A_2^\diamond)$$

$$S(B_1^{\sim_Y}, B_2^{\sim_Y}) \leq S(B_1^\square, B_2^\square), S(B_1^{\sim_Y}, B_2^{\sim_Y}) \leq S(B_1^\diamond, B_2^\diamond).$$

Proof. It holds follows from Proposition 1. □

and

Proposition 9. *Suppose \mathbf{L} is distributive, and $(\bigwedge_{i \in I} a_i)^\sim = \bigwedge_{i \in I} a_i^\sim$,*

$(\bigvee_{i \in I} a_i)^\sim = \bigvee_{i \in I} a_i^\sim, A_i \in L^X, B_i \in L^Y$, we have

$$\bigvee_i [A_i]^\diamond = [\bigvee_i A_i]^\diamond, \bigwedge_i [A_i]^\square = [\bigwedge_i A_i]^\square, \bigvee_i [B_i]^\diamond = [\bigvee_i B_i]^\diamond, \bigwedge_i [B_i]^\square = [\bigwedge_i B_i]^\square.$$

Proof. See Proposition 3, 4. □

5 Related to Concept Lattice Induced by R. Bělohlávek

In the section, suppose \mathbf{L} satisfies the prelinearity axiom and $\otimes = \wedge$, we investigate the relation between the rough operators with hedges and the fuzzy concept lattice with hedges in [3] introduced by R. Bělohlávek, et.al.

On the other hand, for an \mathbf{L} -relation $I \in L^{X \times X}$, the negation of I is defined as $I^*(x, y) = I(x, y) \rightarrow 0$.

As introduced in Section 2.3, suppose X and Y are two sets with \mathbf{L} -equalities \approx_X and \approx_Y , respectively; given an \mathbf{L} -relation I, I^* is also an \mathbf{L} -relation between X and Y which is compatible with respect to \approx_X and \approx_Y . Then $\langle X^{\sim_X}, Y^{\sim_Y}, I^* \rangle$ is called a formal \mathbf{L} -context. A pair $\langle \uparrow_{I^*}, \downarrow_{I^*} \rangle$ of mappings was defined as:

$$\uparrow_{I^*} : L^X \rightarrow L^Y, \text{ for } A \in L^X, A^{\uparrow_{I^*}}(y) = \bigwedge_{x \in X} A^{\sim_X}(x) \rightarrow I^*(x, y).$$

$$\downarrow_{I^*} : L^Y \rightarrow L^X, \text{ for } B \in L^Y, B^{\downarrow_{I^*}}(x) = \bigwedge_{y \in Y} B^{\sim_Y}(y) \rightarrow I^*(x, y).$$

$\langle A, B \rangle$ is called a concept in $\langle X^{\sim_X}, Y^{\sim_Y}, I^* \rangle$, if $A^{\uparrow_{I^*}} = B, B^{\downarrow_{I^*}} = A$. Then A and B are called an extent and an intent of $\langle A, B \rangle$, respectively.

$\beta(X^{\sim_X}, Y^{\sim_Y}, I^*) = \{ \langle A, B \rangle \mid \langle A, B \rangle \text{ is a concept} \}$ is called a formal concept lattice in $\langle X^{\sim_X}, Y^{\sim_Y}, I^* \rangle$. A is also called a concept of objects, B is called a concept of attributes.

By the above analysis, we obtain the following proposition,

Proposition 10. *Suppose \mathbf{L} satisfies the prelinearity axiom and $\otimes = \wedge$, for $A \in L^X, B \in L^Y$, we have,*

$$(1) A^{*\square} = A^{\uparrow_{I^*}}, B^{\diamond*} = B^{\downarrow_{I^*}}, \quad (2) B^{*\square} = B^{\uparrow_{I^*}}, A^{\diamond*} = A^{\downarrow_{I^*}}.$$

Proof. (1) For any $A \in L^X, B \in L^Y$, and $x \in X, y \in Y$, we have

$$\begin{aligned} A^{*\square}(y) &= \bigwedge_{x \in X} (A^{\sim x})^*(y) \vee I^*(x, y) = \bigwedge_{x \in X} [A^{\sim x}(y) \rightarrow 0] \vee [I(x, y) \rightarrow 0] \\ &= \bigwedge_{x \in X} [(A^{\sim x}(y) \wedge I(x, y)) \rightarrow 0] = \bigwedge_{x \in X} [A^{\sim x}(y) \otimes I(x, y) \rightarrow 0] \\ &= \bigwedge_{x \in X} A^{\sim x}(y) \rightarrow [I(x, y) \rightarrow 0] = \bigwedge_{x \in X} A^{\sim x}(y) \rightarrow I^*(x, y) = A^{\uparrow_{I^*}}(y), \end{aligned}$$

and

$$\begin{aligned} B^{\diamond*}(x) &= [\bigvee_{y \in Y} B^{\sim y}(y) \otimes I(x, y)] \rightarrow 0 = \bigwedge_{y \in Y} [B^{\sim y}(y) \otimes I(x, y) \rightarrow 0] \\ &= \bigwedge_{y \in Y} B^{\sim y}(y) \rightarrow [I(x, y) \rightarrow 0] = \bigwedge_{y \in Y} B^{\sim y}(y) \rightarrow I^*(x, y) = B^{\downarrow_{I^*}}(x). \end{aligned}$$

Thus we have $A^{*\square} = A^{\uparrow_{I^*}}, B^{\diamond*} = B^{\downarrow_{I^*}}$.

$$(2) \text{ Similarly, we also have } B^{*\square} = B^{\uparrow_{I^*}}, A^{\diamond*} = A^{\downarrow_{I^*}}. \quad \square$$

That is to say, $*\square$ and $\diamond*$ form an \mathbf{L} -Galois connection between $\langle\langle X^{\sim x}, \approx \rangle\rangle \preceq$ and $\langle\langle Y^{\sim y}, \approx \rangle\rangle \preceq$ induced by a binary relation $I^*(x, y)$.

By Proposition 10, we obtained the relation between the rough approximation operators \square, \diamond and Galois connection \uparrow_I, \downarrow_I in the theory of formal concept analysis.

6 Conclusion

In the paper, we introduced the notion of two rough approximation operators with hedges, investigated some of their properties, which corresponds to the fuzzy concept lattice with hedges introduced by R. Bělohlávek, et. al.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 10771056) and the Doctoral Research Foundation of Shandong University of Technology(2008KQ05).

References

1. Bělohlávek, R.: Fuzzy relational systems: Foundations and Principles. Kluwer, New York (2002)
2. Bělohlávek, R., Funioková, T., Vychodil, V.: Galois connections with hedges. In: Proc. IFSA 2005 Wolrld Congress, Beijing, China, July 28-31, vol. II, pp. 1250–1255. Springer, Heidelberg (2005)

3. Bělohávek, R., Vychodil, V.: Reducing the size of fuzzy concept lattices by hedges. In: FUZZ-IEEE 2005, The IEEE International Conference on Fuzzy Systems, Reno, Nevada, USA, May 22-25, pp. 663–668 (2005)
4. Chen, X.Y.: Concept lattice in L-rough sets. In: The Sixth International Symposium on Neural Networks (ISNN 2009), Wuhan (P. R. China), May 26-29. LNCS, vol. 5552, pp. 50–59. Springer, Heidelberg (2009)
5. Chen, X.Y., Li, Q.G.: Construction of rough approximations in fuzzy setting. *Fuzzy Sets and Systems* 158, 2641–2653 (2007)
6. Düntsch, I., Gediga, G.: Approximation operators in qualitative data analysis. In: de Swart, H., Orłowska, E., Schmidt, G., Roubens, M. (eds.) *Theory and Applications of Relational Structures as Knowledge Instruments*. LNCS, vol. 2929, pp. 214–230. Springer, Heidelberg (2003)
7. Goguen, J.: L-fuzzy sets. *J. Math. Anal. Appl.* 18, 145–174 (1967)
8. Hájek, P.: On very true. *Fuzzy sets and systems* 124, 329–333 (2001)
9. Hájek, P.: *Metamathematics of Fuzzy Logic*. Kluwer, Dordrecht (1998)
10. Kortelainen, J.: On relationship between modified sets, topological spaces and rough sets. *Fuzzy Sets and Systems* 61, 91–95 (1994)
11. Lin, T.Y.: Topological and fuzzy rough sets. In: Slowinski, R. (ed.) *Decision Support by Experience-Application of the Rough Sets Theory*, pp. 287–304. Kluwer Academic Publishers, Dordrecht (1992)
12. Morsi, N.N., Yakout, M.M.: Axiomatics for fuzzy rough set. *Fuzzy Sets and Systems* 100, 327–342 (1998)
13. Pawlak, Z.: Rough sets. *International Journal of Computer and Information Science* 11, 341–356 (1982)
14. Qin, K., Pei, Z.: On the topological properties of fuzzy rough sets. *Fuzzy Sets and Systems* 151, 601–613 (2005)
15. Radzikowska, A.M., Kerre, E.E.: Fuzzy rough sets based on residuated lattices. In: Peters, J.F., Skowron, A., Dubois, D., Grzymała-Busse, J.W., Inuiguchi, M., Polkowski, L. (eds.) *Transactions on Rough Sets II*. LNCS, vol. 3135, pp. 278–296. Springer, Heidelberg (2004)
16. Radzikowska, A.M., Kerre, E.E.: On L-fuzzy rough sets. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) *ICAISC 2004*. LNCS (LNAI), vol. 3070, pp. 526–531. Springer, Heidelberg (2004)
17. Shao, M.W., Liu, M., Zhang, W.X.: Set approximations in fuzzy formal concept analysis. *Fuzzy Sets and Systems* 158(23), 2627–2640 (2007)
18. Wu, W.Z., Zhang, W.X.: Constructive and axiomatic approaches of fuzzy approximation operators. *Information Science* 159, 233–254 (2004)
19. Yao, Y.Y.: A comparative study of formal concept analysis and rough set theory in data analysis. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) *RSCCTC 2004*. LNCS, vol. 3066, pp. 59–68. Springer, Heidelberg (2004)
20. Yao, Y.Y.: Two views of the theory of rough sets in finite universes. *International Journal of Approximate Reasoning* 15, 291–317 (1996)

An Evolutionary Algorithm with Lower-Dimensional Crossover for Solving Constrained Engineering Optimization Problems

Yulong Shi*, Sanyou Zeng, Bo Xiao, Yang Yang, and Song Gao

School of Computer Science, Research Centre for Space Science & Technology, China
University of Geosciences, Wuhan, 430074, China
windog18@163.com, sanyou-zeng@263.net

Abstract. This paper proposes an evolutionary algorithm with lower-dimensional-search crossover for constrained engineering optimization problems. Crossover operator of the algorithm searches a lower dimensional space determined by the parent points. It is favorable to enhance the performance of the algorithm. The algorithm has been used to solve 4 engineering optimization problems with constraints. The results show the performance of the proposed algorithm is better than that of some newly proposed algorithms in solving the 4 engineering optimization problems. Especially, for the Pressure Vessel Problem, its result is much better than that yielded by other known algorithms. The proposed algorithm is simple and readable as well.

1 Introduction

As surveyed by [1], evolutionary algorithms (EAs) for constrained optimization can be classified into several categories by the way the constraints are treated:

- 1) Constraints are only used to test whether a search point is feasible or not [2], [3].
- 2) Constraint violation, the sum of all constraint functions' violation, is combined with the objective function. Approaches in this category are called penalty function methods.
- 3) The constraint violation and the objective function are used and optimized separately [4]–[9]. This method adopts a lexicographic order, in which the constraint violation precedes the objective function, with the relaxation of constraints. The method can optimize such problems with equality constraints effectively through the relaxation of constraints. Deb [4] proposed a method, which used an extended objective function and realized the lexicographic ordering. Runarsson and Yao [5] proposed a stochastic ranking method based on ES using a stochastic lexicographic order to ignore constraint violations with some probability. These methods have been successfully applied to solve various problems.

* Corresponding author.

A new evolutionary algorithm which uses lower dimensional crossover is proposed in this paper. Crossover operator is used to search a lower dimensional space defined by the parents. The lower-dimensional-search crossover operator is favorable to propagate promising offspring. The remainder of this paper is organized as follows: The new evolutionary algorithm is described in II. The description of the new algorithm for solving optimization problems with constraints is given in III. VI contains the validity of the new technique for solving four engineering design problem, and in V concludes.

2 Framework of the New EA (LDNSEA)

Framework of the New EA (LDNSEA)

In this paper, the individual $\vec{x} = (x_1, x_2, \dots, x_n)$ is represented with real-code, where x_j denotes the j th gene. l_j, u_j denote the lower boundary and the upper boundary of x_j ($j=1,2,\dots,n$);

The framework of the new EA with lower-dimensional-search crossover is as follows:

Algorithm 1 framework of the new EA with lower-dimensional-search crossover

Step1 Randomly create a population P of size N. Set generation counter t as 0

REPEAT

Step2 FOR individual \vec{x}^i of population P

Step2.1 Execute breeding operator for \vec{x}^i with output \vec{s} (Algorithm 2)

Step2.2 Compare \vec{s} with \vec{x}^i . If \vec{s} is not worse than \vec{x}^i , then \vec{x}^i is replaced by \vec{s} .

END FOR

Step3 Generation t = t+1

UNTIL evolution ends

B. Breeding Operator

The breeding operator for $\vec{x}^i = (x_1^i, x_2^i, \dots, x_n^i)$ includes three operators: crossover, copy and mutation, which all base on genes. The details of the operator are as follows:

Algorithm 2 Breeding operator for $\vec{x}^{(i)}$.

Step1 Randomly choose M different best individuals $\vec{b}^1, \vec{b}^2, \dots, \vec{b}^M$ from population P except \vec{b}^i .

Step2 **Execute crossover operator:** Randomly create a_1, a_2, \dots, a_M , with $a_1 + a_2 + \dots + a_M = 1$ in the range $-1 \leq a_1, a_2, \dots, a_M \leq 1$.

Let
$$\vec{s}' \leftarrow a_1 \vec{b}^1 + a_2 \vec{b}^2 + \dots + a_M \vec{b}^M \tag{1}$$

Denote $\vec{s}' = (s'_1, s'_2, \dots, s'_n)$

- Step3 Execute copy operator on \vec{s} : For each gene position k ($1 \leq k \leq n$) $s_k' \leftarrow x_k^i$
 With probability p_c , yielding $\vec{s}'' = (s_1'', s_2'', \dots, s_n'')$
- Step4 Execute mutation operator \vec{s}'' : For each gene position k ($1 \leq k \leq n$)
 $s_k'' \leftarrow \text{random}(l_k, u_k)$ with probability p_m , yielding $\vec{s} = (s_1, s_2, \dots, s_n)$.
- Step5 Output $\vec{s} = (s_1, s_2, \dots, s_n)$

The M random numbers a_1, a_2, \dots, a_M in step2 are created in the following way:

Algorithm 3 Create M random numbers $a_1, a_2 \dots a_M$

Step 1 $a_1 = \text{rand}()$; $a_2 = \text{rand}()$; ...; $a_M = \text{rand}()$;

Step 2 $\text{sum} = a_1 + a_2 + \dots + a_M$;

Step 3 $a_1 = a_1 / \text{sum}$; $a_2 = a_2 / \text{sum}$; ...; $a_M = a_M / \text{sum}$;

Step4 $a_1 = (M+1)a_1 - 1$; $a_2 = (M+1)a_2 - 1$; ...; $a_M = (M+1)a_M - 1$;

($\text{rand}()$ is a random function);

The step3, step4 is base on a linear operate so that the random can't be lose. On the other way, it can make a_i in the range of $(-1, 1)$ and keep the sum of a_i to be 1.

The \vec{s}' in the Equation 1 is generated for the crossover in the algorithm 3

With Equation 1 and $a_1 + a_2 + \dots + a_M = 1$, we have

$$a_1(s' - \vec{b}) + a_2(s' - \vec{b}^1) + \dots + a_M(s' - \vec{b}^m) = 0$$

That is, the offspring \vec{s}' of the crossover stays in the linear space defined by the M parents $\vec{b}^1, \vec{b}^2, \dots, \vec{b}^m$. The dimension of the space is smaller than $M-1$. The constraints, $-1 \leq a_1, a_2, \dots, a_M \leq M$, constrain \vec{s}' in a neighborhood of the M parents.

Therefore, \vec{s}' stays in a neighborhood of the M parents with smaller than $M-1$ dimensions no matter how many dimensions the decision space of the optimization problem has. We know the fast convergence of the gradient algorithm is due to its linear search along the gradient direction. Therefore, the new algorithm should converge fast with higher dimensional decision space especially for optimization problems.

3 Application in Constrained Optimization

The comparison of two solutions in Step 2.2 of Algorithm 1 depends on problems to be solved. Here we discuss the comparison operator for constrained optimization.

General Formulation of Constrained Optimization Problem for Minimization

A general constraints optimization problem is described as in Equation 2:

$$\begin{aligned}
 & \text{Minimize } f(\vec{x}) \\
 & \text{Subject to} \\
 & \quad g_i(\vec{x}) \leq 0 \quad i = 1, \dots, q \\
 & \quad h_i(\vec{x}) = 0 \quad i = q + 1, \dots, m \\
 & \quad X = \{ \vec{x} = (x_1, x_2, \dots, x_n) \mid l_i \leq x_i \leq u_i \} \\
 & \quad \vec{l} = (l_1, l_2, \dots, l_n) \\
 & \quad \vec{u} = (u_1, u_2, \dots, u_n)
 \end{aligned} \tag{2}$$

Where \vec{x} is the decision vector, X denotes the decision space, \vec{l} and \vec{u} denote the upper bounds and lower bounds of the decision space.

Usually equality constraints are transformed into inequalities constraints in such form:

$$|h_i(\vec{x})| - \delta \leq 0, i = q + 1, \dots, m \tag{3}$$

A solution \vec{x} is regarded as feasible if $g_i(\vec{x}) \leq 0, i=1, \dots, q$, and $|h_i(\vec{x})| - \delta \leq 0, i=q+1, \dots, m$. In this paper δ is set as 0.0001.

Dynamic Format of the Constraint Optimization Problem

Literature [1] proposed a ϵ -Constrained method to deal with equality constraints. Here we interpret this method in a dynamic way. The constrained optimization problem is transformed into the following dynamic optimization problem:

When $\lim_{t \rightarrow \infty} \epsilon_{hi}(t) = 0$ with $t \rightarrow \infty$, the problem $Q(t)$ in Equation 4 will approach the problem Q in Equation 2. Let denote $\lim_{t \rightarrow \infty} Q(t) = Q$. Suppose that t denote the generation of an evolutionary algorithm. Using the evolutionary algorithm to solve problem $Q(t)$ at generation t , then when t approach to ∞ , the evolutionary algorithm will solve the problem Q finally. Based on this idea, we use evolutionary algorithm to solve problem Q in Equation 2.

$$\begin{aligned}
 & Q(t) \\
 & \text{Minimize } f(\vec{x}) \\
 & \text{Subject to} \\
 & \quad g_i(\vec{x}) \leq 0 \quad i = 1, \dots, q \\
 & \quad |h_i(\vec{x})| \leq \epsilon_{hi}(t) \quad i = q + 1, \dots, m \\
 & \quad X = \{ \vec{x} = (x_1, x_2, \dots, x_n) \mid l_i \leq x_i \leq u_i \} \\
 & \quad \vec{l} = (l_1, l_2, \dots, l_n) \\
 & \quad \vec{u} = (u_1, u_2, \dots, u_n)
 \end{aligned} \tag{4}$$

Denote violation values

$$G_i(\vec{x}) = \max\{0, g_i(\vec{x})\} \quad i=1, \dots, q \tag{5}$$

$$H_i(\vec{x}, t) = \max\{\varepsilon_{hi}(t), |h_i(\vec{x})|\}, \quad i=q+1, \dots, m \tag{6}$$

$\varepsilon_{hi}(t)$ is set as follow:

$$\varepsilon_{hi}(t) = \frac{\max_{j \in \{1, 2, \dots, N\}} \{H_i(\vec{x}^{(j)}, 0)\}}{C^t} \tag{7}$$

The constant $C (> 1)$ is used to control the relaxation of the constraints.

$\max_{j \in \{1, 2, \dots, N\}} \{H_i(\vec{x}^{(j)}, 0)\}$ is the max value of the equality constraint violation

$H_i(\vec{x}, 0)$ in initial population.

Solution Comparisons for the Problem $Q(t)$

Comparison operator here uses three criteria basing on feasibility originally proposed by Deb [2]:

- Between 2 feasible solutions, the one with better objective value wins.
- If one is feasible and the other is infeasible, the feasible solution wins.

If both solutions are infeasible, the one with lower sum of constraint violation is preferred.

In the case of both solutions are infeasible (the third case above), the sum of constraint violation is actually the 1-norm of the constraint violation vector:

$$\vec{v} = (G_1(\vec{x}), \dots, G_q(\vec{x}), H_{q+1}(\vec{x}, t), \dots, H_m(\vec{x}, t)) \tag{8}$$

$$\|\vec{v}\|_1 = \sum_{i=1}^q G_i(\vec{x}) + \sum_{i=q+1}^m H_i(\vec{x}, t) \tag{9}$$

In this paper, infinity norm is used:

$$\|\vec{v}\|_\infty = \max\{G_1(\vec{x}), \dots, G_q(\vec{x}), H_{q+1}(\vec{x}, t), \dots, H_m(\vec{x}, t)\}$$

Suppose the constraint violation vector of $\vec{x}^{(1)}$ is \vec{v}_1 , and that of $\vec{x}^{(2)}$ is \vec{v}_2 . The

detail comparison between $\vec{x}^{(1)}$ and $\vec{x}^{(2)}$ at generation t is as follows:

Algorithm 5 Comparison of $\vec{x}^{(1)}$ with $\vec{x}^{(2)}$

If $\|\vec{v}_1\|_\infty < \|\vec{v}_2\|_\infty$, $\vec{x}^{(1)}$ wins.

If $\|\vec{v}_1\|_\infty > \|\vec{v}_2\|_\infty$, $\vec{x}^{(2)}$ wins.

If $(\|\vec{v}_1\|_\infty = \|\vec{v}_2\|_\infty) \wedge (f(\vec{x}^{(1)}) < f(\vec{x}^{(2)}))$, $\vec{x}^{(1)}$ wins.
 If $(\|\vec{v}_1\|_\infty = \|\vec{v}_2\|_\infty) \wedge (f(\vec{x}^{(1)}) > f(\vec{x}^{(2)}))$, $\vec{x}^{(2)}$ wins.
 If $(\|\vec{v}_1\|_\infty = \|\vec{v}_2\|_\infty) \wedge (f(\vec{x}^{(1)}) = f(\vec{x}^{(2)}))$, $\vec{x}^{(1)}$ is indifferent to $\vec{x}^{(2)}$.

4 Numerical Experimental Result

Parameter Settings and Discussion

(1) Population size $N=50$. As is known, population size is recommended $30 \leq N \leq 100$. Here we choose $N=50$.

(2) Number of Parents M . In a case of a big M , the crossover operator may search a large space and the premise of efficiency of crossover may be broken. Therefore, the algorithm may search ineffectively and converge slowly. While in a case of a small M , the parents of the crossover may lose diversity and the algorithm may converge prematurely. Experiment shows that $3 \leq M \leq 7$ is a proper selection. In this paper, $M=5$.

(3) Crossover probability is 1 (cf. Step 2 of Algorithm 2), copy probability $p_c=0.05$ (cf. Step 3 of Algorithm 2) and mutation probability $p_m=0.05$ (cf. Step 4 of Algorithm 2).

(4) Max function evaluations: 50000.

(5) Constraints relaxation controlling parameter $C=1.02$. Parameter C is related to the max evolutionary generation. The evolutionary generation of the new constrained EA is calculated as follow:

Max evolutionary generation = (Max function evaluations) / (Population size) = $50000 / 50 = 1000$.

For preserving $\mathcal{E}_{hi}(t) \rightarrow 0$ in Equation 7, we set parameter $C=1.02$ where t is the evolutionary generation.

Four Engineering Designing Problem

1. The Pressure Vessel Design

This problem [4]–[7] corresponds to the weight minimization of a cylindrical pressure vessel with two spherical heads. There are four designing variables (in inches): the thickness of the pressure vessel (T_s), the thickness of the head (T_h), the inner radius of the vessel (R) and the length of the cylindrical component (L). Since there are two discrete variables (T_s and T_h) and two continuous variables (R and L), one has a nonlinearly constrained mixed

Discrete continuous optimization problem. The bounds of the designing variables are $0.0625 \leq T_s, T_h \leq 5$ (in constant steps of 0.0625) and $10 \leq R, L \leq 200$. The weight to be minimized, and the constraints are given by:

In this problem, we suppose T_s to be x_1 , T_h to be x_2 , R to be x_3 , and L to be x_4 .

$$\text{Minimize } f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

Subject To:

$$g_1(x) = 0.0193x_3 - x_1 \leq 0 \qquad g_2(x) = 0.00954x_3 - x_2 \leq 0$$

$$g_3(x) = 750 \times 1728 - \pi x_3^2 x_4 - \frac{4}{3} \pi x_3^3 \leq 0 \qquad g_4(x) = x_4 - 240 \leq 0$$

$$X_1 \in [0.0625, 5.0] \quad X_2 \in [0.0625, 5.0] \quad X_3 \in [10.0, 300.0] \quad X_4 \in [10.0, 300.0].$$

Table 1 presents a comparison of results obtained with different algorithms. All algorithms use 80,000 times of function evaluations, except AIS [8], which uses 150,000 times and LDNSEA 50,000 times. The best solution was found by the proposed LDNSEA, and corresponds to a final weight equal to 5804. 38767986. Table 2 displays the final solutions which are all feasible.

Table 1. Design variables found for the best solutions for the pressure vessel design

	AIS[8]	AIS-GA[1]	AIS-GA ^C [1]	AIS-GA[14]	LDNSEA
x1	0.8125	0.8125	0.8125	0.8125	0.727490929353616
x2	0.4375	0.4375	0.4375	0.4375	0.359648573369611
x3	42.0870	42.0950	42.0950	42.0973	37.699011883607
x4	176.7791	176.7031	176.6797	176.6509	240.0
f1(x)	6061.1229	6060.3677	6060.138	6059.8546	5804. 38767986

Table 2. Values of the weight found for the pressure vessel design

	Best	Median	Average	St.Dev	Worst	fr	Ne
AIS[8]	6061.123	-	6734.085	-	7568.060	-	150,000
AIS-GA[1]	6060.368	-	6743.872	-	7346.750	-	80,000
AIS-GAC[1]	6060.138	-	6385.942	-	6845.496	-	80,000
AIS-GA[14]	6059.855	6426.710	6545.126	1.24E+2	7388.160	50	80,000
LDNSEA	5804. 3876	-	5804. 3876	-	5804. 3876	50	50,000

Table 3. Design variables found for the best solutions for the speed reducer design

	ES[10]	AIS-GA[9]	AIS-GAC [8]	AIS-GA[14]	LDNSEA
x1	3.506163	3.500001	3.500000	3.500001	3.49993
x2	0.700831	0.700000	0.700000	0.700000	0.70000
x3	17	17	17	17	17
x4	7.460181	7.300019	7.300001	7.300008	7.3
x5	7.962143	7.800013	7.800000	7.800001	7.71517
x6	3.362900	3.350215	3.350215	3.350215	3.350215
x7	5.308949	5.286684	5.286684	5.286683	5.28652
W	3025.0051	2996.3494	2996.3484	2996.3483	2994.35354

Table 4. Values found for the speed reducer design

	Best	Median	Average	St.Dev	Worst	fr	Ne
ES[10]	3025.0051	-	3088.7778	-	3078.5918	-	36,000
AIS-GA[9]	2996.3494	2996.356	2996.3643	4.35E-3	2996.6277	50	36,000
AIS-GA ^C [8]	2996.3484	2996.3484	2996.3484	1.46E-6	2996.3486	50	36,000
AIS-GA[14]	2996.3483	2996.3495	2996.3501	7.45E-3	2996.3599	50	36,000
LDNSEA	2994.3535	-	2994.3535	0	2994.3535	50	36,000

Table 5. Design variables found for the best solutions for the tension spring design

	AIS-GA[9]	AIS-GA ^C [8]	AIS-GA[14]	LDNSEA
x1	11.852177	11.329555	11.6611924	8.543881
x2	0.347475	0.356032	0.3505298	0.374472
x3	0.051302	0.051661	0.0514305	0.05
f(x)	0.012668	0.012666	0.012666	0.009871

Table 6. Values found for tension/compression spring design

	Best	Median	Average	St.Dev	Worst	fr	Ne
AIS-GA[9]	0.012668	–	0.013481	–	0.016155	–	36,000
AIS-GA ^c [8]	0.012666	–	0.012974	–	0.013880	–	36,000
AIS-GA[14]	0.012666	0.01289 2	0.013131	6.28E- 4	0.015318	50	36,000
LDNSEA	0.009871	0.00987 1	0.009871	0	0.009871	50	36,000

Table 7. Design variables found for the best solutions for the welded beam design

	AIS-GA[9]	AIS-GA ^c [8]	AIS-GA[14]	LDNSEA
x1	0.2443243	0.2443857	0.2434673	0.2058296
x2	6.2201996	6.2183037	6.2507296	3.4683377
x3	8.2914640	8.2911650	8.2914724	9.0366239
x4	0.2443694	0.2443875	0.2443690	0.2057296
f(x)	2.381246	2.38122	2.38335	1.7247170

Table 8. Values found for the cost of the welded beam design

	Best	Median	Average	St.Dev	Worst	fr	Ne
AIS-GA[9]	2.38125	–	2.59303	–	3.23815	–	320,000
AIS-GA ^c [8]	2.38122	–	2.38992	–	2.41391	–	320,000
AIS-GA[14]	2.38335	2.92121	2.99298	2.02E- 1	4.05600	50	320,000
LDNSEA	1.7247170	1.7247170	1.7247170	0	1.7247170	50	50,000

Minimize $f(x) = (x_1 + 2)x_2 x_3^2$

Subject To:

$$g_1(x) = 1 - \frac{x_2^3 x_1}{71785 x_3^4} \leq 0 \quad g_2(x) = \frac{4x_2^2 - x_3 x_2}{12566(x_2 x_3^3 - x_3^4)} + \frac{1}{5108 x_3^2} - 1 \leq 0$$

$$g_3(x) = 1 - \frac{140.45 x_3}{x_2^2 x_1} \leq 0 \quad g_4(x) = \frac{x_2 + x_3}{1.5} - 1 \leq 0$$

The number of function evaluations was set to 36,000. A comparison of results is provided in the Table 5 where the best result is found by the proposed LDNSEA, with a final volume equal to 0.009871. Table 6 shows the values found for the designing variables corresponding to the best solutions that are all feasible.

4. The Welded Beam Design

$$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}, \quad \tau' = \frac{P}{\sqrt{2}x_1 x_2}, \quad \tau'' = \frac{MR}{J}, \quad M = P(L + \frac{x_2}{2})$$

$$R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1 + x_3}{2})^2}$$

$$\sigma(x) = \frac{6PL}{x_4 x_3^2}, \delta(x) = \frac{4PL^3}{E x_3^3 x_4}, P_c(x) = \frac{4.013E \sqrt{\frac{x_3^3 x_4^6}{36}}}{L^2} \left(1 - \frac{x_3}{2L} \sqrt{\frac{E}{4G}} \right)$$

P=6000,L=14,E=30×106,G=12×106, τ_{max} =13600, δ_{max} =30000, δ_{max} =0.25.

The Table 7 shows a comparison of results where the best value found (final cost equal to 1.7247170) corresponds to LDNSEA. The Table 8 shows the design variables corresponding to the best solution. All the solutions are feasible and the number of function evaluations was set to 320,000 except that LDNSEA use 50,000 times.

5 Conclusion

(1) The crossover operator of the new algorithm searches a lower dimensional neighbour of the parents, which decreases the complexity of searching.

(2) The new algorithm finds out, up to now, the best solution for the above four illustrated engineering problems. The solutions are probably the global optimum. And with LDNSEA algorithm, the constraints of the four engineering problems are active.

(3) LDNSEA algorithm does not use gradient method. Therefore, no differentiability is demanded.

(4) LDNSEA algorithm is simple and easy to be implemented.

(5) Experimental result shows that LDNSEA algorithm is effective in solving the 4 engineering optimization problems with constraints.

Acknowledgments. This work was supported by The National Natural Science Foundation of China (Nos: 60473037, 60871021), and by the open research program of the Geological Processes and Mineral Resources (GPMR), China University of Geosciences (No. GPMR200618).

References

1. Coello Coello, C.A.: Theoretical and Numerical Constraint-Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art. *Computer Methods in Applied Mechanics and Engineering* 191(11-12), 1245–1287 (2002)
2. Michalewicz, Z., Nazhiyath, G.: Genocop III: A co-evolutionary algorithm for numerical optimization with nonlinear constraints. In: *Proceedings of the Second IEEE International Conference on Evolutionary Computation*, pp. 647–651. IEEE Press, Los Alamitos (1995)
3. El-Gallad, I., El-Hawary, M.E., Sallam, A.A.: Swarming of intelligent particles for solving the nonlinear constrained optimization problem. *Engineering Intelligent Systems for Electrical Engineering and Communications* 9(3), 155–163 (2001)
4. Deb, K.: An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering* 186(2/4), 311–338 (2000)
5. Runarsson, T.P., Yao, X.: Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation* 4(3), 284–294 (2000)
6. Mezura-Montes, E., Coello, C.A.C.: A simple multimembered evolution strategy to solve constrained optimization problems. *IEEE Trans. on Evolutionary Computation* 9(1), 1–17 (2005)
7. Venktraman, S., Yen, G.G.: A generic framework for constrained optimization using genetic algorithms. *IEEE Trans. on Evolutionary Computation* 9(4), 424–435 (2005)

8. Mezura-Montes, E., Velazquez-Reyes, J., Coello, C.A.C.: Modified Differential Evolution for Constrained Optimization. In: Proceedings of the 2006 Congress on Evolutionary Computation (CEC 2006), Canada, pp. 332–339 (2006)
9. Takahama, T., Sakai, S.: Learning fuzzy control rules by a constrained powell's method. In: Proceedings of 1999 IEEE International Fuzzy Systems Conference, Seoul, Korea, vol. 2, pp. 650–655 (1999)
10. Camponogara, E., Talukdar, S.N.: A genetic algorithm for constrained and multiobjective optimization. In: Alander, J.T. (ed.) 3rd Nordic Workshop on Genetic Algorithms and Their Applications (3NWGA), pp. 49–62. University of Vaasa, Vaasa (1997)
11. Surry, P.D., Radcliffe, N.J.: The COMOGA method: Constrained optimisation by multiobjective genetic algorithms. *Control and Cybernetics* 26(3), 391–412 (1997)
12. Runarsson, T.P., Yao, X.: Evolutionary search and constraint violations. In: Proceedings of the 2003 Congress on Evolutionary Computation, vol. 2, pp. 1414–1419. IEEE Service Center, Piscataway (2003)
13. Aguirre, H., Rionda, S.B., Coello, C.A.C., Lizárraga, G.L., Montes, E.M.: Handling constraints using multiobjective optimization concepts. *International Journal for Numerical Methods in Engineering* 59(15), 1989–2017 (2004)
14. Efrén, M., Coello, C.A.C., Ricardo, L.: Engineering optimization using a simple evolutionary algorithm. In: 15th Intl. Conf. on Tools with Art. Intelligence – ICTAI 2003, CA, USA, pp. 149–156 (2003)

Gene Regulatory Network Reconstruction of P38 MAPK Pathway Using Ordinary Differential Equation with Linear Regression Analysis

Ming Zheng, Gui-Xia Liu, Han Wang, and Chun-Guang Zhou

College of Computer Science and Technology, Jilin University, Changchun, 130012,
P.R. China

Zheng_mth@163.com, liugx@jlu.edu.cn, wanghan79@126.com,
cgzhou@jlu.edu.cn

Abstract. P38 MAPK is one of the most important central regulatory proteins that can respond extra environmental stresses. It can activate or inhibit many other genes, which can lead some disease, such as cancers or inflammations etc. We proposed a new differential equation model using linear regression analysis to calculate the weight values of the Genetic Regulatory Networks to simulate the P38 MAPK pathway in Genetic level. The results of the network are reasonable. We can investigate the P38 MAPK pathway some extra hypotheses from the result model, and provide biologists optimal designs for further experiments of disease researches.

Keywords: P38 MAPK, Genetic regulatory network, differential equation, linear regression.

1 Introduction

Mitogen activated protein kinases (MAPKs) are ubiquitously expressed in mammalian cells and always produce signals in response to various causal extra-interactions, such as growth factors, pro-inflammatory cytokines and stress [1]. MAPKs are divided into four main categories, i.e. the extracellular signal-regulated protein kinase (ERK), Big Mitogen Activated Protein Kinase 1(BMK1/ERK5), c-Jun N-terminal kinase stress-activated protein kinases (Jnk/SAPK) and p38 MAPK [2]. Among the MAPKs, P38 MAP kinase, which is found in 1993 [3], plays a key role in the transduction of signals through both protein kinases and protein phosphatases [4]. So it is so crucial for the diagnosis, treatment and prevention of various disasters, especially cancers and inflammation to investigate P38 MAP Kinase.

There are four isoforms of P38 MAPK (α , β , γ and δ) [5]. Among the isoforms, P38 α has been shown to be crucially linked to the production of pro-inflammatory cytokines [6]. And its pathway can also be found to link to some cancers [7]. Little is known about how T cell receptor (TCR) mediated signaling is coupled to the activation of the MAPK p38 in mammalian cells. Typically the MAPK p38 is activated by environmental stress such as ultraviolet radiation and osmotic shock and

by pro-inflammatory cytokines such as tumor necrosis factor (TNF), interleukin-1 (IL-1), IL-2 and IL-17.

In this paper, we use the mathematical model to simulate P38 MAPK pathway to investigate the molecular interactions between the proteins expressed by the corresponding genes in the model. Although the forms of the involved modules in the pathway level are protein and RNA etc., these modules are also gene productions. To investigate the causal relationships between all the various entities in P38 MAPK pathway in the basic level, we use genes to present the entities in the pathway. Thus genes can influence each other (activate or inhabit) through a chain of proteins and metabolites. At genetic level, it is thus legitimate, and indeed common, to consider gene-gene interactions, and these lead to the concept of genetic regulatory networks (GRN) [8]. GRN ultimately attempt to describe how genes or groups of genes interact with each other and evaluate the complex regulatory mechanisms that control the activity of genes in living cells. The reconstructed gene interaction models should be able to provide biologists a range of hypotheses explaining the results of experiments and suggesting optimal designs for further experiments for P38 MAPK in genetic level.

Gene expression profiles yield quantitative and semi-quantitative data on the concentration of protein expressed by the corresponding genes in a specific condition and time, especially ncRNA. One use of the high-throughput data is to infer or reverse-engineer GRNs among genes using various mathematical approaches, such as Boolean Networks [9], Dynamic Bayesian Networks [10], Information-theoretic models [11], and differential equation additive models [12] etc. Each of these concepts allows for several modifications, such as DDLab [13], Banjo [14], ARACNE [15], MNI [16] etc., which multiplies the number of possible models for data analysis. But the mismatch between the real mechanisms and the inference network always lead to arbitrary network structures. Therefore it is difficult to expect models inferred interaction network which has acceptable performance for any biological system. Hence, we always attempt to develop models that provide greater accuracy with respect to the real bio-system investigated. Among the high-throughput data, in particular, the time series or condition series microarrays represent important information about cellular dynamics. The series status can be the reflection of the real molecular interactions among the genes. Many approaches have been proposed to reconstruct network structures from the two series expression profiles.

For series profiles, the differential equation model is an appropriate approach for network inference. The model is represented by a set of ordinary differential equations (ODE) [12], and each ODE can be defined by Eq. (1):

$$x_i \approx \sum_j w_{ij} x_j + b_i \quad (1)$$

Where x_i is the concentration level of the protein expressed by the corresponding gene i ; w_{ij} is the weight coefficient representing the influence of gene j on the regulation of gene i , including self-regulation (when $i=j$); b_i indicates the constant output observed in the absence of regulatory inputs. We call b_i background parameter for each gene. But the series profiles obtained from experiment are in the form of finite discrete point data, so the continuous differential equation (1) should be converted into the form of discrete-serial form as follows:

$$x_i(p_k) = \sum_j^n w_{ij} x_j(p_k) + b_i \quad (2)$$

Where n is the number of genes, $j=1, 2, \dots, n$; p_k is the k -th case of the gene expression profiles, $k= 1, 2, \dots, m$; m is the number of the gene expression profiles. This function can be the interaction network without any stimulation input, i.e. the casual extra environmental stress. When the stimulation inputs enter the regulatory network described before, the concentration level of each gene production will be changed, so we can have a stimulator coefficient for each gene. And the changed model can be defined by Eq. (3):

$$x_i(p_k) = \sum_j^n w_{ij} x_j(p_k) + b_i + s_i \quad (3)$$

Where s_i represents stimulation input, such as ultraviolet radiation, osmotic shock and pro-inflammatory cytokine etc., for the expression level of gene i in the model.

In this paper, we generalize the first order differential equation regulation model by analyzing linear regression to estimate the weight value matrix W . These values in the matrix can be selected based on prior knowledge or some fitting selection algorithms. This makes the model very flexible and thus capable of covering a broad range of biological systems more adequately and specifically than previous models.

We use the proposed generalization for the differential equation regulation model to infer the P38 MAPK pathway in the genetic level. The model is available as SBML format [17] and can be visualized by Cytoscape [18]. The experiment demonstrates practical applicability of the developed approach.

2 Data

The Data we used in this paper for inferring the p38 regulatory network is in the form of gene expression profiles, which can be downloaded from Developmental Therapeutics Program of NCI [19]. As others did, the data we selected is a subset of Affymetrix U95 [20] provided by Swiss company Novartis. The values of the data are averaged from triplicate array replications. And the values are measured using different carcinoma cells, such as breast, colon, leukemia, etc., and different probes in Homo sapiens. Hence the data are condition series described in the introduction. The sub-dataset contains 18 genes which are all in the P38 MAPK pathway. And each case (gene) in the data has 59 variables (conditions, such as carcinoma cells, different tissue cells and different probes, etc.) in the data. You can download the subset data of the U95 used in this article from [21] using IE 6.0 or higher version.

The sub-data we selected are a part of Homo sapiens P38 MAPK pathway available from BioCarta [22]. And in the genetic regulatory level, the pathway can be converted to GRN like Fig. 1. The relationship between the gene productions in Fig. 1 can be found from CGAP [23] of NCI. The Fig.1 is generated using Cytoscape with unweighted Force-Directed Layout algorithm [24].

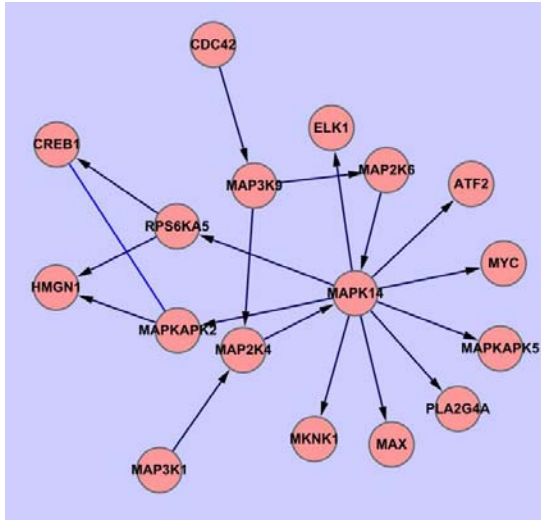


Fig. 1. The GRN level corresponding to P38 MAPK pathway. The line with no arrow on presents the undoubted line in Fig.1. And we remove the translocation of MAPK14 because it could not influence the result of the pathway model.

3 Method

We define W in Eq. (3) as the weight coefficient matrix of which the element is w_{ij} . It is the most important part of the differential equation model in GRN. The 0 value of w_{ij} means no influence between gene i and gene j ; the positive value of w_{ij} means activation influence of gene j on the regulation of gene i ; and the negative value of w_{ij} means inhibition influence of gene j on the regulation of gene i .

The criterion function which can be optimized in this differential equation model is least squared function as follows:

$$c = \sum_i^n \sum_k^m v_{ik} r_{ik}^2 \tag{4}$$

Where r_{ik} is the Residual of function $x_i(p_k)$ in Eq (3); v_{ik} is the estimated weight value of each r_{ik} . v_{ik} can be calculated in the algorithm or fixed according to the prior knowledge, such as 1 or 0.5. r_{ik} can be defined by Eq (5) as follows:

$$r_{ik} = x_i'(p_k) - \sum_j^n w_{ij} x_j'(p_k) - b_i - s_i \tag{5}$$

Where $x_i'(p_k)$ is the measured condition series of gene i at p_k , and the meaning of $x_i'(p_k)$ is the same.

We use the univariate linear regression to calculate weight value matrix and background parameters in Eq. (3), and use the hill-climbing [25] to search the minimum criterion function, i.e. Eq. (4) for each gene.

3.1 Fitting Selection of Connections

The network reconstruction using ODEs model has been described in the introduction. The links in GRN are created from the weight value units in matrix W if the corresponding weight values are significantly different from zero. So the computation of weight value matrix W is the kernel part of the algorithm. W can be calculated by minimizing the Eq. (5) with linear regression. The number of connections of the network is very large for an even medium size series profiles. In fact, GRN are believed to exhibit limited connective. Hence we must pick a good Fitting Selection Algorithm to decide which connections are in Eq. (3) for each gene. For saving much run time and inferring the network properly, we use the prior knowledge of p38 MAPK Signaling pathway. For the regulatory genes of given genes, we use the “enter” method. All the genes using “enter” fitting selection method will be the candidate regulatory genes of given gene. For the genes that are assumed to be the regulatory genes of given genes, we use the forward selection algorithm [26]. To do so, the assumed links can have more opportunities to be existed. For the genes that are not assumed to be the regulatory genes, we use the backward fitting selection method.

3.2 Procedure of the Algorithm

When the regulatory coefficients of a given gene are selected using a certain fitting selection described above, the linear regression begins as follows:

1. The model defined by Eq. (3) with all $w_{ij} = 0$ is assigned to each node, i.e. the network begins without links.
2. For each interested node, we fit the model for the data, and calculate c in Eq. (4) for the interested node.
3. We use the linear regression methodology to estimate the regression coefficients by minimizing c of Eq. (4).
4. We use the hill-climbing searching method to decide which parameters are reserved.
5. The procedure generates links until the stopping criterion is fulfilled. Otherwise we return to 2. We have implemented the following stop criteria:
 - We stop the procedure if the node with the lowest quality of fit is already linked to all the other nodes of the network. And the overall quality criterion is less than a certain limit.
 - There are no more free nodes that can reduce c value of Eq. (4). This indicates that either the algorithm has achieved the local minima or the inference model is not correct.

When we calculate the gene expression profiles, the hill-climbing searching algorithm is used to solve the one-dimensional function minimization problem described above. After the adjustment, the new calculated gene expression profiles are substituted for the old ones, and they are used in the next epoch.

4 Experiment

To show the effectiveness of the proposed method, and apply the method to investigate the P38 MAPK pathway, we use the actual DNA microarray data described in data section to reconstruct the GRN of P38.

4.1 Reconstruction of P38 GRN

In this experiment, we assume that the P38 MAPK pathway provided by BioCarta reflects biological reality. And we use the proposed algorithm in this paper to infer the GRN of P38 MAPK pathway, and summarize the parameters as Table 1:

Table 1. Summary parameters of P38 MAPK pathway in genetic level

Regulator	Relationship	Quality	Target	Constant	Coefficient
MAP2K4	aa	170.011+1.967x	MAPK14	170.011	1.967
MAP2K6	ai	170.011-4.013x	MAPK14	170.011	-4.013
CDC42	aa	3.231+0.093x	MAP3K9	3.231	0.093
MAP3K1	ai	18.432-0.197x	MAP2K4	18.432	-0.197
MAP3K9	ai	18.432-0.087x	MAP2K4	18.432	-0.087
MAPK14	ai	27.855-0.044x	PLA2G4A	27.855	-0.044
MAPK14	aa	118.184+0.032x	MKNK1	118.184	0.032
MAPK14	ai	60.781-0.08x	MAPKAPK5	60.781	-0.08
MAPK14	ai	81.416-0.019x	MAPKAPK2	81.416	-0.019
RPS6KA5	aa	67.207+0.236x	CREB1	67.207	0.236
RPS6KA5	ai	2671.481-3.811x	HMGN1	2671.481	-3.811
MAPKAPK2	ai	2671.481-4.707x	HMGN1	2671.481	-4.707
MAPK14	ai	29.288-0.05x	RPS6KA5	29.288	-0.05
MAPK14	aa	35.332+0.09x	ATF2	35.332	0.09
MAPK14	aa	73.257+0.094x	ELK1	73.257	0.094
MAPK14	ai	61.766-0.025x	MAX	61.766	-0.025
MAPK14	ai	529.89-0.288x	MYC	529.89	-0.288
MAP3K9	ai	4.935-0.119x	MAP2K6	4.935	-0.119

In the Relationship column of Table 1, aa means that the expression level of regulator gene could activate the expression level of target gene all the time under the various causal extra-interactions described in introduction; ai means that the expression level of regulator gene will activate the expression level of target gene when the concentration of regulator gene productions is less than the limit threshold, but the regulator gene will inhibit the expression level of target gene when the concentration of regulator gene productions is greater than the limit threshold. The threshold reflects the b_i , s_i , and w_{ij} in Eq. (3). Its value exists so long as the two entities are in the ai relationship, and can be defined by Eq. (6) as follows:

$$T = |constant / coefficient| \tag{6}$$

Where T presents the threshold, signs for absolute value are used in Eq. (6) because the coefficients with the a_i relationship are the negative float.

The x in the unit of Quality column presents the concentration of the regulator gene productions. The Constant and Coefficient column are derived from the Quality column. And the types of relationship are determined by the signal of the variable coefficient of the formula in the unit of Quality in Table 1.

The constant in Table 1 presents the sum of the b_i and s_i in Eq. (3). And coefficient presents the w_{ij} in Eq. (3).

We can visualize the result of GRN reconstruction using cytospace with coefficient weighted Force-Directed Layout algorithm as Fig.2:

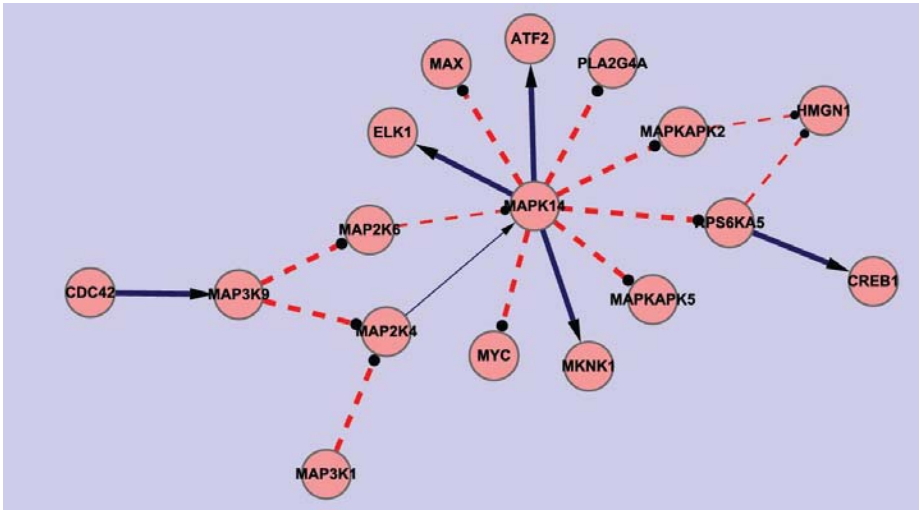


Fig. 2. The GRN of P38 MAPK pathway. The solid rectangle arrow line, from regulator gene to target gene, presents aa relationship in the Table 1. The long dash round arrow, from regulator gene to target gene, presents the a_i relationship in the Table 1.

In Fig.2, the line width, which is reverse proportional to line length, is proportional to the absolute value of coefficient in Table 1. In some certain, line width reflects the strength of the regulatory relationship between two genes. For aa lines, it reflects the activation level; for a_i lines, it reflects attenuation speed of activation level when expression level of regulator gene is lower than threshold or inhibition level when expression level of regulator gene is higher than threshold.

For the lines using the “enter” fitting selection algorithm, we calculate the parameters in Eq. (3) with the proposed mathematical model, and summarize them into Table 1. The purpose of doing so is to inspect and verify the model that already existed, and to assume extra-properties of the model. For the assumed line between CREB1 and MAPKAPK2, we use the forward selection algorithm on it, but the P value is too big to enter the model. The experiment told us that the line doesn’t exist. For the lines that don’t exist in Fig.1, we eliminated all with backward selection algorithm.

From Fig. 2, we know all the relationships between genes are activation, and this result is same as Fig.1. In the proposed simply linear model, we can see the activation relationship is not simply as we think. The relationship is related with the concentration of the regulatory gene productions.

From the constant part, we know the regulatory genes have default power to regulate the target genes. And the stimulate inputs in Eq. (3) which reflect the casual environmental stresses in macro world, such as ultraviolet radiation and osmotic shock etc., works as the regulatory roles in the model.

4.2 Software

The developed algorithm software of the proposed method was implemented with SPSS 17.0 [27], which is a short code file, freely available from [28]. You can download it and reconstruct the P38 MAPK pathway in genetic level, and adjust the code for other gene expression data to reconstruct the corresponding GRN.

5 Conclusion

We proposed a differential equation model using linear regression to calculate the coefficient parameters matrix. And apply the model to P38 MAPK pathway in genetic level. From the result Fig.2, we can find that the MAPK14 (P38 MAPK) is the hub regulator gene of the GRN. The concentration of P38 MAPK can activate many target genes. And MAPK14 can be regulated by some key genes, such as MAP2K6 and MAP2K4. The whole GRN is under the various casual environmental stresses.

By comparing the Fig.1 and Fig.2, we found that the activation relationship between genes in P38 MAPK pathway provided by BioCarta is not unchangeable. When the concentration of regulator gene productions is low enough, the concentration of target gene productions is proportional to the former; when the concentration of regulator gene productions is higher than a threshold, the function of a part of regulator genes is reversed, but the others are still unchanged in the simple model proposed in the paper. The same phenomenon can be found from precedent investigations. And we inspect and verify the phenomenon with the new mathematical model. The new mathematical model can be used for other data with the same formats as the data in this paper.

Because of the hub regulatory role of P38 MAPK, many diseases are related with it, especially cancers and inflammations. So the investigation of P38 MAPK pathway is crucial to us. In this paper, we found some different properties in genetic level of the P38 MAPK pathway. The regulatory role of P38 MAPK is not unchanged, which can reflect to the real disease phenomenon. That is, in some way, the P38 MAPK is not always play an active role or inhibit role to the disease, such as hepatitis. We can use the properties of P38 MAPK to secure some diseases of human not only activating the P38 MAPK, but also considering the concentrations of all the gene productions in genetic level according to Fig.2. And the model will provide biologists optimal designs for further experiments of disease researches.

Acknowledgement

This paper is supported by (1) the National Natural Science Foundation of China under Grant No. 60673099 and No. 60873146, (2) National High Technology Research and Development Program of China under Grant No. 2007AAO4Z114, (3) “211 Project” of Jilin University of China, (4) the Key Laboratory for Symbol Computation and Knowledge Engineering of the National Education Ministry of China, (5) the Key Laboratory for Biological Reorganization New Technology of Jilin Province of China (20082209).

The authors would like to thank the reviewers for their valuable comments.

References

1. Zhang, J., Salojin, K.V., Gao, J.X., et al.: p38 mitogen-activated protein kinase mediates signal integration of TCR/CD28 costimulation in primary murine T cells. *The Journal of Immunology* 162, 3819–3829 (1999)
2. Abe, J.I., Kusuhara, M., Richard, J., et al.: Big Mitogen-activated Protein Kinase 1 (BMK1) Is a Redox-sensitive Kinase. *The Journal of Biological Chemistry* 271(28), 16586–16589 (1996)
3. Galcheva-Gargova, Z., Derijard, B., Wu, I.H., et al.: An osmosensing signal transduction pathway in mammalian cells. *Science* 265, 806–808 (1994)
4. Davis, R.J.: The mitogen-activated protein kinase signal transduction pathway. *The Journal of Biological Chemistry* 286, 14553–14556 (1993)
5. Han, J., Lee, J.D., Bibbs, L., et al.: A MAP kinase targeted by endotoxin and hyperosmolarity in mammalian cells. *Science* 265, 808–811 (1994)
6. Nick, J.A., Young, S.K., Brown, K.K., et al.: Role of p38 Mitogen-Activated Protein Kinase in a Murine Model of Pulmonary Inflammation. *The Journal of Immunology* 164, 2151–2159 (2000)
7. Bradham, C., McClay, D.R.: p38 MAPK in development and cancer. *Cell Cycle* 5(8), 824–828 (2006)
8. Drulhe, S., Ferrari-Trecate, G., de Jong, H.: The Switching Threshold Reconstruction Problem for Piecewise-Affine Models of Genetic Regulatory Networks. *IEEE Transactions on Circuits And Systems I-Regular Papers* 29, 153–165 (2008)
9. Akutsu, T., Miyano, S., Kuhara, S.: Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. *Pacific Symposium on Biocomputing* 4, 17–28 (1999)
10. Pe’er, D., Nachman, I., Linial, M., et al.: Using bayesian networks to analyze expression data. *J.Comput. Biol.* 7, 601–620 (2000)
11. Steuer, R., Kurths, J., Daub, C.O., et al.: The mutual information: detecting and evaluating dependencies between variables. *Bioinformatics* 18(suppl. 2), 231–240 (2002)
12. D’haeseleer, P., Liang, S., Somogyi, R.: Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics* 16, 707–726 (2000)
13. Wuensche, A.: Discrete Dynamics Lab (DDLab), <http://www.ddlab.com/>
14. Yu, J., Smith, V.A., Wang, P.P., et al.: Advances to bayesian network inference for generating causal networks from observational biological data. *Bioinformatics* 20, 3594–3603 (2004)
15. Basso, K., Margolin, A.A., Stolovitzky, G., et al.: Reverse engineering of regulatory networks in human B. *Cells* 37, 382–390 (2005)

16. di Bernardo, D., Thompson, M., Gardner, T., et al.: Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks. *Nat. Biotechnol.* 23, 377–383 (2005)
17. Hucka, M., Finney, A., Bornstein, B.J., et al.: Evolving a lingua franca and associated software infrastructure for computational systems biology: the Systems Biology Markup Language (SBML). project. *IEE Systems Biology* 1(1), 41–53 (2004)
18. Shannon, P., Markiel, A., Ozier, W., et al.: Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Res.* 13, 2498–2504 (2003)
19. Monga, M., Sausville, E.A.: Developmental Therapeutics Program at the NCI: molecular target and drug discovery process. *Nature* 2, 0887–6924 (2002)
20. The Novartis data U95, http://dtpsearch.ncicrf.gov/FTP/WEB_DATA_NOVARTIS.ZIP
21. The data used in this paper, http://uploadingit.com/files/download/1075417_srm3y/Data.xls
22. The P38 MAPK provided by BioCarta, http://www.biocarta.com/pathfiles/h_P38MAPKPATHWAY.asp
23. Krizman, D.B., Wagner, L., Lash, A., et al.: The Cancer Genome Anatomy Project: EST Sequencing and the Genetics of Cancer Progression. *Neoplasia* 1, 101–106 (1999)
24. Samaranyake, M., Ji, H., Ainscough, J.: Force directed graph drawing algorithms for Macro cell placement. *Lecture Notes in Engineering and Computer Science*, vol. S I-III, pp. 222–227 (2008)
25. Kendal, J.R., Rendell, L., Pike, T.W., et al.: Nine-spined sticklebacks deploy a hill-climbing social learning strategy. *Behavioral Ecology* 20, 238–244 (2009)
26. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *Journal of Machine Learning Research* 3, 1157–1182 (2003)
27. Holt, J.S.: Data analysis with SPSS: A first course in applied statistics. *Teaching Sociology* 36, 285–287 (2008)
28. The code of this paper, http://uploadingit.com/files/1076574_i3moj/Codes.zip

A Novel Multi-threshold Segmentation Approach Based on Artificial Immune System Optimization

Erik Cuevas, Valentín Osuna-Enciso, Daniel Zaldivar, and Marco Pérez-Cisneros

Departamento de Ciencias Computacionales, Av. Revolucion 1500, Guadalajara, Jal. Mexico
{erik.cuevas, daniel.zaldivar, marco.perez}@cupei.udg.mx

Abstract. Threshold selection is a critical step in computer vision. Immune systems, has inspired optimization algorithms known as Artificial Immune Optimization (AIO). AIO have been successfully applied to solve optimization problems. The Clonal Selection algorithm (CSA) is the most applied AIO method. It generates a response after an antigenic pattern is identified by an antibody. This work presents an image multi-threshold approach based on AIS optimization. The approach considers the segmentation task as an optimization process. The 1-D histogram of the image is approximated by adding several Gaussian functions whose parameters are calculated by the CSA. The mix of Gaussian functions approximates the histogram; each Gaussian function represents a pixel class (threshold point). The proposed approach is computationally efficient and does not require prior assumptions about the image. The algorithm demonstrated ability to perform automatic threshold selection.

Keywords: Image segmentation, artificial immune system, automatic thresholding, intelligent image processing.

1 Introduction

Biological inspired methods can successfully be transferred into novel computational paradigms as shown by the development and successful use of concepts such as artificial neural networks, evolutionary algorithms, swarming algorithms and so. The human immune system (HIS) is a highly evolved, parallel and distributed adaptive system [1], which exhibits remarkable abilities that can be imported into important aspects in the field of computation. This emerging field is referring to as artificial immune systems (AIS) [2]. It can be defined as computational systems inspired by theoretical immunology and its observed immune functions, principles and models. AIS have recently gained considerable research interest from different communities [3] considering several aspects of optimization, pattern recognition, anomaly detection, data analysis, and machine learning. Artificial immune optimization has been successfully applied to tackle numerous challenging optimization problems with remarkable performance in comparison to other classical techniques [4].

Clonal selection algorithm (CSA) [5] is one of the most widely employed AIO approaches. The CSA is a relatively novel evolutionary optimization algorithm which has been built on the basis of the clonal selection principle [6] of HIS. The clonal

selection principle explains the immune response when an antigenic pattern is recognized by a given antibody. CSA has the ability of getting out of local minima while operates over a population of points within the search space simultaneously. It does not use the derivatives or any of their related information as it employs probabilistic transition rules instead of deterministic ones. Moreover, its implementation is simple and straightforward. CSA has been therefore used extensively in the literature for solving several kinds of challenging engineering problems [7, 8, 9].

For many applications in image processing, abstractions of objects or features, commonly used by other high-level tasks, are derived from images. In particular for the abstraction analysis, the pixels within the image have to be grouped into meaningful regions by applying image segmentation.

In many cases, the gray level value for pixels belonging to the object is substantially different from those belonging to the background. Thus thresholding becomes a simple but effective tool to separate objects from the background yielding several sorts of applications, such as document image analysis, which aims to extract printed characters [10, 11], logos, graphical contents, or musical scores. Also map processing, which seeks to identify legends and characters [12], scene processing which tracks a given target [13] and quality inspection of materials [14, 15] which aims to detect defective parts.

Thresholding is therefore a simple and effective technique for image segmentation. It can be classified into two categories: bi-level and multi-level thresholding. The latter considers images containing several distinct objects which required multiple threshold values to accomplish segmentation. The bi-level case may segment an image into two simple pixel classes, one representing the object and the other accounting for the background pixels.

A wide variety of thresholding approaches have been proposed for image segmentation, by means of conventional methods such as in [16], [17], [18], [19] and those based on computational intelligent techniques as in [20], [21] and [9]. When extended to multi-level thresholding, two major drawbacks may arise: (i) they have no systematic and analytic solution when the number of classes in the image increases, and (ii) the number of classes in the image is difficult to define and it should be pre-specified, although such parameter is unknown for many real-world applications.

In order to contribute towards solving some of these problems, this paper presents an alternative approach using the optimization algorithm based on AIS for multilevel thresholding. In the traditional multilevel optimal thresholding, the intensity distributions belonging to the object or to the background pixels are assumed to follow some Gaussian probability function; therefore a combination of probability density functions is usually adopted to model these functions. The parameters in the combination function are unknown and the parameter estimation is typically assumed to be a nonlinear optimization problem [22]. The unknown parameters that give the best fit to the processed histogram are determined by using the Clonal selection algorithm.

This paper organizes as follows: Section 2 presents the method following the Gaussian approximation of the histogram. Section 3 provides information about the CSA while Section 4 demonstrates the automatic threshold determination. Section 5 discusses some implementation details. Experimental results for the proposed

approach are presented in Section 6, followed by the discussion summarized in Section 7.

2 Gaussian Approximation

Assuming an image has L gray levels $[0, \dots, L-1]$, following a gray level distribution which can be displayed in the form of the histogram $h(g)$. In order to simplify the description, the histogram is normalized and it is considered as a probability distribution function, yielding:

$$h(g) = \frac{n_g}{N}, \quad h(g) \geq 0, \quad N = \sum_{g=0}^{L-1} n_g, \quad \text{and} \quad \sum_{g=0}^{L-1} h(g) = 1 \tag{1}$$

Assuming that n_g denotes the number of pixels with gray level g , while N is the total number of pixels in the image. The histogram function can thus be contained into a mix of Gaussian probability functions, which yields:

$$p(x) = \sum_{i=1}^K P_i \cdot p_i(x) = \sum_{i=1}^K \frac{P_i}{\sqrt{2\pi\sigma_i}} \exp\left[\frac{-(x-\mu_i)^2}{2\sigma_i^2}\right] \tag{2}$$

Considering that P_i is the a priori probability of class i , $p_i(x)$ is the probability distribution function of gray-level random variable x in class i , μ_i and σ_i are the mean and standard deviation of the i -th probability distribution function and K is the number of classes within the image. In addition, the constraint $\sum_{i=1}^K P_i = 1$ must be satisfied.

The typical mean square error consideration is used to estimate the 3K parameters P_i , μ_i and σ_i , $i = 1, \dots, K$. For example, the mean square error between the composite Gaussian function $p(x_i)$ and the experimental histogram function $h(x_i)$ is defined as follows:

$$E = \frac{1}{n} \sum_{j=1}^n [p(x_j) - h(x_j)]^2 + \omega \cdot \left| \left(\sum_{i=1}^K P_i \right) - 1 \right| \tag{3}$$

Assuming an n -point histogram as in [22] and ω being the penalty associated with the constrain $\sum_{i=1}^K P_i = 1$.

In general, the determination of parameters that minimize the square error is not a simple problem. A straightforward method to decrease the partial derivatives of the error function to zero considers a set of simultaneous transcendental equations [22]. An analytical solution is not available due to the non-linear nature of the equations, and therefore a multilevel optimization method must be used. The algorithm makes use of a numerical procedure following an iterative approach based on the gradient information. However, considering that the gradient descent method may easily get

stuck within local minima, the final solution is highly dependant on the initial values. Some previous experiences have shown that the evolutionary-based approaches actually provide a satisfactory performance in case of image processing problems [9, 20, 23, 24, 25]. The CSA is therefore adopted in order to find the parameters and their corresponding threshold values.

3 Clonal Selection Algorithm

In the natural immune systems, only the antibodies (Abs) which are able to recognize the intruding antigens (non-self cells) are to be selected to proliferate by cloning [6]. Therefore, the fundamental of the clonal optimization method is that only capable Abs will proliferate. More precisely, the underlying principles of the CSA borrowed from the CSP are as follows:

- Maintenance of memory cells functionally disconnected from repertoire,
- Selection and cloning of most stimulated Abs,
- Suppression of non-simulated cells,
- Affinity maturation and reselection of clones with higher affinities, and
- Mutation rate proportional to Abs affinities.

Within Immunology, an antigen is any substance that forces the immune system to produce antibodies against it. In CSA, antigens refer to the pending optimization problem. In CSA, B cells, T cells and antigen-specific lymphocytes are generally called antibodies. An antibody is a representation of a candidate solution for an antigen. A selective mechanism guarantees that those offspring cells (solutions) that better recognize the antigen, which elicited the response, are selected to have long life spans. Therefore these cells are named memory cells (M).

Figure 1 shows the diagram of the basic CSA steps as follows:

1. Initialize the Ab pool (P_{init}) including the subset of memory cells (M).
2. Evaluate the fitness of all the individuals in population P. The fitness here refers to the affinity measure.
3. Select the best candidates (P_r) from P, according to their fitness (affinities to the antigens).
4. Clone these Ab's into a temporary pool (C).
5. Generate a mutated Ab pool (C_1). The mutation rate of each individual is inversely proportional to its fitness.
6. Evaluate all the Ab's in C_1
7. Eliminate those Ab's similar to the ones in C, and update C_1 .
8. From C_1 , re-select the individuals exhibiting the best fitness to build M. Other individuals with improved fitness in C_1 can thus replace other members showing poor fitness in P in order to keep the overall Ab diversity.

The CSA can be terminated once one fixed criteria, previously established, is satisfactorily reached.

The clone size in Step 4 is commonly defined as a monotonic function of the affinity measure. Although a unique mutation operator is used in Step 5, the mutated values of individuals are inversely proportional to their fitness by means of choosing different mutation variations, i.e. as Ab shows better fitness, the less it may change.

The similarity property within the Abs can also affect the convergence speed of the CSA. The idea of the Ab suppression based on the immune network theory is introduced to eliminate the newly generated Abs, which is too similar to those already in the candidate pool (Step 7). With such a diverse Ab pool, the CSA can avoid being trapped into local optima. In contrast to the popular genetic algorithms (GA), which usually tend to bias the whole population of chromosomes towards only the best candidate solution [26], it can effectively handle the challenging multimodal optimization tasks [27- 30].

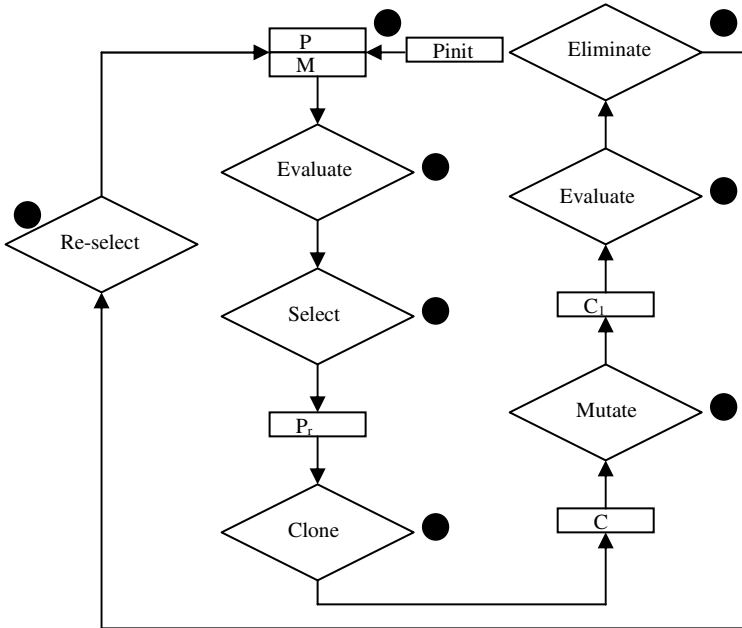


Fig. 1. Basic diagram of clonal selection algorithm (CSA)

The management of population includes a simple and direct searching algorithm for globally optimal multi-modal functions. This is a clear difference in comparison to other evolutionary algorithms, like genetic algorithms, because the method does not require crossover but only cloning and hyper-mutation of individuals in order to use affinity as selection mechanism. The CSA is adopted in this work to find the parameters P, σ and μ of Gaussian functions and their corresponding threshold values for the image.

4 Determination of Thresholding Values

The next step is to determine the optimal threshold values. Considering that the data classes are organized such that $\mu_1 < \mu_2 < \dots < \mu_K$, the threshold values can thus be

calculated by estimating the overall probability error for two adjacent Gaussian functions, as follows:

$$E(T_i) = P_{i+1} \cdot E_1(T_i) + P_i \cdot E_2(T_i), \quad i = 1, 2, \dots, K - 1 \tag{4}$$

Considering

$$E_1(T_i) = \int_{-\infty}^{T_i} p_{i+1}(x) dx, \tag{5}$$

And

$$E_2(T_i) = \int_{T_i}^{\infty} p_i(x) dx, \tag{6}$$

$E_1(T_i)$ is the probability of mistakenly classifying the pixels in the $(i + 1)$ -th class to the i -th class, while $E_2(T_i)$ is the probability of erroneously classifying the pixels in the i -th class to the $(i + 1)$ -th class. P_j 's are the a-priori probabilities within the combined probability density function, and T_i is the threshold value between the i -th and the $(i + 1)$ -th classes. One T_i value is chosen such as the error $E(T_i)$ is minimized. By differentiating $E(T_i)$ with respect to T_i and equating the result to zero, it is possible to use the following equation to define the optimum threshold value T_i :

$$AT_i^2 + BT_i + C = 0 \tag{7}$$

Considering

$$\begin{aligned} A &= \sigma_i^2 - \sigma_{i+1}^2, \quad B = 2 \cdot (\mu_i \sigma_{i+1}^2 - \mu_{i+1} \sigma_i^2), \\ C &= (\sigma_i \mu_{i+1})^2 - (\sigma_{i+1} \mu_i)^2 + 2 \cdot (\sigma_i \sigma_{i+1})^2 \cdot \ln \left(\frac{\sigma_{i+1} P_i}{\sigma_i P_{i+1}} \right) \end{aligned} \tag{8}$$

Although the above quadratic equation has two possible solutions, only one of them is feasible (the positive one which falls within the interval). Figure 2 shows the determination process of the threshold points.

5 Implementation Details

In this work, either an antibody or an antigen will be represented (in binary form) by a bit chain of the form:

$$c = \langle c_1, c_2, \dots, c_L \rangle \tag{9}$$

with c representing a point in an L-dimensional space, $c \in S^L$

The clonal selection algorithm can be stated as follows:

- 1) An original population of N individuals (antibodies) is generated, considering the size of 22 bits.
- 2) The n best individuals based on the affinity measure are selected. They will represent the memory set.

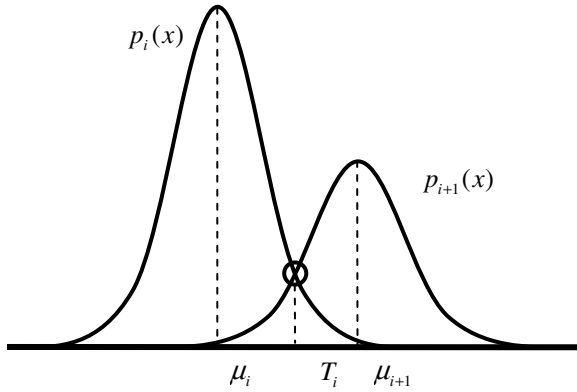


Fig. 2. Determination of the threshold points

- 3) Such n best individuals are cloned m times.
- 4) Performing a hyper-mutation of the cloned individuals which follows the proportion inside the affinity between antibodies and antigens and generating one improved antibody population.
- 5) From the hyper-mutated population, the individuals with the higher affinity are to be re-selected.
- 6) As for the original population, the individuals with the highest affinity are replaced, improving the overall cells set.

Once the above steps are completed, the process is started again, until one individual showing the highest affinity is found, i.e. finding the minimum stated in Equation 3. At this work, the algorithm considers 3 Gaussians to represent the same number of classes. However, it can be easily expanded to more classes. Each single Gaussian has the variables P_i, μ_i, σ_i (with $i=1, 2, 3$) representing the Hamming shape-space by means of an 22 bits word over the following ranges:

$$P_i : [1, \max(h)], \mu_i : [\min(g), \max(g)], \sigma_i : [1, \max(g) * 0.5] \tag{10}$$

With g representing the grey level, and h is the histogram of the grey level image. Hence, the first step is to generate the initial individual population of the antibody population by means of $AB = 2 * \text{rand}(N, S_p) - 1$; and S_p representing the bit string assigned to each of the initial individuals N . Later, in order to perform the mapping from binary string to base 10

$$\langle (c_L, \dots, c_2, c_1) \rangle_2 = \left(\sum_{i=0}^{21} c_i \cdot 2^i \right)_{10} = r' \tag{11}$$

As to find the corresponding real value for r :

$$r = r' \cdot \frac{r_{\max}}{2^{22} - 1} \tag{12}$$

with r_{\max} representing $\max(h), \max(g), \max(g) / 2$.

The population is set to 100 individuals (N), including the best 20 individuals (n). The 20 selected individuals are cloned 10 times (m). The corresponding mutation probability is proportional to the resulting error by Equation 3. The algorithm is thus executed until the minimum possible value of Eq. 3 is reached.

7 Conclusions

This work presents a novel segmentation algorithm which includes an automatic threshold determination approach. The intensity distributions for objects and background within the image are assumed to obey Gaussian distributions with different means and standard deviations.

The histogram of the image is approximated by a combination of Gaussian probability functions. The 1-D histogram of the image is approximated by the addition of several Gaussian functions whose parameters are calculated using the CSA. Each Gaussian function approximates the histogram, representing a pixel class and therefore a threshold point. The clonal selection algorithm is used to estimate the parameters within the mixed density function obtaining a minimum square error between the density function and the original histogram. Experimental results reveal that the proposed approach can produce satisfactory results. The paper also discusses on two benchmark images to test the performance of the algorithm. The proposed approach is not only computationally efficient but also it does not require prior assumptions about the image whatsoever. The method is likely to be most useful for applications considering different and perhaps initially unknown image classes. Experimental results demonstrate the algorithm's ability to perform automatic threshold selection while preserving main features from the original image.

Further work extending the proposed approach to other techniques and its comparison to other newly-developed state-of-the-art image segmentation techniques is currently under development.

References

1. Goldsby, G.A., Kindt, T.J., Kuby, J., Osborne, B.A.: Immunology, 5th edn. Freeman, New York (2003)
2. de Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer, London (2002)
3. Dasgupta, D.: Advances in artificial immune systems. IEEE Computational Intelligence Magazine 1(4), 40–49 (2006)
4. Wang, X., Gao, X.Z., Ovaska, S.J.: Artificial immune optimization methods and applications—a survey. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, The Hague, The Netherlands (October 2004)
5. de Castro, L.N., von Zuben, F.J.: Learning and optimization using the clonal selection principle. IEEE Transactions on Evolutionary Computation 6(3) (2002)
6. Ada, G.L., Nossal, G.: The clonal selection theory. Sci. Am. 257, 50–57 (1987)
7. Coello Coello, C.A., Cortes, N.C.: Solving multiobjective optimization problems using an artificial immune system. Genet. Program. Evolvable Mach. 6, 163–190 (2005)

8. Campelo, F., Guimaraes, F.G., Igarashi, H., Ramirez, J.A.: A clonal selection algorithm for optimization in electromagnetics. *IEEE Trans. Magn.* 41, 1736–1739 (2005)
9. Weisheng, D., Guangming, S., Li, Z.: Immune memory clonal selection algorithms for designing stack filters. *Neurocomputing* 70, 777–784 (2007)
10. Abak, T., Baris, U., Sankur, B.: The performance of thresholding algorithms for optical character recognition. In: *Proceedings of International Conference on Document Analytical Recognition*, 697–700 (1997)
11. Kamel, M., Zhao, A.: Extraction of binary character/graphics images from grayscale document images. *Graph. Models Image Process.* 55(3), 203–217 (1993)
12. Trier, O.D., Jain, A.K.: Goal-directed evaluation of binarization methods. *IEEE Trans. Pattern Anal. Mach. Intell.* 17(12), 1191–1201 (1995)
13. Bhanu, B.: Automatic target recognition: state of the art survey. *IEEE Trans. Aerosp. Electron. Syst.* 22, 364–379 (1986)
14. Sezgin, M., Sankur, B.: Comparison of thresholding methods for non-destructive testing applications. In: *IEEE International Conference on Image Processing* (2001)
15. Sezgin, M., Tasaltin, R.: A new dichotomization technique to multilevel thresholding devoted to inspection applications. *Pattern Recognition Lett.* 21(2), 151–161 (2000)
16. Guo, R., Pandit, S.M.: Automatic threshold selection based on histogram modes and discriminant criterion. *Mach. Vis. Appl.* 10, 331–338 (1998)
17. Pal, N.R., Pal, S.K.: A review on image segmentation techniques. *Pattern Recognit.* 26, 1277–1294 (1993)
18. Shaoo, P.K., Soltani, S., Wong, A.K.C., Chen, Y.C.: Survey: A survey of thresholding techniques. *Comput. Vis. Graph. Image Process.* 41 (1988)
19. Snyder, W., Bilbro, G., Logenthiran, A., Rajala, S.: Optimal thresholding: A new approach. *Pattern Recognit. Lett.* 11, 803–810 (1990)
20. Chen, S., Wang, M.: Seeking multi-thresholds directly from support vectors for image segmentation. *Neurocomputing* 67(4), 335–344 (2005)
21. Chih-Chih, L.: A Novel Image Segmentation Approach Based on Particle Swarm Optimization. *IEICE Trans. Fundamentals* 89(1), 324–327 (2006)
22. Gonzalez, R.C., Woods, R.E.: *Digital Image Processing*. Addison Wesley, Reading (1992)
23. Baştürk, A., Günay, E.: Efficient edge detection in digital images using a cellular neural network optimized by differential evolution algorithm. *Expert System with Applications* 36(8), 2645–2650 (2009)
24. Lai, C.-C., Tseng, D.-C.: An optimal L-filter for reducing blocking artifacts using genetic algorithms. *Signal Process.* 81(7), 1525–1535 (2001)
25. Tseng, D.-C., Lai, C.-C.: A genetic algorithm for MRF-based segmentation of multispectral textured images. *Pattern Recognit. Lett.* 20(14), 1499–1510 (1999)
26. Poli, R., Langdon, W.B.: *Foundations of Genetic Programming*. Springer, Berlin (2002)
27. Yoo, J., Hajela, P.: Immune network simulations in multicriterion design. *Structural Optimization* 18(2-3), 85–94 (1999)
28. Wang, X., Gao, X.Z., Ovaska, S.J.: A hybrid optimization algorithm in power filter design. In: *Proceedings of the 31st Annual Conference of the IEEE Industrial Electronics Society*, Raleigh, NC, November 2005, pp. 1335–1340 (2005)
29. Xu, X., Zhang, J.: An improved immune evolutionary algorithm for multimodal function optimization. In: *Proceedings of the Third International Conference on Natural Computation*, Haikou, China, August 2007, pp. 641–646 (2007)
30. Tang, T., Qiu, J.: An improved multimodal artificial immune algorithm and its convergence analysis. In: *Proceedings of the Sixth World Congress on Intelligent Control and Automation*, Dalian, China, June 2006, pp. 3335–3339 (2006)

Research on Improvement Strategies and Parameter Analysis of Ant Colony Algorithm for One-Dimensional Cutting Stock Problem

Lan Huang, Huayi Chen, Bin Yang, and Chun-Guang Zhou

College of Computer Science and Technology,
Jilin University, Changchun 130012, China
{huanglan, yangbin}@jlu.edu.cn

Abstract. Ant Colony Algorithm is a new bionic algorithm which adopts the positive feedback structure, combines parallel computing and heuristic factors. It showed remarkable performance in combinatorial optimization problem. One-dimensional Cutting Stock is one of the classic NP-hard combinatorial optimization problems. It is widely applied in engineering technology and industrial production. Aiming at the specific characteristics of the problem, a series of improvement strategies and the specific algorithm implementation steps are given. Then the parameters are analyzed in details. Through experiment analysis and results comparison, it is proved that the improvement strategies and adjusted parameters have advantages in implementation efficiency and solving ability.

Keywords: One-dimensional Cutting Stock Problem, Ant Colony Algorithm, Improvement strategies, Parameter analysis.

1 Introduction

Ant Colony Algorithm (ACA) was originally proposed by the Italian scholar Dorigo etc [1][2]. They take full use of the similarity between the process of ant colony searching food and Traveling Salesman Problem (TSP), through simulating ants that exchange information between the individuals and mutual collaboration during searching the shortest path from the nest to the ultimate food source, solve the famous TSP and achieve good results. Its applications have been infiltrated into other areas successfully. The most successful application is in the combinatorial optimization problems, such as Job Scheduling Problem (JSP), Vehicle Routing Problem (VRP), Quadratic Assignment Problem (QAP), etc [3].

One-dimensional Cutting Stock Problem (1CSP) is one of the classic NP-hard combinatorial optimization problems. It has important and extensive applications in engineering technology and industrial production [4][5][6]. According to the basic idea of ACA, the optimized 1CSP model and a series of improvement strategies are proposed; the specific implementation steps of improved ACA for 1CSP are presented. Then the parameters are analyzed in details. Through comparison of

experimental results with [7], this algorithm can find optimal solution more rapidly and efficiently.

2 Improvement Strategies of Ant Colony Algorithm

In ACA a set of artificial agents called ants cooperate to the solution of a problem by exchanging information via pheromone deposited on graph edges. Ants make use of the amount of the pheromone deposited by other ants as their parameters of choosing path. The more pheromone is on the edge, the higher probability is the edge to be chosen with. The update strategy of pheromone is that the shorter edge should get the more pheromone. Searching the better route through this positive feedback is the idea of ACA. Behavior rules of an individual are very easy, but through mutual collaboration the ant colony can perceive the complicated environment, and search the solution space effectively.

Research and application of basic ACA has made great progress. A large number of results proved its advantages, such as: strong robustness, distributed computing ability, and easy to combine with other methods, etc. However, it also has some drawbacks. Firstly, the larger is the problem scale, the longer time it will take to find a better path because the movement of individuals in the colony is random; Secondly, it's easy to fall into local optimal solution as a result of strengthening the role of pheromones.

In response to these two drawbacks we improved the basic ACA. The improved ACA has obvious advantages in solving ability, and its implementation efficiency increase significantly.

2.1 Efficiency Improvement Strategies

Adjustment of the Ants' Number. The number of ants has great relationship with the time complexity of algorithm. If maintaining the same number of ants in each iteration, when the search converges to the vicinity of near-optimal solution, the more is the ants, the more time will be wasted. So the number of ants needs to be adjusted according to the solution searched. The adjustment strategy for the number of ants: In the beginning of searching, set the number of ants to be the maximum number $AntNum_{max}$, to remain the diversity of searching solutions, and avoid falling into local optimal solution. If the optimal solution has no or little change in the last continuous 5 (or 10) iterations, that is, the relationship between searching optimal solution and the number of ants is not very important, then reduce the number of ants accordingly.

Establishment of Candidate Nodes Set. Calculate a candidate nodes set for each node. Ants should choose the next node in the candidate set in priority. Only when all the nodes in the candidate set are in the tabu list of the ant, the ant can choose other node. The size of candidate set is usually set between 10 and 30 so as to reduce the time complexity of constructing ant path greatly. For each node, its surrounding nodes are ordered by distance, and then a fixed number of surrounding nodes are chosen into its candidate set. Because the probabilities of all nodes need not to be calculated, the time complexity can be reduced.

Rank-Based Pheromone Updating Strategy. In the beginning of early iterations, in order to maintain the diversity of the searching solutions, update pheromone of all the ants' paths. With the increasing of iteration times, the solution has been carried on crossover and mutation operation of Genetic Algorithm. Introducing the updating path pheromone strategy of optimized rank-based ant system [8], we use a hybrid strategy of elite and sorting [9] to update the path pheromone. Based on the order of solution quality, we take out the first σ ants to update the pheromone on their path. The path pheromone is controlled in $[\tau_{\min}, \tau_{\max}]$.

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij} + \Delta\tau_{ij}^* \tag{1}$$

where $\Delta\tau_{ij} = \sum_{\mu=1}^{\sigma-1} \Delta\tau_{ij}^\mu$, it is updating pheromone which the former $\sigma - 1$ ants produced on the edge(i,j).

$$\Delta\tau_{ij}^\mu = \begin{cases} (\sigma - \mu) \frac{Q}{L^\mu} & \text{if the } \mu\text{th best ant visits the edge}(i, j) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$\Delta\tau_{ij}^* = \begin{cases} \sigma \frac{Q}{L^*} & \text{if the edge}(i, j) \text{ is a part of the best solution} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where μ is the order number of the best ants; $\Delta\tau_{ij}^\mu$ is the increment pheromone produced on the edge (i,j) by the μ th best ant. L_μ is the path length of the μ th best ant. $\Delta\tau_{ij}^*$ is the increment pheromone produced on the edge (i,j) by the elite ants. σ is the number of elite ants, and L^* is the path length of the found optimal solution.

2.2 Solving Ability Improvement Strategies

Transition Probability Strategy. Combining the above-mentioned candidate nodes set strategy, introduce stochastic transition probability.

The process of ants constructing solution is equivalent to an agent, whose task is to visit all the nodes and returned to the starting node to make a loop. Set the ant k on the node i , whose candidate set is $cset_i$. The ant k will move from the node i to the node j according to the state transition rules, as shown as (4) and (5).

If not all nodes in the candidate set of the node i haven't been added into the tabu list $tabu_k$,

$$s = \begin{cases} \arg \max_{j \in cset_i} \{ [\tau(i, j)]^\alpha \cdot [\eta(i, j)]^\beta \} & (q \leq q_0) \\ P_{ij}^k(t) & \text{otherwise} \end{cases} \tag{4}$$

If all the nodes in the candidate set of the node i have been added into $tabu_k$,

$$s = \begin{cases} \arg \max_{j \in allowed_k} \{ [\tau(i, j)]^\alpha \cdot [\eta(i, j)]^\beta \} & (q \leq q_0) \\ P_{ij}^k(t) & \text{otherwise} \end{cases} \quad (5)$$

where q is a random value well-distributed in $[0,1]$, q_0 is a parameter ($0 \leq q_0 \leq 1$).

$$P_{ij}^k(t) = \begin{cases} \frac{(\tau_{ij}^k(t))^\alpha (\eta_{ij}^k(t))^\beta}{\sum_{j \in allowed_k} (\tau_{ij}^k(t))^\alpha (\eta_{ij}^k(t))^\beta} & s \in allowed_k \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Ant Individual Differences. In the basic ACA, each ant choose next node according to the probability function s , shown as (4) to (6), in which the values of parameter α and β of each ant are fixed, so the behavior strategy of each ant in the colony is the same, while actually ants behavior is the diversity. This algorithm simulates the individual difference in the same colony, in which ants with different choosing strategies interact, so it can get better solution than the same strategy.

Adaptive Adjustment of Parameters α and β . Pheromone heuristic factor α reflects the relative importance of pheromone accumulated in the moving process, while expectation heuristic factor β reflects the relative importance of heuristic information in the guidance process of searching. In the initial stage, small value for α and β can enlarge the search space. In later stage increasing value for α and β can reduce the search space to make the solution close to the optimal path and produce positive feedback. This strategy will not only speed up the algorithm convergence rate, but also when the algorithm falls into a local optimal path, since the role of positive feedback has been strengthened, the algorithm will be easier to find better solutions and jump out of local optimal values in time.

Adjust the value of α and β according to (7) and (8).

$$\alpha(t+1) = \begin{cases} \xi_1 \alpha(t) & \text{if } \xi_1 \alpha(t) < \alpha_{\max} \\ \alpha_{\max} & \text{otherwise} \end{cases} \quad (7)$$

$$\beta(t+1) = \begin{cases} \xi_2 \beta(t) & \text{if } \xi_2 \beta(t) < \beta_{\max} \\ \beta_{\max} & \text{otherwise} \end{cases} \quad (8)$$

where ξ_1 and ξ_2 are constants more than 1.

Combination with Genetic Algorithm. The hybrid ACA combined with GA takes advantages of mutation and crossover of GA. Introducing the crossover operator and mutation operator can enhance the solving ability of searching global optimum and accelerate the convergence speed of ACA.

3 Description of One-Dimensional Cutting Stock Problem

Cutting Stock Problem (CSP) is the problem to divide some stocks with same shape into some smaller parts with different size. If the parts have the same width, the problem is called One-dimensional Cutting Stock Problem (1CSP). These problems often appear in computer science, industrial engineering and machinery manufacturing, etc. Accurate solution of 1CSP is NP problem, coupled with the constraints of the number of cutting ways, and expansion of problem scale, the solution of such problems become more complex. It is difficult to use an accurate method i.e. linear programming, integer programming, branch and bound, etc. for this problem. At present the heuristic algorithm i.e. tabu search method, genetic algorithm, etc. are mainly adopted to solve it. The research on applying ant colony algorithm for 1CPS is still rare [7].

We adopt a multi-objective multi-constrained optimization model of 1CSP based on a minimal total waste and minimal cutting ways.

Given the length of stock is L , and the length of n parts are l_1, l_2, \dots, l_n respectively, the quantity required are b_1, b_2, \dots, b_n respectively. There are m possible cutting ways, and the repetition times of each way is $x_j (j=1, 2, \dots, m)$. In the way j , the repetition times of the i th parts is $a_{ij} (i=1, 2, \dots, n)$. The waste of each cutting seam is LS . Therefore the mathematical model based on the minimum total waste and minimum number of cutting ways is shown as follow.

$$\begin{cases} \min F = L \sum_{j=1}^m x_j - \sum_{j=1}^m \sum_{i=1}^n l_i a_{ij} x_j \\ \min m \end{cases} \quad (9)$$

$$\begin{cases} L - \sum_{i=1}^n (l_i + LS) a_{ij} \geq 0 \\ s.t. \sum_{j=1}^m a_{ij} x_j = b_i \\ x_j \geq 0 \end{cases} \quad (10)$$

(9) are the objective functions of the model, and (10) are the constraints, where $j=1, 2, \dots, m, i=1, 2, \dots, n$. Here, to facilitate solving, we adopted the linear weighted method [10] to transform the multi-objective problem to a single objective problem. The optimized model is shown as follow.

$$\min (F + qm) \quad (11)$$

where q is the weighted coefficient.

For the convenience of comparison, example data adopts the same data as [7]. As shown in Table 1, 53 kinds of parts are planed to be cut. The length of stock $L = 30$ mm, and 5 mm loss will be produced at every cutting seam.

Table 1. Various length of parts

Part No	1	2	3	4	5	6	7	8	9	10	11
Length	1743	1680	1532	1477	1313	1285	1232	1217	1180	1177	1105
Part No	12	13	14	15	16	17	18	19	20	21	22
Length	1055	1046	1033	1030	975	893	882	847	845	830	795
Part No	23	24	25	26	27	28	29	30	31	32	33
Length	766	645	732	719	714	690	665	633	630	600	590
Part No	34	35	36	37	38	39	40	41	42	43	44
Length	588	582	578	540	488	455	434	420	415	414	411
Part No	45	46	47	48	49	50	51	52	53		
Length	405	328	313	290	275	265	255	184	155		

4 Improved Ant Colony Algorithm for One-Dimension Cutting Stock Problem (IACA-1CSP)

1CSP is a classic combinatorial optimization problem, and ACA shows excellent performance in solving these problems, especially in solving single objective problem. Optimized model of 1CSP, shown as (6), is suitable for using ACA. [7] gives an ACA for 1CSP. The IACA-1CSP proposed in this paper has obvious advantages in solving ability and implementation efficiency.

4.1 Parts Encoding

Encode the B parts with the number from 1 to B ($B=b_1+b_2+\dots+b_n$). The same parts with different code are regarded as different parts. The collection of encoded parts is named W ($W=\{1,2,\dots,B\}$). Regard the B parts as the B nodes of ACA. At the beginning, randomly initialize m ants, and place these ants randomly on the B nodes.

4.2 Solution Path

From the first part node which an ant choose, when the total length of parts close to L but not larger than L , we look it as one cutting way. From the next part we recalculate the total length of parts in the same method, until all the parts on the path are calculated. Then we get the total waste length and the number of cutting ways of the path. All the parts codes in all cutting ways are connected one by one to get an ant path.

4.3 Specific Implementation Steps

Step 1: Initialize parameters. Set the cycle times $NC \leftarrow 0$, and set the maximum number of cycles NC_{max} .

Step 2: Encode parts. The collection of encoded parts is set W ($W=\{1,2,\dots,B\}$).

Step 3: Calculate the candidate nodes set. According to the heuristic information $f_{ij}(f_{ij}=L-l_i-l_j)$, calculate the corresponding candidate nodes set of each node i , named $cset_i$.

Step 4: Set the initial pheromone on each edge(i, j), namely $\tau_{ij}(0) = const$, where $const$ is a constant and initially $\Delta\tau_{ij}(0) = 0$. Set the values of α and β .

Step 5: Set cycle times $NC \leftarrow NC + 1$.

Step 6: Initialize m ants, and placed the m ants to the B nodes. Add the current node i of the ant k to the tabu list $tabu_k$.

Step 7: Set ant index $k = 1$.

Step 8: Set ant index $k \leftarrow k + 1$.

Step 9: Adjust the ants' individual differences. Properly adjust each ant's α and β . Floating range of α and β is not more than 10%.

Step 10: Choose the next node according to state transition probability rules (4) to (6). In (6), $P_{ij}^k(t)$ is the state transition probability of the ant k from the node i to the j , $\eta_{ij}^k(t) = L - l_i - l_j$, which is the total waste if the ant k chooses the node j after i .

Step 11: According to the calculated probability, choose the node j and move the ant k to it, where $j \in \{W - tabu_k\}$. Then modify the tabu list, namely add the node j to $tabu_k$.

Step 12: If not all the nodes in W are visited, i.e. $k < B$, go to step 8, else to step 13.

Step 13: If $NC > 10$, introduce the crossover and mutation of GA. If the new path solution calculated by GA is better than the original one, update the path solution. Go to step 14. If $NC \leq 10$, update the pheromone on paths of all ants. Go to step 15.

Step 14: Rank the solutions by quality, and update the pheromone on paths of the first σ ants according to (1) to (3).

Step 15: Compare the optimal solutions of each cycle. If the optimal solution has no or little change in the last continuous 5 (or 10) cycles, reduce the number of ants.

Step 16: If meet the end condition, namely if the number of cycle $NC \geq NC_{max}$, end the experiment and output the results, otherwise clear the tabu lists and go to step 5.

5 Parameter Analysis

In the experiment, the different choices of ACA parameters of is essential, but by so far there are no mature selection method and guiding principle, especially in ACA for different application problems. Owing to the lack of theoretical basis, we can only through the simulation experiments to analyze the various parameters of the algorithm, and then discuss the optimal setting principles of ACA parameters.

In ACA, the parameters which affect the quality of solution mainly include several factors in the probabilistic function P_{ij}^k , i.e. α, β, η_{ij} and $\tau_{ij}(t)$. Among them, η_{ij} is the waste length after cutting parts i and j from the stock. Its value is fixed at the beginning of the algorithm. According to (1), $\tau_{ij}(t)$ is determined by the parameters $\rho, \Delta\tau_{ij}$ and $\Delta\tau_{ij}^*$, and according to (2) and (3), $\Delta\tau_{ij}$ and $\Delta\tau_{ij}^*$ are determined mainly by the value of Q .

Based on above analysis, parameter selection mainly include the relative importance of pheromone α , the relative importance of heuristic information β ,

volatile factor ρ , and the number of ants m . For each different magnitude values of the parameters, three experimental simulations are conducted.

5.1 Number of Ants m

Set $Q = 5000$, $\alpha = 1$, $\beta = 1$, $\rho = 0.5$, and mark the iteration times of the optimal solution as n . The results shown as the Table 2, when the number of ants $m = 53$, the frequency of optimal solution is highest. If the number of ants is enhanced, though the stability and the global search performance are improved, the running time of the algorithm is also increased. If the number of ants outweighs the problem scale, enhancing the number of ants has no obvious effect on the algorithm performance. Therefore, from the experimental results the number of ants m should be set to be the problem scale.

Table 2. Influence of the number of ants m

Iteration times	$m=1$		$m=10$		$m=100$		$m=\text{problem scale}(53)$	
	optimum	n	optimum	n	optimum	n	optimum	n
1	4285	1	4285	2	4285	1	1285	486
2	7285	1	4285	1	1285	176	4285	1
3	7288	1	4285	2	1287	682	1285	135

5.2 Pheromone Intensity Q

In (2) and (3), Q is the total pheromone released on the path, and $\Delta\tau_{ij}^{\mu}$ is updating pheromone released on the path after a cycle. $\Delta\tau_{ij}^{\mu}$ is determined by the ratio of constant Q to the path length. The larger is Q , the faster does the pheromone accumulate on the path. It can strengthen the positive feedback performance to speed up convergence.

Set $m = 53$, $\alpha = 1$, $\beta = 1$, $\rho = 0.5$. The results shown as the Table 3, when the total pheromone $Q = 6000$, the frequency of optimal solution is highest. Therefore, the total pheromone intensity Q should be set $Q = 6000$.

Table 3. Influence of pheromone intensity Q

Iteration times	$Q=60$		$Q=600$		$Q=6000$		$Q=60000$	
	optimum	n	optimum	n	optimum	n	optimum	n
1	4285	1	4285	1	1285	120	1285	170
2	4285	3	4285	1	4285	1	4285	1
3	4285	1	4285	1	1285	167	4285	1

5.3 Pheromone Heuristic Factor α

The value of α reflects the relative importance of pheromone $\tau_{ij}(t)$. The bigger is α , the more likely the ants choose the path previously travelled, and the less likely to

search randomly, Therefore, if α is too large, the algorithm will be premature in a local optimal search, while if α is too small, convergence will be slow, and the algorithm fall into local optimal solution easily.

Set $m = 53, Q = 6000, \beta = 6, \rho = 0.5$. The results shown as the Table 4, when $\alpha = 1$, the frequency of optimal solution is highest. Therefore, α should be set $\alpha \approx 1$.

Table 4. Influence of α

Iteration times	$\alpha = 0.01$		$\alpha = 0.1$		$\alpha = 1$		$\alpha = 10$	
	optimum	<i>n</i>	optimum	<i>n</i>	optimum	<i>n</i>	optimum	<i>n</i>
1	4285	5	4285	1	1285	356	4285	1
2	7285	1	4285	1	4285	1	4285	2
3	1285	1419	4285	1	1285	212	4285	2

5.4 Expectation Heuristic Factor β

The value of β reflects the relative importance of expectation heuristic information. The bigger is β , the more likely the ants choose the node close to it. Although the convergence rate is accelerated, the randomness of search is weakened, it easily step into the local optimum. If β is too small, the algorithm is led to approximate stochastic search, can hardly find the optimal solution.

Set $m = 53, Q = 6000, \alpha = 1, \rho = 0.5$. The results shown as the Table 5, when $\beta = 1$, the frequency of optimal solution is highest. Therefore, β should be set $\beta \approx 1$.

Table 5. Influence of expectation heuristic factor β

Iteration times	$\beta = 0.01$		$\beta = 0.1$		$\beta = 1$		$\beta = 10$	
	optimum	<i>n</i>	optimum	<i>n</i>	optimum	<i>n</i>	optimum	<i>n</i>
1	4285	1	4285	1	1285	411	4285	1
2	4285	3	4285	2	4285	1	4285	1
3	4285	1	1285	151	1285	116	4285	3

5.5 Volatile Factor ρ

ρ reflects the residue of pheromone, and $1 - \rho$ reflects the volatile degree and speed of pheromone. When $1 - \rho$ is too big, due to volatilization of pheromone, the pheromone on the edges that have never been visited will be reduced (in particular, when the problem scale is big enough, the pheromone on the edges that have never been visited will be close to 0), so as to lower the global searching ability. However, if reduce the speed of pheromone volatile $1 - \rho$, though it can improve the random performance and the global searching ability of the algorithm, it also makes the convergence slow.

Set $m = 53$, $Q = 60$, $\alpha = 1$, $\beta = 1$. The results shown as the Table 6, when $\rho = 0.1$, performance of the algorithm is stable and uniform, searching global ability and convergence speed are quite good. Therefore, the pheromone volatility should be set $1 - \rho = 0.9$, namely $\rho = 0.1$.

Table 6. Influence of volatile factor ρ

Iteration times	$\rho = 0.1$		$\rho = 0.3$		$\rho = 0.5$		$\rho = 0.7$		$\rho = 0.9$	
	optimum	n	optimum	n	optimum	n	optimum	n	optimum	n
1	1285	235	4285	2	1285	788	4285	1	1285	586
2	1285	115	1285	136	1285	1626	1285	373	1285	647
3	1285	29	1285	482	1285	923	1285	292	1285	823

We study the optimal parameters combination according to [11], which summered an effective method named "three-step" based on the experimental analysis of the ACA parameters selection rules. Specific steps are:

Step 1: Determine the number of ants.

Step 2: Adjust the pheromone heuristic factor α and the expectation heuristic factor β .

Step 3: Adjust the pheromone volatile factor ρ .

The relationship of ρ , n , m and Q is that: n , m and ρ should be set relative small, so the pheromone of the edges which have been visited will not be volatilizes to 0. The larger is Q , the larger should ρ be set. Because when Q increases, if ρ remains unchanged, the pheromone of the edges which have been visited will increase, the randomness of the algorithm is reduced.

Through a lot of simulation experiments, an optimal parameters combination of the IACA-1CSP is determined: when $m = n$, $Q=6000$, $\alpha = 1$, $\beta = 1$, $\rho = 0.01$, the performance of the algorithm will be best in all experiments.

6 Experimental Results

The experimental results of [7] is that the total waste is 1288 mm and the cutting ways is 14.

Our improved algorithm is programmed with Matlab V7.1. Through a series of comparative simulation experiments and research, we finally get an optimal parameter combination of IACA-1CSP, i.e. $m = n = 53$, $Q = 60$, $\alpha = 1$, $\beta = 1$, $\rho = 0.01$. Taking the average of 10 experiments, when $NC=51$, we get the best solution for 1285 mm wastes and 14 cutting ways.

7 Conclusion

To improve solving ability and implementation efficiency of ACA, we put forward a series of improvement strategies and analyze parameters in details. Through the

experimental results comparison with [7], the IACA-1CSP proposed by us is proved more efficient, more stable and stronger.

Acknowledgments. This work was supported by the National High Technology Development Project of China (2007AA04Z114).

References

1. Colorni, A., Dorigo, M., Maniezzo, V.: Distributed optimization by ant colonies. In: Proceedings of 1st European Conference on Artificial Life, Paris, France, pp. 134–142. Elsevier publishing, Amsterdam (1991)
2. Dorigo, M., Gambardella, L.M.: Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation* 1, 53–66 (1997)
3. Huang, L., Zhou, C.G., Wang, K.P.: Hybrid Ant Colony Algorithm for Traveling Salesman Problem. *Progress in Nature Science* 13, 295–299 (2003)
4. Jia, Z.X., Yin, G.F., Hu, X.B., Shu, B.: Optimization for One-Dimensional Cutting-Stock Problem Based on Genetic Algorithm. *Journal of Xi an Jiaotong University* 36, 967–970 (2002)
5. Umetani, S., Yagiura, M., et al.: One-dimensional Cutting-stock Problem with a Given Number of Setups: A hybrid Approach of Metaheuristics and Linear Programming. *Journal of Mathematical Modelling and Algorithms* 5, 43–64 (2006)
6. Wang, L.J.: Research and Development on One Dimensional Optimum Cutting System Base Linear Programming. Master Thesis of Dalian university of technology (2003)
7. Wu, Z.J., Zhang, L.P., Wang, K.: An Ant Colony Algorithm for One-dimensional Cutting-stock Problem. *Mechanical Science and Technology for Aerospace Engineering* 27, 1681–1684 (2008)
8. Bullnheimer, B., Hartl, R.F., Strauss, C.: A New Rank-based Version of the Ant System: A Computational Study. *Central European Journal for Operations Research and Economics* 7, 25–38 (1999)
9. Maniezzo, V., Dorigo, M., Colorni, A.: The Ant System Applied to the Quadratic Assignment Problem. Technical Report IRIDIA/94~28, Universite de Bruxelles, Belgium (1994)
10. Operations research textbook compiling group: Operations research. Tsinghua University Press, Beijing (2004)
11. Duan, H.B.: Theory and Application of Ant Colony Algorithm. Science Press, Beijing (2005)

Mixture of Experts with Genetic Algorithms

Laura Cleofas, Rosa Maria Valdovinos, and C. Juárez

Computo Aplicado Group, Centro Universitario UAEM Valle de Chalco
Universidad Autonoma del Estado de México
Hermenegildo Galeana 3, Col. Ma. Isabel, 56615, Valle de Chalco (México)
laura18cs@hotmail.com, li_rmvr@hotmail.com,
cjlandin@yahoo.com.mx

Abstract. Mixture of Experts constructing (MxE) is visualized from two slopes: considering diversity in the original training set or diversity in each classifier. Traditionally, the label of the test patterns has been determined by means of an individual classifier, nevertheless another non-traditional methodology of classification will be presented in this work (Mixture of Experts based in Evolutionary Algorithms), with this methodology is possible to guarantee diversity in each member of the MxE. The rules of apprenticeship considered for the MxE are: the Nearest Neighbor Rule and a Modular Neuronal Network. The experiments were obtained using real data bases form the UCI repository.

1 Introduction

Several researches had determined that is not possible to find the optimal solution with an individual classifier [1]. Dietterich [1], [2], considered a new form to use the classification algorithms, these are known as: Mixture of Experts, Multiple Systems of Classification, Hybrid Methods, Combination of Decisions, Multiple Experts, Opinions Group, methods of ensemble, among others [3], [4]. The MxE is a set of classifiers, where the decision of each member is used to obtain the final prediction.

A way to construct the MxE is by means of the representative resample (speaking of the number of classes within from each data set) from an original training set, each data set appears as experts or classifiers in the MxE.

When constructing the MxE with traditional methods (Bagging, Bosting, among others), sometimes difficult to obtain subsamples with high accuracy and an acceptable diversity. That situation affects in the performance of the classifier. For that, is necessary to count with algorithms or methods that consider both aspects (diversity, precision) of balanced form [1]. There are methods (Bagging and Boosting) that obtain training subsamples that in some form guarantee a certain degree of diversity in the MxE, but it does not necessarily promise to increase the accuracy in the classification step. The Genetic Algorithm (GA) guarantees diversity and precision, due to its capacity to look for optimal solutions from the sample of original training [5], [6].

The GA was proposed in 1960 by John Holland [7], and it is defined as a heuristic search algorithm and a numerical optimization, inspired by processes normally associated with the evolution of live beings. Thus, the individual that adapts faster and better, will prevail in the following generation [2], [6], [7], [8].

In this work a methodology of MxE based on a GA that it treats with the unbalance the classes in the original training set (AGM) is presented. With the AGM, the best individuals of each population were obtained (the best solutions or subsamples), these are used to construct the MxE. In section 2 the GA that treats the unbalance is presented, in section 3 the MxE and the subsequent sections the results (section 4) and the conclusions (section 5).

2 Genetic Algorithms

The evolutionary algorithms try to model the evolution of a population, their learning is based on groups of individuals or populations, and each population is a point in the search space of the optimal solutions for a certain problem [2]. The populations of individuals or chromosomes are made better by means of competition genetic procedures (to select the best individuals against the worse ones) and of controlled variation (through mutation and crossover). Each individual of the population has associate an adaptation measurement, this is used to choose a individual that will be selected in order to form part of the new generation in the competition process [2], [9].

The current investigation begins with a Genetic Algorithm (AGSB) proposed by Diaz et al [10], they consider a binary representation in the GA, an elitist selection (to evaluate the individuals fitness they used the "leave one out" method), where the better chromosomes are chosen for the following generations (they were determined up to 30 epochs, with 15 individuals each one) in such a way that the worse individuals did not survive. For the obtaining the following generations, the operators of crossover (uniform and in a point) and mutation by insertion were considered. Diaz, et al. additionally determined the reduction of training patterns (of 20%) in each chromosome of each population of the GA. The reduction was realized in order to diminish the processing time in the GA, in addition they determined that the reduction of patrons does not affect the performance of the MxE based on a GA.

This same scheme of the GA was modified by the current investigation, the change consisted of dealing with the unbalance of classes in the first population of the GA (AGM), this can be realized in the following way: to generate the first population of the GA, to determine a threshold (total of patterns of the sample of training between the number of the classes), if the number of patterns of each class by each individual of the first population is greater than the threshold, a new population is generated until a balance of the classes is realized in each individual of the first population, in this way it is tried the unbalance the classes. If unbalance of the classes does not exist in the first population, it is followed with the basic procedure of the GA.

The 5 best individuals (with the best aptitude) of all the populations are considered and used to construct the Mixture of Experts.

3 Mixture of Experts

The Mixture of Experts can be seen from two slopes: through the Nearest Neighbor rule (NN) and by means of Modular Neuronal Network (RNM).

3.1 Mixture of Experts with 1-NN

The the NN, known as 1-NN (one nearest neighbor) [2], since its creation by Fix and Hodges in 1951 and 1952, has been one of the methodologies most studied and used for the patterns recognition problem [11], [12]. This technique is a part of the learning based on samples of training controlled.

In the current work the MxE was constructed with the five best individuals obtained with the GA tried with the unbalance datasets and were used as a classifier with 1-NN (it is considered a Euclidian distance). In order to determine the final decision the fusion was chosen: simple majority voting and weighted voting.

In the simple majority voting all members of the MxE have equal weight (Eq. 1) and with weighted voting (Eq. 2) only a member of the MxE determines the final decision [13].

$$\frac{1}{t} \tag{1}$$

t it is the number of classifiers

$$w(r_j) = \frac{1}{d_j} \quad d_j \neq 0 \tag{2}$$

d_j represents the minimum distance (with the neighbor nearest) when classifying a pattern of test, and by the classifier is determined by means of *r_j*.

3.2 Mixture of Experts with MNN

The modular network is constituted by several independent modules, its are mixed by the integrating unit, this last one determines which exits will be combined and which patterns will learn each module. It is possible to see this methodology like the division of a complex problem in its simple parts [14], [15].

In this study, each expert in the Modular Neuronal Network (RNM) is formed by several single neurons (Feedforward), with a entrance layer and a output layer. The neurons of the entrance layer correspond to the number of characteristics of each pattern and the number of neurons of the output layer corresponds to the number of classes. For the integrating network follows the same scheme, with the exception of the exit layer, the number of neurons of the exit layer corresponds to the number of experts (see Fig. 1).

All modules, including the gating network, have *n* input units, that is, the number of features. The number of output neurons in the expert networks is equal to the number of classes *c*, whereas in the gating network it is equal to the number of experts, say *r*. The learning process is based on the stochastic gradient algorithm, being the objective function defined as follows:

$$-\ln \left(\sum_{j=1}^r g_j \cdot \exp \left(-\frac{1}{2} \|s - z_j\|^2 \right) \right) \tag{3}$$

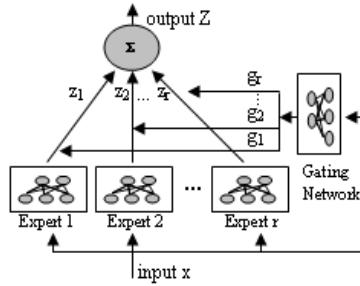


Fig. 1. Representation of the modular neural network architecture. Each module is a feedforward network and receives the same input vector. The final output of the system is the sum of $z_j g_j$.

where s is the output desired for input x , $z_j = xw_j$ is the output vector of the j 'th expert network, $g_j = \frac{\exp(u_j)}{\sum_i \exp(u_j)}$ is the normalized output of the gating network, u_i is the total weighted input received by output unit j of the gating network, and g_j can be viewed as the probability of selecting expert j for a particular case.

The 5 better subsamples are presented to the RNM, they are obtained with the GA treats with the unbalance training sets. The patterns are presented to the RNM randomly with the purpose to avoid that the network learns only the patterns of certain a class.

4 Experimental Results

The data sets used for the experiments were obtained from the UCI Repository (<http://www.ics.uci.edu/~mllearn>). Each one is different in the classes number, in the patterns number and in the features number Table I

Table 1. Databases used

Database	Classes	Features	Training samples	Test samples
Cancer	2	9	546	137
German	2	24	800	200
Heart	2	13	216	54
Liver	2	6	276	69
Pima	2	8	615	153
Sonar	2	60	167	41
Phoneme	2	5	4322	1082

The information contained in the data bases corresponds: to the different tumors that present the patients of the University Hospital of Wisconsin (Cancer), to the different clients of a bank (Geman), to patients who can suffer cardiopathy or not suffer cardiopathy (Heart), to patients that have problems of the liver (Liver), to diabetes problems (Pima), to types of sounds (Sonar), to oral and nasal vowels (Phoneme), to types of crystals (Glass), to images obtained by the Landsat satellite (Satimage) and to types of cars (Vehicle). For the Glass data sets (six classes), Satimage (six classes) and Vehicle (four classes).

Table 2. Adapted datasets

Database	Classes	Features	Training samples	Test samples	Majority class	Minority class
Glass	2	9	174	40	1,2,3,4,5	6
Satimage	2	36	5147	1288	1	2,3,4
Vehicle	2	18	678	168	1,2,3,5,6	4

The experiments were made with an adaptation, that is to say, these data sets are of more than two classes and they are adapted to data sets of two classes Table 2. From each data set mentioned, five partitions were obtained with crossed validation, 80% for the sample of training (ME) and 20% for the test sample (MC).

In Machine Learning has emerged the necessity to evaluate the effectiveness of different methods when classifying the test patterns. In the current investigation is used the geometric mean with the purpose to evaluate the methods previously explained [17,18].

$$G = \sqrt{(-g)(+g)} \quad (4)$$

where $(-g)$ indicate the accuracy for the minority class and $(+g)$ is the majority class accuracy.

The results obtained with the MxE (1-NN/RNM) based on a GA that it tries with the unbalance the classes were compared: with an individual classifier (CI) and with the MxE based on a Genetic Algorithm that not it tries with the unbalance the classes (MxE (1-NN) /AGSB). For the individual classifier it was used as apprenticeship rule the 1-NN, as well as the original data sets. In order to construct the MxE (1-NN) were obtained the best individuals with the GA that not it tries with the unbalance the classes and like apprenticeship rule the nearest neighbor.

In Table 3 there are presented the experiments made with the individual classifier and the MxE based on a GA that it tries with the unbalance the classes, both methodologies have like apprenticeship rule the nearest neighbor, nevertheless with the MxE/AGM also were realized the experiments with a Modular Neuronal Network, in addition was considered the diminution of patterns of 20% and 0% in each chromosome of GA that it tries with the unbalance the classes.

In Table 3 is seen the convenience of using the MxE (1-NN) /AGM with respect to the individual classifier and MxE (RNM) /AGM, nevertheless with the training sample German and Heart was obtained a better result of classification with MxE (RNM) /AGM in comparison with the individual classifier and the MxE (1-NN)/AGM. The MxE (1-NN/RNM) /AGM presents a better result of classification (70%) in comparison with the individual classifier (20%). The standard deviation is indicated between parentheses in Table 3 and Table 4, the best results acquired with the mentioned methodologies and with the used data bases they are indicated with bolds.

In Table 4 are presented the experimental results obtained with the MxE (1-NN) based on GA that not it tries with the unbalance the classes and with the MxE (1-NN/RNM) based on GA that it tries with the unbalance the classes. Both methodologies consider the diminution of patterns of 0% and 20% of each chromosome of GA without trying or treating the unbalance the classes. Can be appreciated a better result of classification with Mixture of Experts (1-NN) based on GA that treats with the unbalance

Table 3. Comparison of the MxE (1-NN/RNM)/AGM with the individual classifier (IC)

Database	IC	MxE based on GA			
		MxE (1-NN)/AGM		MxE (RNM)/AGM	
		Without diminution	With diminution	Without diminution	With diminution
Cancer	94.0(4.1)	96.4(2.9)	96.0(3.2)	86.7(2.0)	84.6(2.9)
German	49.8(8.0)	53.7(7.4)	57.6(8.9)	57.1(19.9)	59.2(20.0)
Heart	58.6(6.2)	66.3(9.0)	65.1(6.3)	56.1(15.1)	59.8(12.2)
Liver	65.2(4.8)	64.5(5.7)	65.9(6.3)	56.1(15.1)	59.8(12.2)
Pima	58.4(8.1)	70.7(16.5)	65.5(2.6)	53.7(7.6)	57.2(13.6)
Sonar	82.0(9.4)	79.5(8.5)	77.9(9.1)	63.0(12.9)	60.6(8.6)
Phoneme	73.8(6.0)	74.0(5.2)	80.4(14.2)	34.5(4.9)	27.7(8.9)
Glass	86.7(12.2)	84.6(16.3)	84.9(16.0)	78.2(15.4)	80.4(15.9)
Satimage	70.9(15.1)	68.6(18.4)	66.4(19.4)	34.4(16.6)	44.0(18.6)
Vehicle	55.8(7.2)	55.2(2.9)	63.3(19.3)	56.8(34.4)	53.1(33.3)

the classes in comparison with: the MxE (1-NN) based on GA that does not treat with the unbalance the classes and with MxE (RNM) /AGM.

Table 4. Comparison of the MxE (1-NN/RNM)/AGM with the MxE(1-NN)/AGSB (IC)

Database	MxE/AGSB		MxE/AGM			
	MxE(1-NN)/AGSB		MxE(1-NN)/AGM		MxE (RNM)/AGM	
	Without diminution	With diminution	Without diminution	With diminution	Without diminution	With diminution
Cancer	95.8(3.4)	96.5(2.4)	96.4(2.9)	96.0(3.2)	86.7(2.0)	84.6(2.9)
German	54.2(5.1)	55.9(4.9)	53.7(7.4)	57.6(8.9)	57.1(16.9)	59.2(20.0)
Heart	64.5(9.0)	64.5(9.0)	66.3(9.0)	65.1(4.7)	76.1(7.5)	77.69(5.74)
Liver	63.4(8.6)	64.4(5.7)	64.5(5.7)	65.9(6.3)	56.1(15.1)	59.8(12.2)
Pima	65.1(4.9)	64.9(5.2)	70.7(16.5)	65.5(2.6)	53.7(7.6)	57.2(13.6)
Sonar	83.1(10.1)	77.4(7.4)	79.5(8.5)	77.9(9.1)	63.0(12.9)	60.6(8.6)
Phoneme	74.1(8.2)	73.7(7.3)	74.0(5.2)	80.4(14.2)	34.5(4.9)	27.7(8.9)

It is possible to mention that with the MxE (RNM) /AGSB and the samples German and Heart it is presented a better result of classification. It is observed a better performance of classification with Mixture of Experts (1-NN/RNM)) based on a GA that treats with the unbalance the classes (71,4%) in comparison with MxE (1-NN) based on GA that not treats with the unbalance the classes (28,6%).

5 Concluding Remarks

By means of the methodology of Mixture of Experts (1-NN/RNM) based on GA that treats with the unbalance the classes, it is possible to increase in some cases the precision in the classification with respect to the results obtained with the individual classifier and to the MxE (1-NN)) based on a GA that not treats with the unbalance the classes.

With the Genetic Algorithm was obtained the best subsamples, that is to say, optimal solutions (quality in the subsamples) of the universe study. The construction of the MxE

was done with those solutions, and the classification was made by means of the MxE: it is used like rule of apprenticeship the nearest neighbor or a RNM. It is possible to mention that the experiments with the MxE (1-NN) /AGSB y with the MxE (1-NN) /AGM were considered 5 experts, with MxE (RNM) /AGM were used 5 experts for the samples glass, satimage and vehicle, and for the rest of the samples was used 3 experts. Generally the results of classification with the methodology MxE (1-NN) /AGM are best in comparison with the others.

Although the GA that treats with the unbalance the classes to takes some time in providing the best individuals, in some cases has obtained good results of classification with the MxE (1-NN/RNM).

Future works: to implement a tournament selection to make the GA more efficient, in computer terms. Also to contemplate in the GA a Scaling methodology with the purpose to avoid that the best individuals lose themselves in each generation or with the operators (crossover and mutation). To consider the selection method in order to determine the final decision in the MxE, where each member of the MxE is sufficiently expert in some region of the space of decisions. To implement an hidden layer in each Expert of the Modular Neuronal Network. To contemplate in the RNM as each expert to a neuronal network LVQ or a classifier gama. Finally, we consider to use and to compare random selection methods (Arc-x4 or Boosting) and non supervised (k-means, SOM) methods for constructing the individual members in the MxE with our proposal.

Acknowledgments

This work has been partially supported by grants: PROMEP/103.5/08/3016, UAEM-CA-114 and SBI112 from the Mexican SEP, and by the 2703/2008U UAEM project.

References

1. Dietterich, T.G.: Ensemble Methods in Machine Learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000)
2. Hernández, J., Ramírez, M.J., Ferri, C.: Introducción a la minería de datos. Prentice-Hall, Englewood Cliffs (2004)
3. Kittler, J., Roli, F.: Multiple Classifier Systems. Springer, Heidelberg (2001)
4. Saddys Francia, S., Moreno García, M.: Multiclassifiers: Applications, Methods and Architectures. In: Proc. of International Workshop on Practical Applications of Agents and Multi-agents Systems, IWPAAMS 2005, pp. 263–271. Universidad de León (2005)
5. Hernández, J.K.: Reducción de muestras de entrenamiento. Tesis de Maestría en Ciencias, en Ciencias Computacionales. Instituto Tecnológico de Toluca. Toluca, México (2002)
6. Goldberg, D.E.: Genetic Algorithms. Addison-Wesley, Reading (1989)
7. Coley, D.A.: An introduction to genetic algorithms for scientists and Engineers. World scientific, Singapore (2005)
8. Mitchell, M.: An introduction to genetic algorithms. MIT Press, Cambridge (1998)
9. Pajares, G., Santos, M.: Inteligencia Artificial e Ingeniería del Conocimiento. Alfaomega (2005)
10. Diaz, R.I., Valdovinos, R.M., Pacheco, J.H.: Comparative Study of Genetic Algorithms and Resampling Methods for Ensemble Constructing. In: Proceedings of IEEE Congress on Evolutionary Computation (2008)

11. Devroye, L., Györfi, L., Lugosi, G.: *A Probabilistic Theory of Pattern Recognition*. Springer, Heidelberg (1997)
12. Cover, T.M., Hart, P.E.: Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory*, Enero (13), 1 (1967)
13. Valdovinos, R.M., Sánchez, J.S., Barandela, R.: Dynamic and static weighting in classifier fusion. In: Marques, J.S., Pérez de la Blanca, N., Pina, P. (eds.) *IbPRIA 2005*. LNCS, vol. 3523, pp. 59–66. Springer, Heidelberg (2005)
14. Fariñas, M., Pedreira, C.E.: Mixture of Experts and Local-Global Neural Networks. In: *Proc. European Symposium on Artificial Neural Networks*, Bruges, Belgium (2003)
15. Bauckhage, C., Thureau, C.: Towards a Fair'n Square Aimbot – Using Mixtures of Experts to Learn Context Aware Weapon Handling. In: *Proc. GAME-ON*, pp. 20–24 (2004)
16. Sánche, O.: *Probabilidad y estadística*. McGraw-Hill, New York (2004)
17. Spiegel, M.R., Stephens, L.J.: *Estadística*. McGraw-Hill, New York (2001)

Opposition-Based Particle Swarm Optimization with Velocity Clamping (OVCPSO)

Farrukh Shahzad, A. Rauf Baig, Sohail Masood, Muhammad Kamran,
and Nawazish Naveed

FAST-NUCES, National University of Computer and Emerging Sciences,
A.K. Brohi Road, 44000, Islamabad, Pakistan
farrukh.shahzad@nexginrc.org, rauf.baig@nu.edu.pk,
{rsmbhatti,mohammad.kamranpk,nawazishnaveed}@yahoo.com
<http://www.nu.edu.pk>

Abstract. This paper presents an Opposition-based PSO(OVCPSO) which uses Velocity Clamping to accelerate its convergence speed and to avoid premature convergence of algorithm. Probabilistic opposition-based learning for particles has been used in the proposed method which uses velocity clamping to control the speed and direction of particles. Experiments have been performed upon various well known benchmark optimization problems and results have shown that OVCPSO can deal with difficult unimodal and multimodal optimization problems efficiently and effectively. The numbers of function calls (NFC) are significantly less than other PSO variants i.e. basic PSO with inertia weight, PSO with inertia weight and velocity clamping (VCPSO) and opposition based PSO with Cauchy Mutation (OPSOCM).

Keywords: Swarm Intelligence, Optimization, PSO, Opposite number.

1 Introduction

Particle swarm optimization (PSO) is a population based stochastic algorithm developed for continuous optimization problem by J. Kennedy et. al. [1]. The inspiration has come from bird flocking or fish schooling in nature. In PSO, each bird is referred as a particle. All particles are initialized with random position and random velocity. Each particle moves to the new position on the basis of its own experience and the experience of all particles in its neighborhood. The particle updates its velocity and position parameters using equation (1) and (2) respectively.

$$v_i(t+1) = v_i(t) + c_1 * r_1 * (pb_i - x_i(t)) + c_2 * r_2 * (gb - x_i(t)) \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

Where x_i , v_i and pb_i are the position, velocity and personal best position of some particle of the i^{th} particle respectively. gb is the global best position of the entire swarm whereas r_1 and r_2 are two numbers whose values are chosen within the range of $[0,1]$. Finally, c_1 and c_2 are called learning factors which specify the relative influence of the social and cognitive components respectively.

It has been empirically investigated that standard PSO could easily fall into local optima in many optimization problems. Some research has been done to overcome this problem [2]. As indicated by equation (1), velocity, and thus the position, of a particle is largely influenced by its personal best position and the global best position. In a particular scenario where a particle's current position is same as that of its personal best position (pb) and global best position (gb) then that particle will not change its position. If this particle happens to be the global best particle of the entire swarm then all the other particles will tend to move towards this particle. The end result is that the swarm converges prematurely to some local optimum. One approach is to incorporate the concept of opposition based learning in PSO as used by H. Wang et. al. [3] and L. Han et. al. [4] to avoid such kind of situations.

In addition to being trapped in a local optimum and premature convergence, another problem with PSO is that when the current position of a particle is quite far from its personal best position and the global best position, the velocity bursts out to large values. This, in turn, has a direct effect on the position of the particle on next time step. So now the particle has large position update as a result of which it may leave the boundaries of the search space. To address this problem we have used velocity clamping that control the global exploration of particles.

In this paper, a new opposition-based velocity clamping PSO algorithm (OVCP SO) is proposed which incorporates the measures to handle the above-mentioned two problems. It avoids premature convergence and allows swarm of particles to continue search for global optima. OVCP SO uses opposition-based learning [5][6] and velocity of particles are clamped to control the speed of particles. Another opposition learning based algorithm with Cauchy Mutation was presented [3], which increases the mutation ability of opposition based PSO to avoid local optima but does not perform convincingly well for multimodal problems. Also, NFC is quite large with this approach. These two facts motivated us to develop our approach OVCP SO. OVCP SO has been tested on both unimodal and multimodal function optimization problems. Comparison of OVCP SO has been done with OPSOCM [3], PSO with inertia weights and velocity clamping (VCP SO) and PSO with inertia weights (PSO).

The rest of the paper is organized as follows: related work is discussed in Section 2, Section 3 describes the new OVCP SO algorithm. Section 4 defines the benchmark continuous optimization problems used in the experiments and gives the experimental settings. Section 5 presents performance comparison and discusses the experimental results. Finally, Section 6 concludes with a summary, a few remarks and future work.

2 Related Work

2.1 Essence of Opposition-Based Learning and PSO

The concept of Opposition-based Learning (OBL) was proposed by H. R. Tizhoosh in [7] and [8]. The opposite y' of a number y can be calculated using equation (3).

$$y' = a + b - y \tag{3}$$

where $y \in \mathbb{R}$ within an interval of $[a, b]$. In the case of multidimensional vector y_i , its opposite vector y' can be calculated using equation (4).

$$y'_i = a_i + b_i - y_i \tag{4}$$

OBL methodology with Differential Evolution (DE) algorithm is used successfully to solve different optimization problems effectively [5][6]. H. Wang et. al. have used OBL with PSO and Cauchy Mutation [3] and L. Han [4] have used OBL with PSO for noisy problems and obtained very improved results as compared to simple PSO. The summary of OBL with PSO is stated below.

Let $f(x)$ be a fitness function for some given problem. Assume $S_i(x_1, x_2, \dots, x_n)$ is a particle in n -dimensional space and x_i ranges from $[a_i, b_i] \forall i \in \{1, 2, 3, \dots, n\}$. And $S_{0i}(x'_1, x'_2, x'_3, \dots, x'_n)$ is the opposite position particle of S_i . In order to remain in the boundaries of converged space of current population, a_i and b_i should be updated dynamically. If $f(S_{0i})$ is better than $f(S_i)$, then update S_i with S_{0i} . The minimum and maximum values ($[a_i^s, b_i^s]$) are used to calculate the opposite vector instead of using some predefined interval boundaries ($[a_i, b_i]$). The opposition-based population can be computed using equation (5).

$$S_{0i,j} = a_j^s + b_j^s - S_{i,j} \tag{5}$$

$S_{i,j}$ represents the position vector j of some particle i and $S_{0i,j}$ represents the opposite solution of $S_{i,j}$. While a_j^s and b_j^s are the minimum and maximum values of dimension j in current population.

2.2 Velocity Clamping in PSO

In usual optimization algorithms *exploration-exploitation trade off* is used to measure the efficiency and accuracy of the algorithm. Exploration ability of algorithm is to explore different regions of the search space in order to find the global optima. While in exploitation, algorithm concentrates on some specific region to glean a candidate solution. A superior optimization algorithm balances the contradictory aspects of exploration and exploitation.

In PSO, velocity update equation (1) manages the aspects of exploration and exploitation detailed in [9]. Traditionally, in basic PSO velocities quickly explode to larger values, especially for the particles distant from the neighborhood best and personal best positions. As a result particles have larger position updates and begin to leave the boundaries of the search space. To handle exploration problem,

velocities of particles are clamped to control the global exploration in search space. If the velocity v of some particle i exceeds a maximum allowed velocity limit, it is set to the maximum value of velocity ($v_{max,j}$). So $v_{max,j}$ denotes the maximum allowed velocity of a particle in j^{th} dimension. The velocity of particles is adjusted using equation (6) before position updates.

$$v_{ij}(t + 1) = \begin{cases} v'_{ij}(t + 1), & \text{if } v_{ij}(t + 1) < v_{max,j} \\ v_{max,j}, & \text{otherwise} \end{cases} \tag{6}$$

Larger values of $v_{max,j}$ cause global exploration, on the other hand smaller values encourage local exploitation. Usually, the $v_{max,j}$ values are selected as a fraction of the domain of each dimension of search space. It controls the movement of particles and exploration and exploitation aspects. Velocity clamping doesn't effect the particle's position. It decreases only the step size determined by velocity and can change the search direction of the particle. The change in search direction may permit the particle for better exploration. But it can also negatively effect and the optimum value could not be found at all. We have used the equations (7) and (8) for initialization of maximum and minimum velocities for our solution.

$$v_{max,j} = \delta(x_{max,j} - x_{min,j}) \tag{7}$$

$$v_{min,j} = \delta(x_{min,j} - x_{max,j}) \tag{8}$$

Where $x_{max,j}$ and $x_{min,j}$ are the maximum and minimum positions of particles in j^{th} dimension, These values are taken from test function's initial x values. δ is a constant factor and is taken 0.1 in our OVCP SO solution. In spite of all this there still exist an issue with velocity clamping. If all velocities become equal to v_{max} , particle may continue to search on the boundaries of some hypercube and may subtle upon some optima but would not be able to converge in this local area. This issue can be resolved by multiple ways. In our solution we have used inertia weights to overcome the problem.

2.3 Inertia Weight

The idea of decreasing inertia weight w was proposed by Y. Shi et. al. [10]. Inertia weight controls that how much memory of the previous flight direction will influence the new velocity. If $w > 1$ then velocities increase over time, particles accelerate towards maximum velocity and the swarm diverges. If $w < 1$, particles decelerate until their velocities reach zero. Larger values of w facilitate exploration while smaller value promote exploitation. In our solution, the values of w are assigned between 0.9 and 0.4. For each iteration, w is decremented gradually from 0.9 to 0.4. Equation (9) provides the velocity update with inertia weights.

$$v_i(t + 1) = \mathbf{w} * v_i(t) + c_1 * rand_1() * (pbest_i - x_i(t)) + c_2 * rand_2() * (gbest - x_i(t)) \tag{9}$$

Algorithm 1. *Opposition-based PSO with Velocity Clamping (OVCP SO)*

```

n ← initial population size;
S ← current swarm population;
S0 ← the opposite population of swarm S;
xjs ← [ajs, bjs]{interval boundaries of the jth dimension in population S}
po ← probability for opposition calculation
best_fitness_value_so_far ← Selection of fittest particle
VTR ← desired optimum value
MaxNFC ← number of function calls (NFC)
Calculating the opposition-based solution of initial population
for i = 1 to n do
    Calculate the opposite particle S0,i using equation (5);
end for
Calculate fitness of each particle in opposite vector S0;
Select n fittest particles from S and S0 to create a population of size n;
Update pbesti, gbest ;
while best_fitness_value_so_far > VTR and NFC ≤ MaxNFC do
    if rand(0,1) < po then
        Update the interval boundaries [ajs, bjs] in current population;
        for i = 1 to n do
            Calculate the opposite particle S0i using equation (5);
        end for
        Calculate fitness each particle in opposite vector S0;
        Select n fittest particles in S and S0 to create a new population os size n;
        Update pbesti, gbest;
    else
        for each particle Si do
            Calculate particle velocity using equation (9)
            Update particle position using equation (2)
            Calculate fitness value of particles Si
            Update pbesti, gbest;
        end for
    end if
    if the fitness of gbest' is preferred than gbest then
        gbest ← gbest'
    end if
    if the velocity > vmax,j or velocity < vmin,j then
        Clamp the velocity to vmax,j or vmin,j using equations (7) and (8)
    end if
end while
end procedure

```

3 OVCP SO Algorithm

Opposition-based learning and velocity clamping are two fundamentals of our OVCP SO algorithm. In Algorithm (1) we initialize an original swarm S and then we generate its opposite swarm S_0 . Each swarm consists of N number of particles. We calculate the fitness of both swarms (S and S_0), merge and sort them on the basis of their fitness. Finally select N best particles from the collection of $2N$ particles. After the initialization step the algorithm continues in while loop until any of the stopping criteria is reached. The stopping criteria in our case are MaxNFC i.e. 100,000 and the optimum value of test functions. For each iteration in while loop following three steps are performed.

1. Apply the equations for updating the velocity and positions and clamp the velocity if it crosses the upper limit in some dimension (i.e. $v_{max,j}$).
2. Calculate the opposite particles according to an opposite probability p_0 .
3. Compare the fitness of every particle in both the swarms out of $2N$ particles and select the best N particles to form the new swarm.

4 Experiments and Results

We have performed the experiments using the algorithms, basic PSO with inertia weight w , PSO with velocity clamping and inertia weight (VCPSO) and opposition-based PSO with velocity clamping and inertia weight (OVCP SO). The fitness of test functions and number of function calls of 50 trials are recorded and reported. The performance of all these algorithms is compared with each other and also with opposition-based PSO with Cauchy Mutation, it will be referred in next sections as OPSOCM [3].

4.1 Benchmark Functions with Brief Description

We have tested our algorithm on eight (all minimization) renowned test functions. The brief description of these functions given in Table 1. The functions f_1, f_2, f_3, f_4 are unimodal and f_5, f_6, f_7, f_8 are multimodal functions. The optimum value of test function f_5 is -12569.5 and all other functions have 0 as optimum value (i.e. 10^{-25} or below). The detailed description of these functions is available in [11].

Table 1. Description of test functions used for algorithm’s evaluation

Function Name	Definition	Dims	X	Min
Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	30	[-5.12,5.12]	0
Weighted Sphere	$f_2(x) = \sum_{i=1}^n i * x_i^2$	30	[-5.12,5.12]	0
Quartic	$f_3(x) = \sum_{i=1}^n i * x_i^4 + random[0, 1]$	30	[-1.28,1.28]	0
Rosenbrock	$f_4(x) = \sum_{i=1}^n [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$	30	[-30,30]	0
Schwefel	$f_5(x) = \sum_{i=1}^n -x_i * \sin\left(-\sqrt{ x_i }\right)$	30	[-500,500]	-12569.5
Rastrigin	$f_6(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	[-5.12,5.12]	0
Ackley	$f_7(x) = -20 * e^{(-0.2 * \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2})} - e^{(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i))} + 20 + e$	30	[-32,32]	0
Griewangk	$f_8(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600,600]	0

4.2 OVCP SO Parameters Initialization

The initial value of PSO parameters are as follows: $c_1=1.49612$ and $c_2=1.49612$, inertia weight w is calculated using equation (10).

$$w_{Iter} = w_{Max} - \left(\frac{w_{Max} - w_{Min}}{Iter_{Max}} \right) * Iter \tag{10}$$

Where $w_{Max}=0.9$ and $w_{Min}=0.4$, $Iter$ = current iteration number and $Iter_{Max}$ = maximum number of iterations. The value of w gradually decreases from 0.9 to 0.4 with increase in number of iterations. Minimum and maximum values of velocities are calculated using equations (7) and (8). To gauge the performance of

our algorithm, number of function calls (NFC) are computed and compared for different versions of PSO. All other common parameters used in our solution are similar to the initial values of parameters used in [3], for the sake of a fair comparison. Other common parameters are: population size= 10, maximum NFC=100,000 and opposition probability $p_0 = 0.3$.

5 Performance Comparisons and Discussion

Minimum fitness values achieved by algorithms on functions ($f1$ to $f8$) are listed in Table 2. Mean, median, best and worst fitness values are reported . Table 3 provides the number of function calls in which the algorithms attained different fitness values. Mean, minimum (Min) and maximum (Max) number of function calls along with standard deviation (STDEV) are reported. The performance of algorithms over individual functions is discussed below.

$f1$: The performance of OVCP SO is remarkable on function $f1$. It achieved its optimum minima i.e. 0. Both OVCP SO and OPSOCM converged to the optimum value 0 but the performance of OVCP SO is better than OPSOCM in terms of functions calls. It converged in 22230 which is 35747 calls less than OPSOCM. PSO and VCP SO couldn't reach the optimum values.

Table 2. Mean, median, best and worst fitness of PSO, VCP SO and OVCP SO

	PSO - Fitness				VCP SO - Fitness				OVCP SO - Fitness			
	Mean	Median	Best	Worst	Mean	Median	Best	Worst	Mean	Median	Best	Worst
$f1$	16.04	14.3598	5.6728	40.9215	7.4828	5.6386	0.4734	20.9475	0	0	0	0
$f2$	371.73	398.2062	123.4785	472.6122	88.5931	71.5598	26.5750	161.2851	0	0	0	0
$f3$	3.8526	2.5602	1.0042	13.2189	1.8916	1.1647	0.5146	6.9100	1.4e-3	9.56e-4	1.62e-4	3.5e-3
$f4$	6.7e6	6.6e6	2.5e6	1.6e7	1.4e6	9.7e5	8.4e4	6.4e6	28.7012	28.7272	28.5518	28.7583
$f5$	-3324.6	-2641.5	-3403.7	-1043.8	-4905.5	-4836.9	-6195.7	-3543.8	-11618	-10709	-12299	-8763.7
$f6$	207.6588	205.7647	177.0517	242.8048	170.8818	182.3447	112.1837	221.6075	0	0	0	0
$f7$	15.3498	15.1969	13.5297	17.0427	12.4623	12.6366	9.3112	18.9863	0.0891	0.0891	0.0891	0.0891
$f8$	84.5817	79.7629	54.1204	151.2837	14.3627	14.1731	5.9813	34.5196	0	0	0	0

$f2$: OVCP SO has shown incredible results on function $f2$ also. It converged to the optimum minima (i.e. 0), in just 27800 number of function calls. It's mean NFC is 22230 calls less than OPSOCM. PSO and VCP SO could not perform well on $f2$.

$f3$: OVCP SO could not reach the optimum value on function $f3$. It converged to the mean and best fitness values 1.4e-3 and 1.62e-4 respectively, in maximum number of function calls. We performed further experiments by reducing the maximum number of iterations and came to know that OVCP SO can achieve this local minimum value even in almost 36000 NFC. OPSOCM achieved, mean and best values i.e. 1.84e-2 and 1.14e-2 in 100,001 NFC. So OVCP SO performed better than OPSOCM, in terms of optimum value. PSO and VCP SO could not perform well on $f3$.

$f4$: OVCP SO hasn't converged to the optimal minima of the function $f4$. It prematurely converged to values 28.7012 and 28.5518, in the case of mean and best fitness using maximum number of function calls. This local minima can be achieved by OVCP SO even in 14000 NFC. Other algorithms VCP SO and PSO

Table 3. Performance comparison of PSO, VCPSO and OVCPSO

	PSO				VCPSO				OVCPSO			
	Mean	Max	Min	STDev	Mean	Max	Min	STDev	Mean	Max	Min	STDev
$f7$	100,000	100,000	100,000	0	100,000	100,000	100,000	0	22230	26730	12250	3789.7
$f2$	100,000	100,000	100,000	0	100,000	100,000	100,000	0	21582	27800	10390	4174.2
$f3$	100,000	100,000	100,000	0	100,000	100,000	100,000	0	100,000	100,000	100,000	0
$f4$	100,000	100,000	100,000	0	100,000	100,000	100,000	0	100,000	100,000	100,000	0
$f5$	100,000	100,000	100,000	0	100,000	100,000	100,000	0	100,000	100,000	100,000	0
$f6$	100,000	100,000	100,000	0	100,000	100,000	100,000	0	19752	23340	9470	4250.4
$f7$	100,000	100,000	100,000	0	100,000	100,000	100,000	0	100,000	100,000	100,000	0
$f8$	100,000	100,000	100,000	0	100,000	100,000	100,000	0	20159	24880	8390	3521.6

performed worst in the case of $f4$ and could converge to 7 to 8 digits positive numbers. OPSOCM used a different definition of $f4$ so we can't compare it with OVCPSO. The definition of $f4$ used in this paper is taken from [11] and [12].

$f5$: In the case of $f5$ OVCPSO couldn't reach its optimum value. But it's performance is better than OPSOCM. It reached the mean and best fitness values -11595 and -12299 respectively, in 100,000 number of function calls. OPSOCM achieved mean and best fitness values -10986.7 and -11250.6 respectively, in 100,000 NFC. So OVCPSO performed better than OPSOCM. PSO and VCPSO performed better on $f5$ but not comparable to OVCPSO.

$f6$: On function $f6$, OVCPSO has given extraordinary results. It converged to its optimum value 0 in just 19752 number of function calls. All other algorithms PSO, VCPSO and OPSOCM could not converge in the case of function $f6$, even in maximum number of function calls 100,000. PSO and VCPSO also prematurely converged on $f6$

$f7$: On $f7$, OVCPSO could not achieve notable results as shown in Figure 1. It trapped to value $8.91e-2$ in maximum limit of NFC i.e. 100,000. This local minima can also be achieved in almost 14000 NFC. But still the performance of OVCPSO is better than OPSOCM that also prematurely converged to value 1.19, in the case of mean fitness with 100,000 NFC. But in the case of best fitness OPSOCM performed better and converged to 0. Both PSO and VCPSO prematurely converged on $f7$.

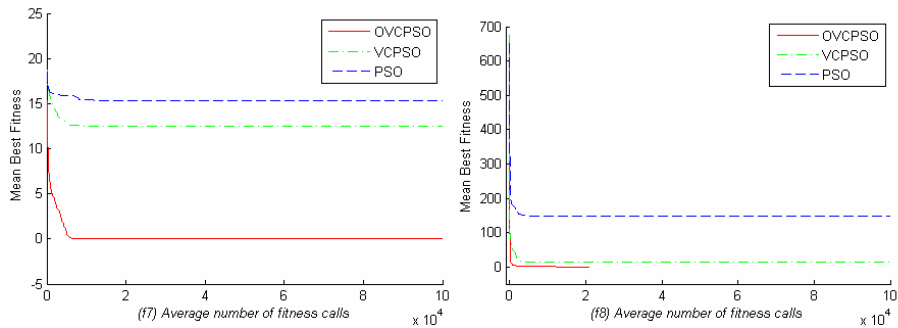


Fig. 1. Performance comparison of algorithms on functions $f7$ and $f8$. Similar patterns have been observed in all other converging and prematurely converging functions.

Table 4. Multiple values of probability (p_0) used for OVCPSO

	$p_0=0.3$		$p_0=0.5$		$p_0=0.7$		$p_0=0.9$	
	MBFitness	NFC	MBFitness	NFC	MBFitness	NFC	MBFitness	NFC
$f1$	0	22230	0	28749	0	31818	0	40394
$f2$	0	27800	0	30965	0	32633	0	39024
$f3$	0.0014	100,000	0.0048	100,000	0.0124	100,000	0.0465	100,000
$f4$	28.7012	100,000	28.6729	100,000	28.6349	100,000	28.6103	100,000
$f5$	-11618	100000	-8745.8	100,000	-8371.2	100,000	-4805.2	100,000
$f6$	0	19752	0	30121	0	43183	0	61693
$f7$	0.891	100,000	0.891	100,000	0.891	100,000	0.891	100,000
$f8$	0	20159	0	27820	0	34024	0	42718

$f8$: OVCPSO achieved significant results in the case of $f8$ as shown in Figure 1. It converged to its optimum value 0 in just 20159 average number of function calls. OPSOCM could converge to $4.72e-2$ and 0 in its mean and best cases respectively, in 96442 NFC. So OVCPSO performed better than OPSOCM both in terms of early response and fitness. PSO and VCPSO couldn't get performance even near to OVCPSO.

The detailed analysis of algorithms performance and results depicts that OVCPSO converged to optimum minima of functions $f1$, $f2$, $f6$ and $f8$. On functions $f3$, $f5$, $f5$ and $f7$ OVCPSO could not achieve the optimum minima but still its performance in terms of quick response (average number of fitness calls) and convergence level is better than other algorithms i.e. OPSOCM, VCPSO and PSO. This proves that PSO performs significantly better than its already available variants when it is applied with OBL and velocity clamping.

5.1 OVCPSO and Probability Ranges

The opposite particle x' of a particle x is calculated with the probability p_0 . Different values of p_0 (i.e. 0.3, 0.5, 0.7, 0.9) were investigated but $p_0=0.3$ showed the best results. OVCPSO acts as VCPSO with $p_0=0$. Table 4 depicts the results of OVCPSO with different values of p_0 . MBFitness indicates the mean best fitness and NFC represents the average number of function calls over 50 trials of the algorithm.

6 Conclusion and Future Work

Opposition based learning is employed in a variant of classical PSO. By applying velocity clamping with OBL on entire population probabilistically, OVCPSO showed better performance than OPSOCM, VCPSO and PSO on eight benchmark test functions (4 unimodal and 4 multimodal). Experiments confirmed better performance of OVCPSO on both unimodal and multimodal functions. However, for some functions OVCPSO could not find the optimum results, trapped into local minima. Future work will be, to tackle the shortcomings of

OVCPSO and also its application on large number of benchmark test functions. Also the opposite probability p_0 can be made adaptive for better results.

References

1. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: IEEE International Conference on Neural Networks, Perth, Australia (1995)
2. Parsopoulos, K.E., Plagianakos, V.P., Magoulas, G.D., Vrahatis, M.N.: Objective Function “stretching” to Alleviate Convergence to Local Minima. *Nonlinear Analysis TMA* 47, 3419–3424 (2001)
3. Wang, H., Liu, Y., Li, C., Zeng, S.: Opposition-based Particle Swarm Algorithm with Cauchy Mutation. *IEEE CEC*, 4750–4756 (2007)
4. Han, L., He, X.: A novel Opposition-based Particle Swarm Optimization for Noisy Problems. In: Proceedings of the Third International Conference on Natural Computation, vol. 3, pp. 624–629. IEEE Press, Los Alamitos (2007)
5. Rahnamayan, S., Tizhoosh, H.R., Salama, M.M.A.: Opposition-based differential evolution algorithms. In: IEEE CEC, Vancourver, BC, Canada (2006)
6. Rahnamayan, S., Tizhoosh, H.R., Salama, M.M.A.: Opposition-based differential evolution for optimization of noisy problems. In: IEEE CEC, Vancourver, BC, Canada (2006)
7. Tizhoosh, H.R.: Opposition-based learning: A new scheme for machine intelligence. In: International Conference on Computational Intelligence for Modeling Control and Automation - CIMCA 2005, Vienna, Austria, val. I, pp. 695–701 (2005)
8. Tizhoosh, H.R.: Opposition-based reinforcement learning. *Journal of Advanced Computational Intelligence and Intelligent Informatics* 10(3) (2006)
9. Engelbrecht, A.P.: Fundamentals of Computational Swarm Intelligence. John Wiley & Sons, Chichester (2006)
10. Shi, Y., Eberhart, R.: A Modified Partilce Swarm Optimzer. In: Proceedings of the IEEE CEC, Piscataway, NJ, pp. 69–73 (1998)
11. GEATbx: (example functions),
http://www.geatbx.com/docu/fcnindex-01.html#P160_7291 (viewed on 24-05-2009)
12. Mathworld (Rosenbrock function),
<http://mathworld.wolfram.com/RosenbrockFunction.html>

Designing a Compact Genetic Algorithm with Minimal FPGA Resources

Alejandro León-Javier, Marco A. Moreno-Armendáriz, and Nareli Cruz-Cortés

Centro de Investigación en Computación - IPN
Av. Juan de Dios Batíz s/n Unidad Profesional Adolfo López Mateos
Col. Nueva Industrial Vallejo, Mexico City, 07738, Mexico
ljavierb07@sagitario.cic.ipn.mx,
{marco_moreno,nareli}@cic.ipn.mx

Abstract. The Compact Genetic Algorithms (cGA) are searching methods used in different engineering applications. These algorithms have interesting features such as their capability to operate with very low memory resources while solving complex optimization problems. In this paper we present a novel design for the implementation of a cGA on a FPGA. This design is modular, so its components would be used for designing other Evolutionary Algorithms.

Keywords: Compact Genetic Algorithm, FPGA implementation, minimal resources.

1 Introduction

The Compact Genetic Algorithms (cGA) are searching methods based on the behavior of the standard Genetic Algorithm with uniform crossover [1]. The cGA are very useful tools for applications where there are only few resources of memory and a complex optimization problem must be solved [2].

For applications that require real time processing while doing global numerical optimization, the Evolutionary Algorithms, such as the Genetic Algorithms, can be a competitive option; however most of them require a huge amount of resources and execution time. To overcome this problem the cGA were developed [3]. In our opinion, a competitive cGA implemented in hardware, must accomplish the three following requirements: 1) it should be based on a top-down methodology with a group of basic synchronized components through a machine states, 2) to use minimal resources, and 3) it must have the ability to operate in real time. So, we decided to implement it by using a FPGA with the VHDL language.

In the specialized literature, we can find some works implementing Evolutionary Algorithms based on a hardware platform. For example, in [4] it was proposed the design of a Genetic Algorithm (GA) working as an optimization engine on a FPGA. In that work the authors take advantage of some important features exhibited by a FPGA, such as parallelization and reprogramming, to increase the GA's speedup.

Some other proposals presented VLSI-based architectures for coarse and fine grained parallel GA [5].

In [6] a chip LSI is described. It includes hardware, reconfigurable logic hardware, a memory to maintain the individuals, a training data memory and, a 16-bits CPU core.

In [7] a cGA design and implementation was presented by using language VERILOG.

Gallagher et al., [8] proposed to add elitism and mutation to the cGA. These features improved the solutions found by the cGA in a very important manner. In addition to that, they proposed to use a modular design to implement this algorithm on micro-controllers using VHDL.

In [9] the authors proposed a parallel cellular cGA topology which is implemented on a FPGA.

In addition to the mentioned proposals, there exist other GA implementations. Some of them require more than one FPGA. All these disadvantages show up the necessity of having a cGA version properly implemented on a hardware platform.

2 The Compact Genetic Algorithm

The cGA represents a population of potential solutions to a problem, through a probability distribution over the set of possible solutions, called a Probability Vector (*PV*). This *PV* is processed by an updating rule based on the typical operations of selection and crossover of a GA [2]. The *PV* is initialized with value 0.5 and, at the end of the algorithm's execution it must contain only 0's or 1's. This final vector represents the solution obtained by the cGA.

The pseudo code of the cGA is presented in Figure 1. The input parameters for this algorithm are the following: population size (n) and the chromosome length (l). A population is represented by a 1-dimensional Probability Vector (*PV*). The $PV[i]$ is the probability that the i^{th} -position bit of an individual, randomly picked up from the population, will be one. First, the *PV* is initialized to (0.5, 0.5, ..., 0.5). Next, the individuals a and b are generated according to the *PV*. Then, the fitness value of the individuals is compared and, the element with better fitness is named the winner and the other one is called loser. Now, if $winner[i] \neq loser[i]$, the *PV* of the winner will be updated as follows: If the $winner[i]=1$ then the *PV* will be increased by $1/n$, if $winner[i] \neq loser[i]$ the *VP* will be decreased by $1/n$. Note that if $winner[i] = loser[i]$, the *VP* will not be updated. The loop is repeated until each $PV[i]$ becomes zero or one. Finally, *PV* represents the final solution [7].

3 VHDL Design

For the VHDL design we need to consider different issues. The cGA algorithm needs that the *PV* contains real numbers and its updating rule (plus or minus $1/n$) must keep its values into the interval (0,1). However, for this implementation, we decided to use a random generator [10] that generates integer numbers in the range (0, 2147483647).

```

1) Initialize probability vector
   for i:=1 to l do VP[i]:=0.5;

2) Generate two individuals from the vector
   a:=generate(VP);
   b:=generate(VP);

3) Let them compete
   winner, loser:=compete(a,b);

4) Update the probability vector towards the better one
   for i:=1 to l do
     if winner[i]≠loser[i] then
       if winner[i]=1 then
         VP[i]:= VP[i]+1/n;
       else
         VP[i]:= VP[i]-1/n;

5) Check if the vector has converged
   for i:=1 to l do
     if VP[i]>0 and VP[i]<1 then
       return to step 2;

```

Fig. 1. Pseudo code for the cGA [3]

Then, this means that the values in the *PV* must be in the same range because we need to do comparisons between these elements.

In this work, we defined different components in order to obtain the possibility to perform different calculus at same time. These components are explained next:

Random Number Generator (RNG). This component generates the random numbers used for the individual's generation. There exists different kinds of algorithms, we decided to use a very efficient one proposed by Park & Miller in [10], and it generates the next random number I_{j+1} based on the following expression:

$$I_{j+1} = aI_j \bmod m. \quad (1)$$

With $a = 7^5 = 16807$ and $m = 2^{31}-1 = 2147483647$. In order to use this RNG with 32 bits, we must join it with the Schrage algorithm [10]. So, the new form of the expression is:

$$I_{j+1} = aI_j \bmod m = \begin{cases} a(I_j \bmod q) - r[I_j / q] & \text{If } I_{j-1} \geq 0 \\ a(I_j \bmod q) - r[I_j / q] + m & \text{Otherwise} \end{cases} \quad (2)$$

With $q = 127773$ and $r = 2836$.

Individual Generator (IndGen). This component receives two random numbers and the PV. It generates two bits (*rng1* and *rng2*), the first bit (*rng1*) corresponds to the individual one (*ind1*) and, the second bit (*rng2*) to the individual two (*ind2*) at the same position. The conditions for the generation of the numbers are shown in Figure 2.

```

if rng1<=pv then
    ind1<= '1' ;
else
    ind1<= '0' ;
end if;
if rng2<=pv then
    ind2<= '1' ;
else
    ind2<= '0' ;
end if;

```

Fig. 2. Pseudo code of the individual generator component

Fitness Evaluator (FEv). The evaluation is made by using two individuals in the following way: First, we measure the fitness of each individual and compare them, if the fitness of the individual one is greater or equal than the fitness of the individual two, then the winner is the individual one, otherwise the individual two wins (assuming that we are handling a maximization problem). The definition of the fitness function depends on the kind of optimization problem to be solved. Once the fitness function is selected, this function must be implemented in VHDL code to finish this component.

Probability Vector Updating (PVU). To update the PV three data are received: 1) who is the winner, 2) the bit from individual 1 in the current position and 3) the bit from individual 2 in the current position. Then we apply the conditions presented in Figure 3.

Where:

- **op** indicates which individual is the winner. The value '1' is for the number one and '0' for the number two.
- **oe** is used to specify if the current bits of both individuals are equal (value '1') or not (value '0').

The conditions $vect < 2147483623(\max)$ and $vect > 23(\min)$ are required to restrict the variations of the vector to the interval (0, 2147483647), for the reason that we are working with elements of 32 bits. The max and min values of the variable *vect* are defined by the number of individuals in the population, in our design we use 50, so the step size is 42949672. It is possible to modify this number by changing the values of the step size and the max and min values.


```

if (oe='1' and ind1='1') then
  if (op='1' and vect<2147483623) then
    vect<=vect+42949672;
  elsif (op='0' and vect>23) then
    vect<=vect-42949672;
  end if;
elsif oe='1' and ind1='0' then
  if (op='1' and vect>23) then
    vect<=vect-42949672;
  elsif (op='0' and vect<2147483623) then
    vect<=vect+42949672;
  end if;
end if;
end if;

```

Fig. 3. Conditions for updating the PV

Probability Vector Checking (PVC). Here we verify if the probability vector (*PV*) already converges to the max or min values, when this event occurs the flag is set to *RDY='1'*.

Compact Genetic Algorithm (cGA). This is the main component and is based on the other components. To obtain a correct synchronization we use a finite state machine, presented in the next section.

4 Finite-State Machine

A finite-state machine is a sequential circuit with a finite number of states. The usage of this machine guaranties the correct behavior of the whole design. The machine is

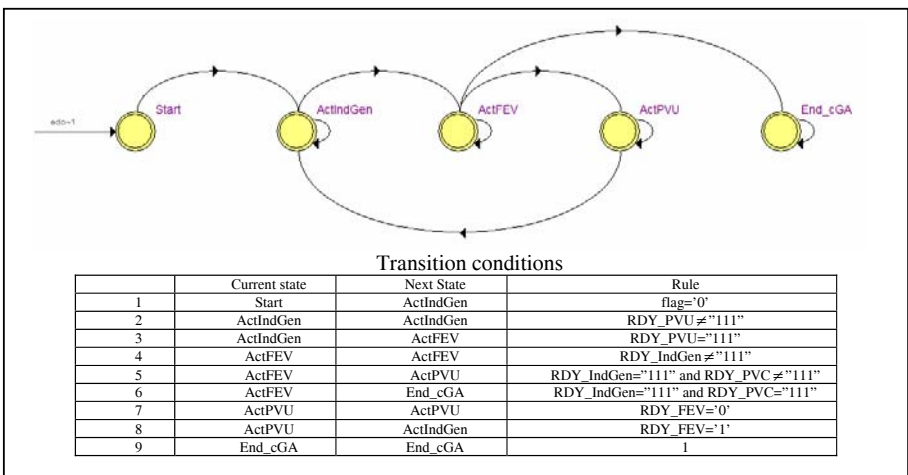


Fig. 4. Finite-state of the cAG component

defined by two functions, the first one calculates the next state of the system and the second one the output [11].

There are two kinds of finite-state machines, the Mealy's and the Moore's. In our case we use the Mealy. This type of machine uses a clock as synchronized signal for the transitions. We used it for the cGA component as it is shown in Figure 4. For the transitions, all the components include a READY signal (as example: RDY_PVU, RDY_PVC, etc.), so based on these signals the correct behavior is guaranteed.

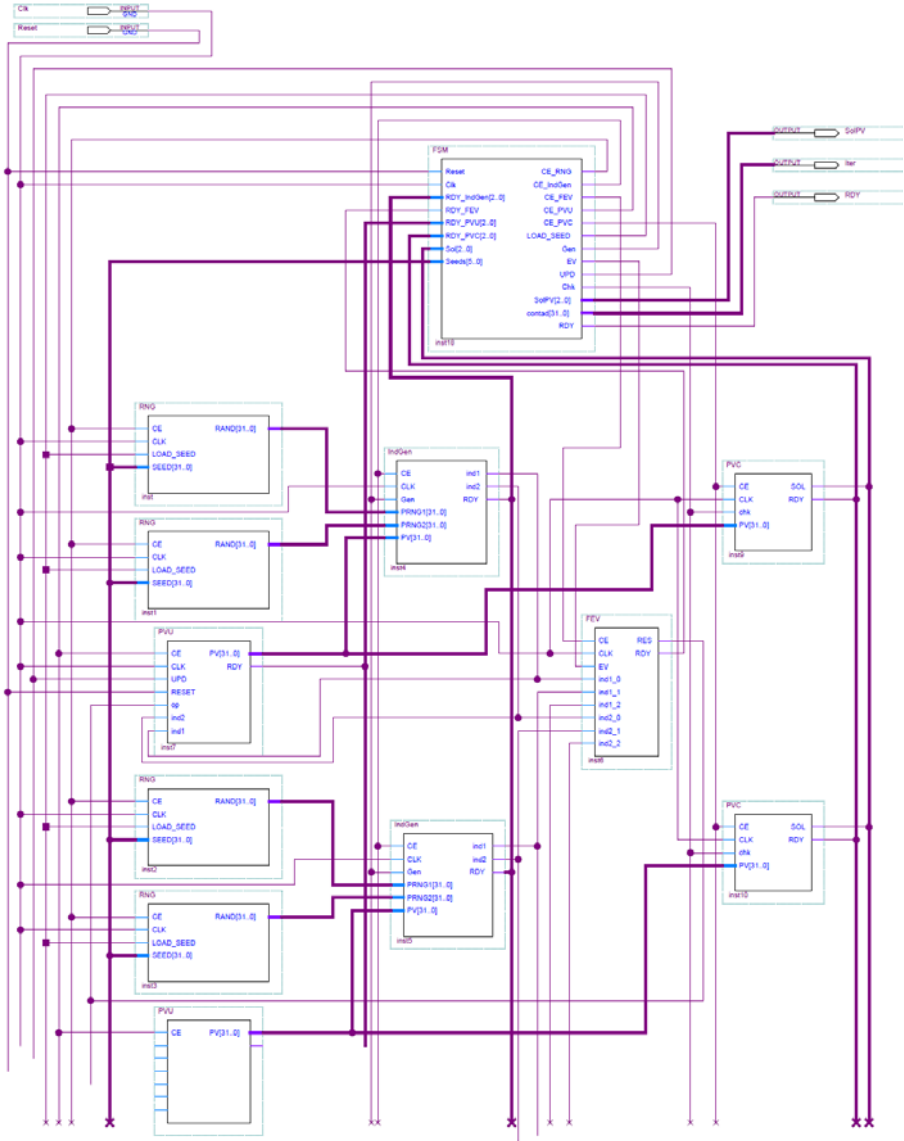


Fig. 5. Interconnection of cGA's components

The READY signals containing bit arrays indicate if all the components of the same type have finalized their operation, that is, all bits should be "1".

In the START state, the finite-state machine initializes the RNG components, that is, the random seeds are initialized with some proposed values then, and the RNG components are activated generating the random numbers. The ActIndGen state is the component responsible of the individuals' generation. It is activated and in turn, the component that verifies the convergence of the PV. The ActFEv state determines which the winner individual is, furthermore the algorithm finishes if the PVC component announces that the vector has converged. The PVU state activates the block that updates the PV. The End_cGA state activates the RDY flag of the main block of the cGA indicating that the algorithm has finished and showed the PV.

Figure 5 shows the interconnection of the cGA's components.

5 Experiments and Results

For the experiments, it is necessary to add or remove components for processing the PV with a desired length, that is, to scale our design. It is important to mention that the objective function evaluation is the more costly part (in terms of processing time) in the Genetic Algorithms, therefore, the implementation of it impacts in the execution time of the algorithm either in hardware or software.

To evaluate our cGA we conducted some simulation tests on a Quartus II of Altera, which is a software tool for analysis and synthesis of HDL designs. We accomplished the simulation using a Cyclone II EP2C70F896C6 obtaining the results shown on Table 1. The selected problem for the simulations was the max-one, which consists in maximize the number of one's of a bitstring, this implies a simple objective function, which is not costly.

Table 1. Simulation results

Max-One	Simulation Results
8-bits	Run time: 40.28μs
	Iterations: 335
	Combinational functions: 18796/68417 Dedicated logic registers: 826/68416 Embedded Multiplier 9-bits elements: 192/300
12-bits	Run time: 41.48μs
	Iterations: 345
	Combinational functions: 28149/68417 Dedicated logic registers: 1214/68416 Embedded Multiplier 9-bits elements: 288/300

Furthermore, we accomplished software tests on a PC with 3 GB of RAM and Core 2 Duo processor to 1.80 GHz. The cGA was programmed of the same manner than in VHDL, but on a sequential way on the language C#. The obtained results are shown in Table 2.

Table 2. Results in software

Max-One	Simulation Results
8-bits	Run time: 1.4996 s Iterations: 335
12-bits	Run time: 2.1840s Iterations: 345

The Table 3 shows the features of a synthesized Random Number Generator component.

It is important to note that the generation of random numbers can be costly according to the algorithm that we used. For our case, this component occupies many elements of the FPGA (embedded multipliers), since we required two RNG components for each *PV*'s bit, which allows parallelizing this part of the algorithm.

One way of decreasing the usage of the FPGA's resources for the component RNG, is using a cellular automata. The cellular automata only require a memory cell to store each bit of the random number and a combinational function representing the rules of the automata. This reduces the consumption of resources in a FPGA. Besides with cellular automata we can generate numbers of 8, 16, 32 and 64 bits of length.

Table 3. RNG Component

Component	Data Synthesis
Random Number Generator	Combinational functions: 1108/68417 Dedicated logic registers: 64/68416 Embedded Multiplier 9-bits elements: 12/300

6 Conclusions and Future Work

The Compact Genetic Algorithms (cGA) are a viable option to compute global numerical optimization. Their main feature is that they need only very few computational resources when compared against the classic Genetic Algorithms. For this reason and because their inherent parallelism, the cGA are especially well suited to be implemented in a hardware architecture.

In this work we presented some experiments implementing a cGA on hardware and software platforms. As it was expected, the hardware version obtained the quickest results.

Meanwhile in the hardware version the results were obtained in only few microseconds, for the software version, they required several seconds. This feature makes the cGA especially attractive to be used in real time processing applications.

As a future work, we are planning to implement a fitness function design capable of handling floating point numbers.

Acknowledgment

The authors thank the support of the Mexican Government (CONACYT, SNI, SIP-IPN, COFAA-IPN, and PIFI-IPN), also we appreciate the support of Altera Corporation and Víctor Maruri for the donation of DE2-70 kit and Quartus II academic licenses.

References

1. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computation, 1st edn. Springer, Heidelberg (2003)
2. Cupertino, F., Mininno, E., Lino, E., Naso, D.: Optimization of Position Control of Induction Motors using Compact Genetic Algorithms. In: IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics, pp. 55–60 (2006)
3. Harik, G., Lobo, F.G., Goldberg, D.E.: The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation* 3(4), 287–297 (1999)
4. Scott, S., Seth, A.: HGA: A hardware-based genetic algorithm. In: Proceedings ACM/SIGDA 3rd Int. Symposium Field-Programmable Gate Arrays, pp. 53–59 (1995)
5. Yoshida, N., Yasuoka, T.: Multi-gap: Parallel and distributed genetic algorithms in VLSI. In: Proceedings Int. Conf. Systems, Man, Cybernetics, Tokyo, Japan, October 1999, vol. 5, pp. 571–576 (1999)
6. Kajitai, T., et al.: A gate-level EHW chip: Implementing GA operations and reconfigurable hardware on a single LSI. In: Sipper, M., Mange, D., Pérez-Uribe, A. (eds.) ICES 1998. LNCS, vol. 1478, pp. 1–12. Springer, Heidelberg (1998)
7. Apornawan, C., Chongstitvatana, P.: A Hardware Implementation of the Compact Genetic Algorithm. In: Proceedings 2001 IEEE Congress Evolutionary Computation, Seoul, Korea, vol. 1, pp. 624–629 (2001)
8. Gallagher, J.C., et al.: A Family of Compact Genetic Algorithms for Intrinsic Evolvable Hardware. *Proceedings 2004 IEEE Transactions on Evolutionary Computation* 8(2), 111–125 (2004)
9. Jewajinda, Y., Chongstitvatana, P.: FPGA Implementation of a Cellular Compact Genetic Algorithm. In: NASA/ESA Conference on Adaptive Hardware and Systems, June 2008, vol. 22, pp. 385–390 (2008)
10. Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T.: Numerical Recipes in C: The Art of Scientific Computing, 2nd edn. Cambridge University Press, Cambridge (1992)
11. Pardo, F., Boluda, J.A.: VHDL Lenguaje para síntesis y modelado de circuitos, 2nd edn. Alfaomega, México (2004) (in Spanish)

Application of DNA Self-assembly on Maximum Clique Problem

Guangzhao Cui¹, Cuiling Li¹, Haobin Li¹, Xuncaizhang², and Xiaoguang Li¹

¹ Henan Key Lab of Information-based Electrical Appliances, 450002, Zhengzhou, China

² Department of Control Science and Engineering, Huazhong University of Science and Technology, 430074, Wuhan, China

Abstract. Computation by self-assembly of DNA is an efficient method of executing parallel DNA computing, in which information is encoded in DNA tiles and a large number of tiles can be self-assembled via sticky end associations. Here, we investigate how basic ideas on tiling can be applied to solve maximum clique problem (MCP). We suggest that these procedures can be realized on the molecular scale through the medium of self-assembled DNA tiles. By creating billions of copies of the participating DNA tiles, the algorithmic will run in parallel on all possible cliques. The potential of DNA computing by self-assembly for this problem is promising given the operational time complexity of $\Theta(n)$. This work shows further evidence for the ability of DNA computing to solve NP-Complete problems.

Keywords: Self-Assembly, MCP, Tiling, Nondeterministic, DNA Computing.

1 Introduction

Since the seminal work of Adleman on the HPP [1], DNA computing has experienced a flowering growth and leaves us with a rich legacy. Given its vast parallelism and high-density storage, DNA computing approaches have been employed to solve many combinatorial optimization problems. However, most of these proposals implement the computation by performing a series of biochemical reactions on a set of DNA molecules, which require human intervention at each step. Thus, one difficulty with such methods for DNA computing is the number of laboratory procedures.

Winfrey et al. first proposed the idea of computation by self-assembled tiles. Computation and self-assembly are connected by theory of tiling, of which Wang tile [2] is a prime example. Because DNA tiles can be more easily “programmed” to incorporate the constraints of a given problem, it is possible to exercise some degree of controlling over the appearance of considerable waste of material that belongs to the traditional approach, and therefore parallel computation can be enhanced by self-assembling process, where information is encoded in DNA tiles. The relation of DNA computation to self-assembling structures was developed in the mid-1990s, largely through the theoretical and experimental work of Winfrey et al [3], [4], Seeman [5], Reif [6] and Rozenberg et al. [7]. Computational systems based on self-assembly have been demonstrated in both 1-D arrangements called “string tiles” [3], [8] and 2-D

lattices [4]. Other stable forms of nucleic acids include Z-DNA, non-migrating Holliday junctions and duplexes with triple crossovers or “pseudoknots” [5], [9], [10]. For 2-D self-assembly, Winfree has proposed the Tile Assembly Model (TAM)[4] and demonstrated that it is Turing- universal by showing that a tile system can simulate Wang tiles which Robinson has shown to be universal [11].

The first experimental demonstrations of computation using DNA tile assembly was in Ref. [10]. It demonstrated a two-layer, linear assembly of triple-crossover (TX) tiles that executed a bitwise cumulative XOR computation. Barish et al. demonstrated a tile system that copies an input and counts in binary [12]. Cook et al. used the TAM to implement arbitrary circuits [13]. Similarly, Rothmund et al. demonstrated a DNA implementation of a tile system that computes the XOR function, resulting in a Sierpinski triangle [14]. Brun proposed and studied some systems that compute the sum and product of two numbers using the TAM [15]. He then combined these systems to create systems with more complex behavior, and designed two systems which factor numbers and decide subset sum problems [16], [17]. We proposed two systems that compute the difference and quotient of two numbers [18].

2 DNA Self-assembly

DNA nanostructures provide a programmable methodology for bottom-up nanoscale construction of patterned structures, utilizing macromolecular building blocks called DNA tiles based on branched DNA. These tiles have sticky ends that match the sticky ends of other DNA tiles, facilitating further assembly into larger structures known as DNA tiling lattices.

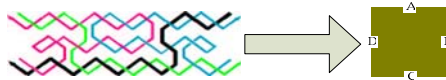


Fig. 1. Abstracted tile representation

2.1 DNA Tile

Branched DNA molecules provide a direct physical motivation for the TAM. There is also a logical equivalence between DNA sticky ends and Wang tile edges. These DNA tiles have unpaired ends of DNA strands sticking out and through these sticky ends, they can attach themselves with other tiles having the Watson-Crick complementary sticky end. Thus, these tiles can stick with one-another to assemble into complex superstructures and through this process they can compute.

The most relevant constructions for DNA computing by self-assembly of DNA tilings include double-crossover (DX) [19] and TX [20] complexes. These tiles can be constructed using a variety of possible nucleotide sequences. We can use different sequences to denote different symbols or values. For example, we can have one sequence denoting the value 0 and another sequence denoting the value 1. We can also use the sticky ends of a tile to encode certain values or symbols. The tile that

attaches to this tile would have the complementary sticky end encoding the same value or symbol. In this way, we can pass information from one tile to another.

2.2 Molecular Self-assembly Processes

Molecular self-assembly is the ubiquitous process by which simple objects autonomously assemble into intricate complexes. It has been suggested that intricate self-assembly processes will ultimately be used in circuit fabrication, nanorobotics, DNA computation, and amorphous computing [21].

There are three basic steps that define a process of molecular self-assembly [22]. 1. Molecular recognition: elementary molecules selectively bind to others. 2. Growth: elementary molecules or intermediate assemblies are the building blocks that bind to each other following a sequential or hierarchical assembly. 3. Termination: a built-in halting feature is required to specify the completion of the assembly. Without it, assemblies can potentially grow infinitely, in practice, their growth is interrupted by physical and/or environmental constraints. On the other hand, molecular self-assembly also has three characteristics [22]: 1. It is a time-dependent process and because of this, temporal information and kinetic control may play a role in the process before thermodynamic stability is reached. 2. It is a highly parallel process, where many copies of different molecules bind simultaneously to form intermediate complexes. 3. Another characteristic of a molecular self-assembly is that the hierarchical build-up of complex assemblies allows one to intervene at each step.

2.3 Programming Self-assembly of DNA Tiling

A tiling is an arrangement of a few tiles that fit together perfectly in the infinite plane. Programming DNA self-assembly of tiling amounts to the design of the pads of DNA tiles, ensuring that only the adjacent pads of neighboring tiles are complementary so that tiles assemble together as intended. The use of pads with complementary base sequences allows the neighbor relations of tiles in the final assembly to be intimately controlled, thus the only large-scale superstructures formed during assembly are those that encode valid mappings of input to output. The progress of self-assembly of DNA tiling can be carried out by the following four steps: 1. Mixing the input oligonucleotides to form the DNA tiles. 2. Allowing the tiles to self-assemble into superstructures. 3. Ligating strands that have been co-localized. 4. Performing a single separation to identify the correct output.

3 Maximum Clique Problem

We denote an undirected graph by $G=(V; E)$, where V is the set of vertexes and E is the set of edges. Two vertexes are said to be adjacent if they are connected by an edge. A clique of a graph is a set of vertexes, any two of which are adjacent. Cliques with the following property has been studied over the last three decades: maximal clique, whose vertexes are not a subset of a larger clique, in other words, it is the largest among all cliques in a graph. Vertices 1, 2, 5 form a clique in the graph shown in Fig.2, because each has an edge to all the others.

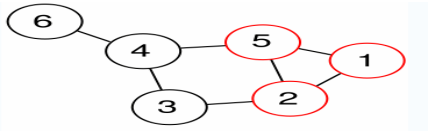


Fig. 2. A graph of maximum clique problem

4 DNA Self-assembly for MCP

There are well-known algorithms for MCP. However, all known algorithms for this problem have exponential worst-case complexity, and hence there are instances of modest size for which these algorithms require an impractical amount of computer time to render a solution. Because the MCP has been proved to be NP-complete, it seems likely that there is no efficient (polynomial time) algorithms to solve it.

This work proposes a way to perform this search procedure on molecular substrate using 2D DNA self-assembly. All searches could be done in parallel by an exponential number of computers, completing the whole computation in linear time.

4.1 Non-deterministic Search Cliques

In nondeterministic computation, the cell which connect to some tile is nondeterministic, in our system the first connected tile is the nondeterministic one. Aiming at MCP, clique scheme should be searched stochastically, according to the parallelism of computing. That is to say, for each assembly body, selecting one vertex from $\{v_1, v_2, \dots, v_n\}$, to be the searching point randomly. This is a straightforward system that will not need much explanation. The bisque tile is the non-determination tile in Fig.3(a), and some important computing tiles shown in Fig.3(b). The concepts include variables v_i and the middle of the tiles' value $E(i,j)$ refer to formula (1).

$$E(i,j) = \begin{cases} OK & \text{if inputs are } v_i, v_j \text{ and } (v_i, v_j \in E_{ij}) \\ NO & \text{if inputs are } v_i, v_j \text{ and } (v_i, v_j \notin E_{ij}) \\ blank & \text{otherwise} \end{cases} \quad (1)$$

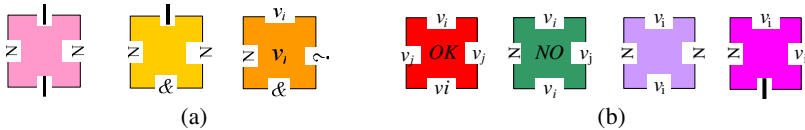


Fig. 3. Computation boundary tiles (a), and the tiles used in search clique system (b)

We have just described a tile system non-deterministically choice a searching point(vertex). Now given a graph G whose adjacency matrix is A . If v_i, v_j are connected by an edge then $a_{ij}=1$, otherwise $a_{ij}=0$. How can we construct a clique which contains the selected vertex? First we define a function $E(i,j)$, if two vertexes connected by an edge in graph G , then $E(i,j)=OK$, otherwise $E(i,j)=NO$ and if inputs are not both $v_i(i=1, 2 \dots n)$, $E(i,j)$ is not gotten value. Second, we need to check up

which vertexes adjacent to the selected vertex. If we chose v_1 as the first vertex, v_3, v_4, v_6, v_7, v_8 can be find connect with v_1 in matrix A , then we chose the first element v_3 of the new set $V_1=\{v_3, v_4, v_6, v_7, v_8\}$ as next standard vertex. Third, we check up which vertexes connect with v_3 in V_1 set, v_4, v_6, v_7, v_8 are found comply to our requirement. Now, we get a new set $V_2=\{v_4, v_6, v_7, v_8\}$ again, here, we check up which vertexes connect with v_4 in V_2 , v_8 is the only one which meet our requirement. Last, a clique is found, which is the maximal clique that contains vertex v_1 .The clique includes four vertexes v_1, v_3, v_4, v_8 . All in all, we can represent the desired computation in a table format which is transferred to DNA self-assembly easily. Table I describes a search scheme, 9 vertexes represented in the rightmost column of the table, and counting from right to left, bottom to up. Fig.5 shows the self-assembly body according to Table1.

$$A = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Table 1. The general situation for searching operation

END	v_8	v_4	v_1	v_3	
	(v_8)	(v_4)	(v_1)	NO(v_3)	v_9
	NO(v_8)	OK(v_4)	OK(v_1)	OK(v_3)	v_8
		v_8	v_8	v_8	
		NO(v_4)	OK(v_1)	OK(v_3)	v_7
			v_7	v_7	
		NO(v_4)	OK(v_1)	OK(v_3)	v_6
			v_6	v_6	
		(v_4)	NO(v_1)	OK(v_3)	v_5
			v_5	v_5	
		NO(v_4)	OK(v_1)	OK(v_3)	v_4
			v_4	v_4	
			(v_1)	NO(v_3)	v_3
			(v_1)	NO(v_3)	v_2
			NO(v_1)	OK(v_3)	v_1
			v_1	v_1	
				v_3	?
					S

Intuitively, the table is filled in a bottom-up manner, one column at a time. Each computing cell is filled by $E(i,j)$ ($i, j = 1, 2, \dots, n$). These rules for constructing the search table can be translated into a set of geometrical tiles, known as Wang tiles. Fig.3(b) shows the computing tiles, each tile has two input sides (south and east) and two output sides (north and west) and the middle of tile is its value $E(i,j)$. The purple tile must satisfied the south input side is v_i and the east input side is a rotated 'N', It

indicates those cells are not computing, their output sides only pass the information of their input sides. The function of the magenta tile is to get the information of v_i which is the first tile of the set V_j that has been just formed by its right column, and passing v_i to its upper cell through its north side.

Input of the system is provided in the form of a preassembled arrangement of tiles which encodes the mathematical value of the input. The input DNA structure is shown in Fig.4(a) and the computation result of each step will be kept in its left column that maybe become input of the next circle computing.

4.2 The Non-deterministic Algorithm for MCP

In some implementations of the tile assembly systems, many assemblies happen in parallel. In fact, it is often almost impossible to create only a single assembly. In this case, there are many possible valid tiling, any or all of which may be produced. Here, we designed the non-deterministic algorithm to solve it. The basic idea is to exploit the massive parallelism possible in DNA self-assembly in order to solve the MCP in linear time. By creating billions of copies of the participating DNA tiles, we expect that the procedure will run in parallel on all possible cliques. These cliques will be created dynamically as part of the assembly. In effect, that will make the computation time linear in the size of the problem, while pushing the exponential dimension of the problem into the large number of DNA assemblies, and thus into space occupied by the DNA molecules.

Our design is described at the algorithmic level. It attempts to simulate a nondeterministic algorithm for the problem. Nondeterministic implies that at some steps the algorithm make a nondeterministic choice (as if some oracle could tell the algorithm what the right choice is). In our case, this simulated by an exponential number of DNA assemblies, expecting to cover all possible cliques. The algorithm is given below. Notice that step 2 is the nondeterministic step.

```

Non-Deterministic Algorithm (E, V) {{
1) for ( i=1, ..., n) do
2)   Assign a vertex  $v_i$  to begin searching a clique
3)   for ( j=1, ..., n) do
4)     {Check up which vertex( $v_j$ ) connect with  $v_i$  and build a
       new set  $V_k$  that contains all met requirement  $v_j$ .}
5)     ...
6)   If there are not vertexes are adjacent in the assemble
    system
7)   Then return and output}
8) Compare length of these reporter strands, find the
   longest one.
9) Get the maximum clique of the graph}

```

The idea can be carried out almost directly by assembly of DNA tiles. We now present the system builds on L-shape seed configuration. This system will use the two sides of the L-configuration to encode inputs, and produce its result on the top row of an almost complete rectangle. Systems could chain computations together, using the output of this computation as an input to another computation. The input structure that encodes V on the rightmost is shown in Fig.4(a). This input structure is reproduced in

billions and are mixed with all kinds of computing tiles which from a fixed library. The appropriate tiles will self-assemble on those input layers. Use the massive parallelism of DNA to find out all possible cliques in the destination graph.

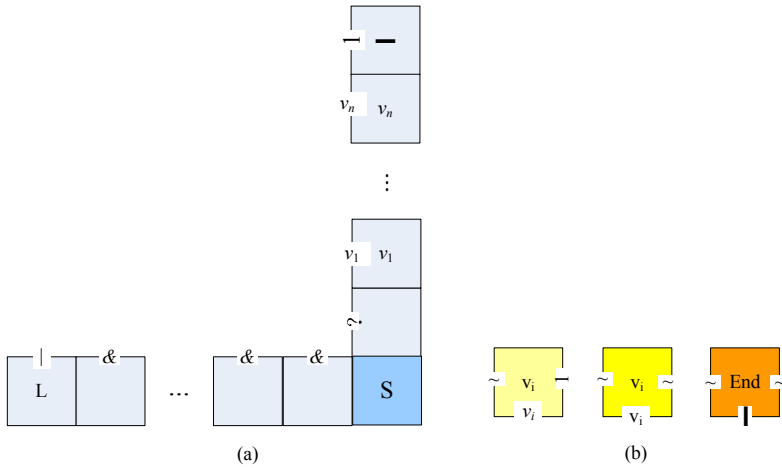


Fig. 4. The L seed configuration (a), and top output tiles (b)

Follow the algorithm above, the appropriate tiles will self-assemble on this input structure. Reading of the operation result is done by the reporter strand method [10]. In order to output the result in the form of a DNA strand, we add some output tiles of the top shown in Fig.4(b). On adding these tiles, and allowing them to anneal, we get the final tile assembly. On adding ligase to seal the bonds, we will have a single strand of DNA passing through the tiles in the final output layer, which encodes the result of the computation. Endpoints of this kind of single strands with unique nucleotide sequences labeled ‘ v_i ’ (the light yellow tile) and ‘End’ shown in Fig.4(b). Therefore, we will get n differences strands through PCR, which utilizing endpoints tiles as primer. Now, we have gotten all sequences that indicates all possible cliques, then we can find the longest strand(s) after gel electrophoresis. In the end, getting the maximum clique through measure DNA sequences.

As an example, let us consider applying this algorithm to matrix A , a successful assembly computation is shown in Fig.5. The actual implementation detail is not discussed here since they fall outside of the scope of this paper. However, we believe that we make no arbitrary hypotheses. In fact, our work is based on the assumptions and achievements that come with DNA tiling computation in general.

4.3 Complexity Analysis

The complexity of the design is considered in terms of computation time, computation space and the number of distinct tiles required. It is obvious from the example given that the computation time T is equal to the width (depth) of the assembly. In fact, it is $T \leq n+n=\Theta(n)$. It follows directly from the assembly time corollary in Ref.[15]. The space S taken for each assembly is the area of the assembly. $S \leq (n+3)* (n+2)= \Theta(n^2)$.

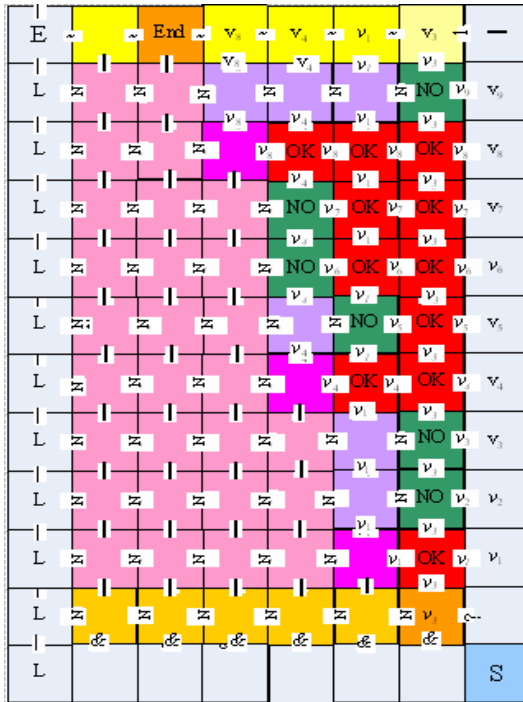


Fig. 5. A final configuration with the output $(v_3, v_1, v_4, v_7, v_5)$ on the top row

Finally, the library of fixed tiles needed contain the following tiles:

Variables. There have to be n tiles coding n variables, n is the number of vertexes.

Boundaries. There are five tiles which shown in Fig.6 to mark the beginning of computation in the input assembly, the end of variable column, the last tile of the bottom line, the non-determination tile and the tile as the most body of the bottom line whose number is decided by the scale of the special question. In our instance, we get six which is the two-thirds of the total vertexes' number.

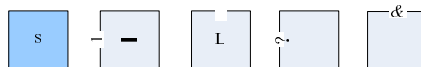


Fig. 6. Boundary tiles of DNA self-assembly

Computation and Computation Boundaries. For search operations there has to be four different tiles, they are shown in Fig.3(b). For search operation complete smoothly, there has to be two connection tiles and a nondeterministic tile shown in Fig.3(a).

Output. Finally, there have to be output tiles to output results. There are $2*n+1$ tiles shown in Fig.4(b). Summing up all the numbers, we have a total number of tiles: $N=n+6+4+3+2*n+1=\Theta(n)$.

5 Conclusions

As the massive parallelism of DNA computing, it has the unexampled dominance in solving difficult problems, especially for NP complete problems. In this study, a completely different DNA computing approach for solving MCP is proposed. The advantage of our method is that once the initial strands are constructed, each operation can compute fast almost together through the process of DNA self-assembly without any participation of manpower.

A final detail is crucial for the success of algorithm. The concentrations of assignment tiles corresponding to v_i have to be equal, so that there is equal chance of assigning each vertex. Otherwise, there might be assignments that will never be explored because of this “discrimination” in assigning values.

A limitation of the algorithm, which is common for most DNA computations, comes from the fact that the exponential dimension of the problem has been pushed into the physical space (volume) occupied by the DNA molecules. This will eventually become a restrictive factor. The input size cannot grow forever, it implies an upper bound to the size of instances that can be solved in practice.

The field of DNA self-assembly holds tremendous promise. Our ultimate goal is to test the design experimentally. However, there are still many technical hurdles to overcome before self-assembly can be developed into a reality. If the molecular word can be controlled at will, then it may be possible to achieve vastly better performance for computers and memories. We hope that this paper helps to demonstrate that molecular computing is a technology worth pursuing.

Acknowledgments. This paper is supported by the National Natural Science Foundation of China (Grant Nos. 60573190, 60773122), Basic and Frontier Technology Research Program of Henna Province (Grant No. 082300413203), and Innovation Scientist and Technicians Troop Construction Project of Henna Province.

References

1. Adleman, L.M.: Molecular Computation of Solutions to Combinatorial Problems. *Science* 266, 1021–1024 (1994)
2. Wang, H.: Proving Theorems by Pattern Recognition I. *Bell System Technical Journal* 40, 1–42 (1961)
3. Winfree, E., Eng, T., Rozenberg, G.: String tile models for DNA computing by self-assembly. In: Condon, A., Rozenberg, G. (eds.) *DNA 2000*. LNCS, vol. 2054, pp. 63–88. Springer, Heidelberg (2001)
4. Winfree, E.: *Algorithmic Self-Assembly of DNA*. Ph.D. Dissertation, California Institute of Technology, Pasadena, CA (1998)
5. Seeman, N.C.: DNA Nanotechnology: Novel DNA Constructions. *Annual Review of Biophysics and Biomolecular Structure* 27, 225–248 (1998)

6. Reif, J.H.: Computing: Successes and Challenges. *Science* 296, 478 (2002)
7. Rozenberg, G., Spink, H.: DNA Computing by Blocking. *Theoretical Computer Science* 292, 653 (2003)
8. Winfree, E., Liu, F., Wenzler, L.A., Seeman, N.C.: Design and Self-Assembly of Two-Dimensional DNA Crystals. *Nature* 394, 539 (1998)
9. Mao, C., Sun, W., Seeman, N.C.: Designed Two-Dimensional DNA Holliday Junction Arrays Visualized by Atomic Force Microscopy. *J. Am. Chem. Soc.* 121, 5437 (1999)
10. Mao, C., LaBean, T.H., Reif, J.H., Seeman, N.C.: Logical Computation Using Algorithmic Self-Assembly of DNA Triple-Crossover Molecules. *Nature* 407, 493–496 (2000)
11. Robinson, R.M.: Undecidability and Nonperiodicity for Tilings of the Plane. *Inventiones Mathematicae* 3, 177 (1971)
12. Barish, R., Rothmund, P., Winfree, E.: Two Computational Primitives for Algorithmic Self-Assembly: Copying and Counting. *Nano Letters* 5(12), 2586–2592 (2005)
13. Cook, M., Rothmund, P., Winfree, E.: Self-Assembled Circuit Patterns. In: *Proceedings of the 10th International Meeting on DNA Based Computers*, pp. 91–107 (2004)
14. Rothmund, P., Papadakis, N., Winfree, E.: Algorithmic Self-Assembly of DNA Sierpinski Triangles. *PLoS Biology* 12, 2041 (2004)
15. Brun, Y.: Arithmetic Computation in the Tile Assembly Model: Addition and Multiplication. *Theoretical Computer Science* 378, 17–31 (2006)
16. Brun, Y.: Nondeterministic Polynomial Time Factoring in the Tile Assembly Model. *Theoretical Computer Science* 395, 3–23 (2008)
17. Brun, Y.: Solving NP-Complete Problems in the Tile Assembly Model. *Theoretical Computer Science* 395, 31–44 (2008)
18. Zhang, X.C., Wang, Y.F., Chen, Z.H., Xu, J., Cui, G.Z.: Arithmetic Computation Using Self-Assembly of DNA Tiles: Subtraction and Division. *Progress in Natural Science* (in print, 2009)
19. Li, X., Yang, X., Qi, J., Seeman, N.C.: Antiparallel DNA Double Crossover Molecules as Components for Nanoconstruction. *J. Am. Chem. Soc.* 118, 6131 (1996)
20. Liu, D., Park, S.H., Reif, J.H., LaBean, T.H.: DNA Nanotubes Self-Assembled from Triple-Crossover Tiles as Templates for Conductive Nanowires. *Proceedings of the National Academy of Science*, 717–722 (2004)
21. Carbone, A., Seeman, N.C.: Molecular Tiling and DNA Self-Assembly. In: *Jonoska, N., Păun, G., Rozenberg, G. (eds.) Aspects of Molecular Computing*. LNCS, vol. 2950, pp. 61–83. Springer, Heidelberg (2004)
22. Reif, J.H., LaBean, T.H., Seeman, N.C.: Challenges and applications for self-assembled DNA nanostructures. In: *Condon, A., Rozenberg, G. (eds.) DNA 2000*. LNCS, vol. 2054, pp. 173–198. Springer, Heidelberg (2001)

Modeling of the Relative Humidity and Control of the Temperature for a Bird Incubator

Jose de Jesus Rubio¹, Martin Salazar², Raul Lugo³, Jaime Pacheco¹,
and Angel D. Gomez²

¹ Instituto Politecnico Nacional - ESIME Azcapotzalco
Seccion de Estudios de Posgrado e Investigacion
Av. de las Granjas 682, Col. Sta. Catarina. Azcapotzalco, Mexico D.F.
jrubioa@ipn.mx

² Tecnologico de Estudios Superiores de Ecatepec
Division de Ingenieria Mecatronica Industrial
Av. Tecnologico, Col. Valle Anahuac, Ecatepec, Edo. de Mex.

³ Universidad Autonoma Metropolitana - Iztapalapa
Departamento de Ingenieria de Procesos e Hidraulica
Av. Sn. Rafael Atlixco 186, Col. Vicentina Iztapalapa, Mexico D.F.

Abstract. In this paper the mathematical model for the control of temperature in an bird incubator is presented. Two classic laws of control are compared for the control of temperature: the on-off (*ON – OFF*), and the proportional integral (*PI*). It is proposed a functional network for the modeling of the relative humidity behavior in the bird incubator. It is gotten that the *PI* is the best controller.

1 Nomenclature

h_i	Coefficient of convection in the surface [$Wm^{-2}C^{-1}$]
T_i	Inside temperature of the incubator [$^{\circ}C$]
T_{pi}	Temperature in the wall inside of the incubator [$^{\circ}C$]
h_e	Coefficient of convection in the exterior [$Wm^{-2}C^{-1}$]
T_{ext}	Outside temperature of the incubator [$^{\circ}C$]
T_{pe}	Temperature in the wall outside of the incubator [$^{\circ}C$]
T_o	Initial temperature [$^{\circ}C$]
λ	Thermic conductivity [$Wm^{-2}C^{-1}$]
e	Width of the cover [m]
A	Area of the surface [m^2]
Σ	Constant of Stefan Boltzman [$Wm^{-2}C^{-1}$]
T_{foco}	Temperature of the radiation body (light bulb) [$^{\circ}C$]
h_r	Coefficient of the radiant hot [$Wm^{-2}C^{-1}$]
ε	Emission of the object [-]
k_v	Hot capacity of the air [$Wm^{-2}C^{-1}$]
Φ_{vent}	Air flow [m/s]
a_1, a_2, a_3, a_4	Constants [-]

U_{ws}, U_{ls}	Degree of temperature [%]
W	Speed of the air [ms^{-1}]
A_1	Area of opening [%]
C_{cap}	Thermic capacity of the air [$Wm^{-2}C^{-1}$]
Q_{cal}	Energy of the heating system [Wm^{-2}]
Q_{vent}	Lost energy for ventilation [Wm^{-2}]
Q_{rad}	Energy of radiation [Wm^{-2}]
I	Intensity of the luminous body [<i>Candela</i>]
r	Distance from the luminous body to the objective [m]
e_{ss}	Error in the stable state

2 Introduction

In the operation of a bird incubator, it is necessary to consider two variables, the temperature and the relative humidity with the objective to get a good environment for letting the birds to grow.

Actually, in the market, there are bird incubators that work with ON-OFF controllers and are designed for only one special kind of birds. In addition, the bird incubators do not get the measure of the humidity.

It is important to propose a control system which lets us to change the temperature for several kind of birds and a system that lets us to get the measure of the humidity.

If the inside temperature of the bird incubator is controlled, the relative humidity is controlled, too.

In this paper, the inside temperature of the bird incubator is controlled, as a consequence, the relative humidity is controlled, too.

3 The Bird Incubator System

The bird incubator system is an experimental environment called incubator. Inside of the incubator are deposited eggs of some kind of bird (for example chick, duck, quail or ostrich) over a metallic mesh. In the bottom there is a container with water. In the top there are some heater devices (focus), one in the right and another in the left side. In the top and in the bottom there are holes to let the mixing of the air inside and outside of the incubator. In the top there is a fan to regulate speed of the air in the incubator, the correct speed is $0.2 m/s$. The structure of the incubator is shown in Fig. 1.

The phenomenon is shown in Fig. 1. The heating devices let the increase in the temperature to follow a reference temperature (the desired temperature), it depends of the kind of bird that is in the incubator. At the same time the fan moves the hot air to all the incubator. The water in the bottom is evaporated giving that the humidity is sent to the eggs. Finally, the air goes outside through the holes that are in the top and in the bottom.

In addition, Fig. 1 shows some devices that are in the incubator as are the container of water or the place where the eggs are placed. The arrows describe the direction of the hot air flow in the incubator.

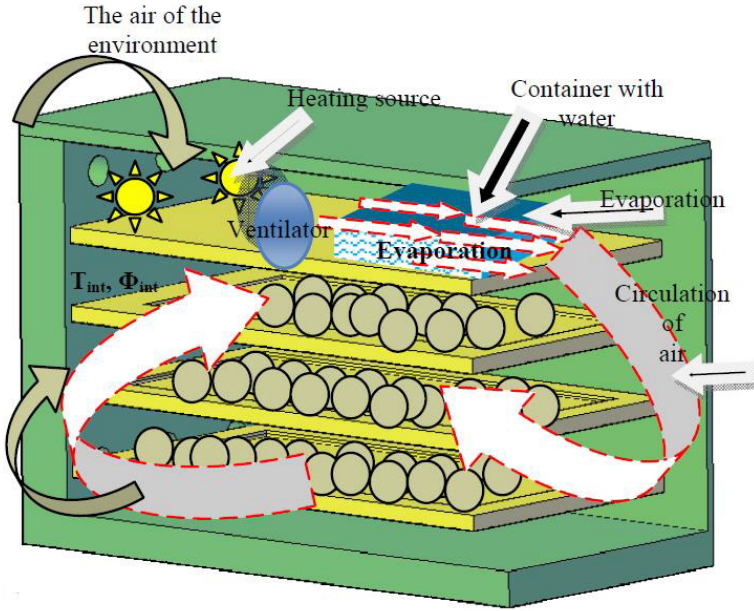


Fig. 1. Structure of the incubator

The control gives an input such that the output of the plant can follow the reference as best as possible. For this system the inside temperature of the incubator should be $37\text{ }^{\circ}\text{C}$. The sensor gives the measure of the temperature inside of the incubator. The control implemented in one micro-controller lets that the actuator sends a signal to the system to get the desired output (the reference). The control is based in varying the intensity of two light bulb of 100 W each one.

4 Mathematical Model of the Temperature of the Incubator

The mathematical model of the incubator is based in two parts: the first one is the thermic in the interior and the second one is the humidity. For the mass and energy balance in the incubator is given by conduction, convection and radiation.

It is considered that the inside temperature of the incubator is bigger than outside of the incubator ($T_i > T_o$). In this way, the hot from inside to outside of the incubator is considered as a difference of temperature $\Delta T = T_i - T_o$.

The differential equation that models the system is the derivative of the inside temperature of the incubator with respect to the time and is equal to the inverse of the hot capacity and the sum of the thranfered hot in the system:

$$\frac{dT_i}{dt} = \frac{1}{C_{cap}}(Q_{cal} + Q_{rad} - Q_{vent}) \quad (1)$$

Considering the dimensions of the incubator and the conditions of the environment it is possible to have the following data:

$$\begin{aligned} Q_{cal} &= \frac{I}{4410} \\ Q_{vent} &= 2.31(T_i - T_o) \\ Q_{rad} &= 39.147 \end{aligned} \quad (2)$$

As the equation (1) is a differential equation of first order, it can be written as a state equation. First the state, the input and the output of the system are:

$$\begin{aligned} x &= T_i \\ u &= I \\ y &= x \end{aligned} \quad (3)$$

And the state equation is:

$$\dot{x} = \frac{1}{C_{cap}} \left(\frac{u}{4410} - 2.31x + 2.31T_o + 39.147 \right) \quad (4)$$

Or can be written as:

$$\dot{x} = -\frac{2.31}{C_{cap}}x + f(t) + \frac{1}{4410C_{cap}}u \quad (5)$$

Where:

$$f(t) = \frac{1}{C_{cap}}(2.31T_o + 39.147) \quad (6)$$

The equation that represents the mathematical model of the inside temperature of the incubator is (4) or (5).

5 Modeling of Relative Humidity Behavior Using Functional Networks

It is not easy to obtain a mathematical model to control the relative humidity, that is way it is proposed an approximation of this behavior via functional networks. It is possible to get the measure of the relative humidity as a consequence of the control of the temperature inside of the incubator. The functional network is similar to a neural network. The difference between the neural networks and the functional networks is that while the first one uses a sigmoid, hyperbolic tangent or signum functions letting to get a local minimum in the approximation of a behavior, the second one uses base functions which makes that the functions of this network are independent one to another one letting to get a global minimum in the approximation of a behavior.

In the functional networks, it is important the organization of the neurons and the properties of the functionals giving a resultant network [2], [6]. A functional network has an input layer which contains the input data, an output layer which contains the output data and one or many processor layers which evaluate the

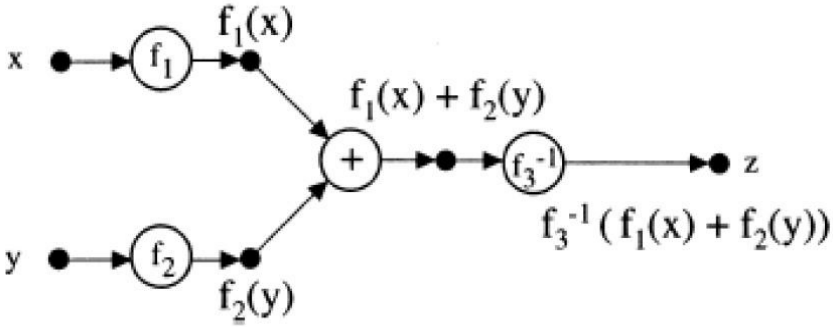


Fig. 2. Unit model

data coming from the previous layer and gives some data that will be used in the subsequent layer. Each processor layer has an functional unit. One functional unit is a set of nodes with data (Y, F, Z) , where Y, Z are disjoint sets not empty of X and $F : Y \rightarrow Z$ is a known function. The elements Y, Z and F are called the input nodes, output nodes and the processor function, respectively.

The functional network is defined as (X, U) , where X is a set of nodes and U is defined as:

$$U = \{(Y_i, F_i, Z_i) | i = 1, \dots, n\} \tag{7}$$

Where U is a set of units on X which satisfy the following condition: each node $X_i \in X$ needs to be an input or output node of at least one functional unit U . X can be divided in three subsets $\{X_{en}, X_{int}, X_{sal}\}$ where X_{int} can be an empty set giving that the network has one layer.

The model of the relative humidity considers the temperature in the dry bulb and in the humid bulb known as the unit model from which its structure is shown in Fig. 2. The mathematical for the unit model is given as:

$$z = f_3^{-1}(f_1(x) + f_2(y)) \tag{8}$$

The training of the function (8) is the same as to learn the functions $f_1(x), f_2(x)$, and $f_3(x)$. Then the learning is to approximate this functions from some known data as follows:

$$\{(x_{1j}, x_{2j}, x_{3j}) | x_{3j} = F(x_{1j}, x_{2j}); j = 1, \dots, n\} \tag{9}$$

The approximation using the unit model gives the following polynomial functions:

$$\begin{aligned} f_1(x) &= 0.950306 + 0.0496945x \\ f_2(x) &= 1.04978 - 0.0497837x \\ f_3(x) &= 0.989221 + 0.0107788x \end{aligned} \tag{10}$$

Once the approximation functions are obtained, they are rewritten as follows:

$$\begin{aligned} z &= f_3^{-1}(f_1(x) + f_2(y)) \\ z &= \frac{f_1(x) + f_2(x) - 0.989221}{0.0107788} \\ z &= 92.7746(1.01087 + 0.0496945x_1 - 0.0497837x_2) \end{aligned} \tag{11}$$

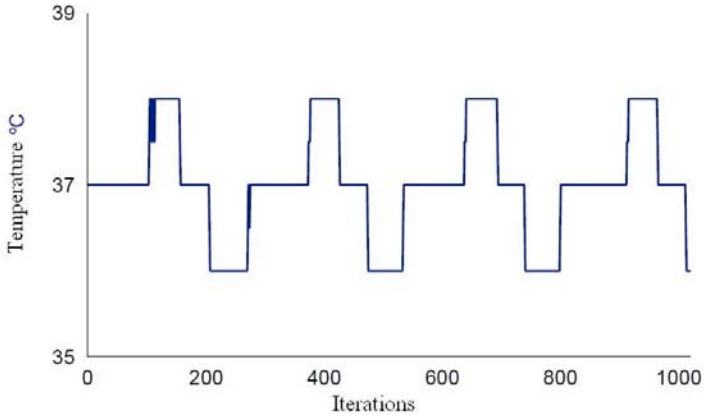


Fig. 3. Temperature for the control ON-OFF

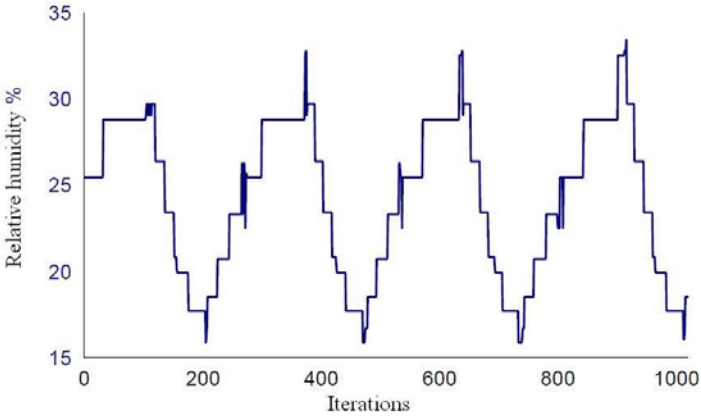


Fig. 4. Relative humidity for the control ON-OFF

Where x_1 is the temperature of the humid bulb, x_2 is the temperature of the dry bulb and z is the relative humidity. Finally, the approximation of the relative humidity can be considered as the substitution of the temperatures of the humid and the dry bulbs.

6 Simulations

In this paper, the tuning methods of Siegler-Nichols are used to get the gains for the controller PI , this gains are in the Table 1.

Table 1. Gains for the controller

	K_p	K_d	K_i
PI	21.83	0	7.35

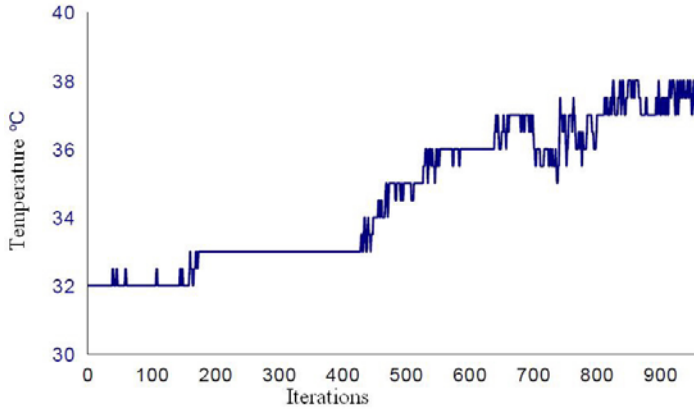


Fig. 5. Temperature for the control PI

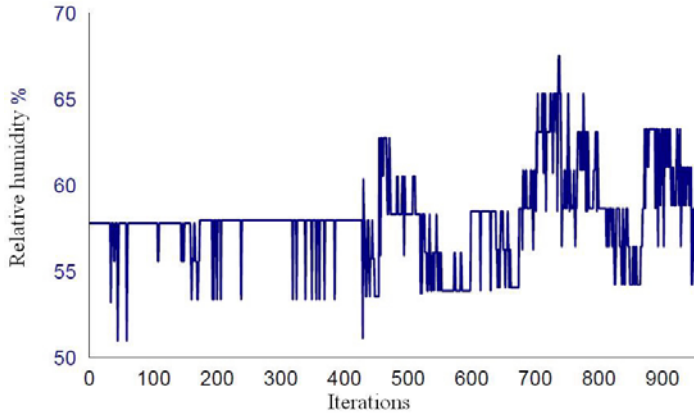


Fig. 6. Relative humidity for the control PI

In Table 1 it is shown the proportional constant K_p , and the integral constant K_i for the proportional integral controller (*PI*).

The controller *ON – OFF* is actually used in the market. The conditions for the proof of one controller *ON – OFF* are used as the bounds of the behavior of the controllers:

- Lower bound of $36^{\circ}C$
- Upper bound of $38^{\circ}C$
- The inside temperature of the incubator is bigger than temperature of the environment

The Fig. 3 shows the results of the temperature with an *ON – OFF* controller. The relative humidity obtained with the *ON – OFF* controller is shown in Fig. 4. The relative humidity is variant and it is smaller than the necessary to have

a right process of the incubator, i.e., the relative humidity for a right process of the incubator is 40% and the upper relative humidity given by the *ON – OFF* controller is 30%. That is way it is necessary to consider another controller.

For the *PI* controller it is considered the following:

- The reference is $36^{\circ}C$
- The inside temperature of the incubator is bigger than temperature of the environment

The Fig. 5 shows the output of the system considered as the inside temperature of the incubator for the *PI* controller which is acceptable. The Fig. 6 shows the results of the relative humidity for the *PI* controller being 60% which is over the necessary to get a right process of the incubator 40%.

7 Conclusions

In this paper a functional network to approximate the relative humidity behavior in a bird incubator was presented, also four kind of controller were compared giving that the proportional integral *PI* gave good behavior because they provided a very near temperature to the reference inside temperature of the incubator being of $36^{\circ}C$ and giving a relative humidity over the least necessary to get a right process of the incubator being 40%. The *ON – OFF* controller did not give a good behavior because it did not give a relative humidity over the least necessary to get a right process of the incubator. As a future research the functional network will be compared with some neural networks to get the best of them and the controller *PI* will be compared with other kind of classic controllers as are the *P* called proportional or the *PID* called proportional, integral derivative.

Acknowledgment. The authors are thankful with the editor to invite them to be part of the committee.

References

1. Chen, C.T.: Linear Systems: Theory and Design, 3rd edn. Oxford University Press, Oxford (1999)
2. Castillo, E., Gutierrez, J.M., Cobo, A.: Some Learning methods in functional networks. Computer-Aided Civil and Infrastructure Engineering (2000)
3. Hocking, L.M.: Optimal Control, An Introduction to the Theory with Applications, 2nd edn. Clarenton Press, Oxford (1997)
4. Khalil, H.K.: Nonlinear Systems, 3rd edn. Prentice-Hall, Upper Saddle River (1996)
5. Kirk, D.E.: Optimal Control Theory: An introduction. Prentice-Hall, Englewood Cliffs (1970)
6. Lacruz, B., Perez, A.: Functional networks for classification and regression problems. In: International Conference on Mathematical and Statistical Modeling (2006)
7. Ogata, K.: Ingenieria de Control Moderno, 3rd edn. Prentice-Hall, Englewood Cliffs (1998)

8. Rugh, J.W.: Linear System Theory, 2nd edn. Prentice-Hall, Upper saddle River (1996)
9. Sage, A.P., White, C.C.: Optimum System Control, 2nd edn. Prentice-Hall, NJ (1977)
10. Sastry, S.: Nonlinear Systems: Analysis, Stability and Control. Springer, NJ (1999)

A Novel Automatic Method on Diagnosing Movement Disorders

Masood Banaie¹, Mohammad Mikaili², and Mohammad Pooyan²

¹ Laboratory of Biomedical Engineering, Department of Biomedical Engineering,
Faculty of Engineering, Shahed University, Tehran, Iran
Tel.: (+98) 935 2128208; Fax: (+98) 21 51212020

Banaie@Gmail.com

² Department of Biomedical Engineering, Faculty of Engineering, Shahed University,
Tehran, Iran

Abstract. Diagnosing different diseases with some similarities (e.g. in symptoms) using an automated approach and a simple and available feature may help physicians to focus on the correct disease and its treatment and to avoid wasting time on diagnosis, Especially for diseases that the treatment time is critical to control and cure. In this study, we try to develop a new automated approach for classifying (diagnosing) locomotive patients using features that may be extracted from their gait signal. We selected 3 groups of patients: patients with Huntington's disease, Parkinson's disease and Amyotrophic Lateral Sclerosis- and a group of healthy control subjects. Examining different available classifiers on all proposed features, we have introduced a novel feature with acceptably low error rate using quadratic Bayes classifier.

1 Introduction

It is obvious that diagnosing different disorders in a correct and fast way is highly important in prescribing the correct treatment for the disease. Moreover, identifying the disease in a more simplified way will help physicians in decreasing the cost and time of the diagnosis process. Diagnosis time is an important factor in the treatment. In such a situation that we may need medical imaging or genetic tests to detect the disease, diagnosing movement diseases that have a similar cause through their effect on the movements of the patient will be precious. Gait signal may be a good factor for discriminating movement disorders that is caused by malfunctioning of some brain parts. It can also be used for validation of models that are introduced for the diseases. [1, 2]

In this study, we will evaluate different classifiers on the features that will be extracted from gait signal of patients with three different movement disorders including Huntington's Disease (HD), Parkinson's Disease (PD), and Amyotrophic Lateral Sclerosis (ALS). These diseases are selected as they have similar causes and all are neurodegenerative diseases. Control subjects were included in the study to present the healthy state.

HD is a genetic neurological disorder characterized by abnormal body movements called chorea and a lack of coordination. Because some of its symptoms are similar to

other neurological disorders, such as Alzheimer's disease, it has been extensively studied by different researchers. If the symptoms are noticeable before a person is twenty, then their condition is called Juvenile HD. As there is currently no proven cure, symptoms are controlled with various medications and supportive services. Complications caused by its symptoms reduce life expectancy. [2, 3, 4]

PD is also a degenerative disorder of the central nervous system that often impairs the sufferer's motor skills and speech, as well as other functions. It is characterized by muscle rigidity, tremor, a slowing of physical movement (bradykinesia) and, in extreme cases, a loss of physical movement (akinesia). The primary symptoms are due to decreased stimulation of the motor cortex by the basal ganglia. PD is both chronic and progressive. [1, 3, 4]

ALS is a progressive, usually fatal, neurodegenerative disease caused by the degeneration of motor neurons, the nerve cells in the central nervous system that control voluntary muscle movement. As a motor neuron disease, the disorder causes muscle weakness and atrophy throughout the body as both the upper and lower motor neurons degenerate, ceasing to send messages to muscles. [3, 4]

2 Materials and Methods

We have used a database from Physionet [5]. The database includes data recorded from 15 patients suffering PD, 20 of HD, 13 of ALS and 16 healthy (control) subjects. The raw data were obtained using force-sensitive resistors, with the output roughly proportional to the force under the foot. Each record included two signals recorded from each foot of the subject. [5]

Recordings were first analyzed visually and we removed 2 records (one of PD and one of HD records); because they were erroneously recorded in a limited period of time that may be caused by sensor malfunctioning. Before preparing our test and training datasets, we removed linear trends of the data to eliminate their offset. Then, we tried different features that seemed to be useful on classification of the data. First, we tried time-domain features such as zero-crossing, mean, and standard deviation of stride-to-stride time. We used a toolbox written for MATLAB named PRTools. This toolbox consists of a complete and useful set of functions for pattern recognition and includes most of the well known classification algorithms [6]. We converted the mentioned dataset to PRTools standard format. These data was applied to 18 of well known classifiers listed in table 2 and unacceptably high error rates were achieved (more than 50% in most cases) Using other features such as coefficients of a second order ARX model and fractal dimension of the signal did not make the results much better.

Looking for a more efficient feature, we have focused on a set of frequency-domain features (i.e. band powers). There are a couple of methods to calculate band powers including FFT and DCT. FFT (Fast Fourier Transform) was taken from the signal and its power in specific frequency bands were calculated as features. The error for this trial was similar to using the ARX coefficients. Even using a mixture of time features and frequency domain features extracted from FFT did not make the classification results reasonably better.

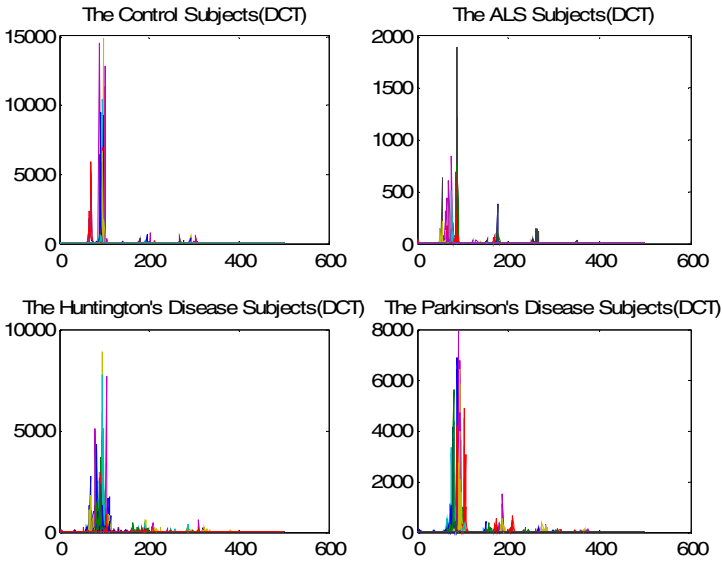


Fig. 1. DCT for the four classes (3 disease classes & the control healthy subjects)

Finally, we have proposed our novel method on extracting features using DCT (Discrete Cosine Transform). Because the gait signal is not stationary, we thought that DCT may result in better error rates. So, we used DCT instead of FFT to extract some features, calculating the trapped zone of DCT curve in special bands. We have taken DCT of all the signals and compared them visually. The formula for calculating discrete cosine transform of a signal is:

$$y(k) = w(k) \sum_{n=1}^N x(n) \cos \frac{\pi(2n-1)(k-1)}{2N}, \quad k = 1, \dots, N$$

Where

$$w(k) = \begin{cases} \frac{1}{\sqrt{N}} & k = 1 \\ \sqrt{\frac{2}{N}} & 2 \leq k \leq N \end{cases}$$

N is the length of x, and x and y are the same size. The time series are indexed from n = 1 and k = 1 instead of the usual n = 0 and k = 0 because MATLAB® vectors run from 1 to N instead of 0 to N- 1.

The DCT plots of the four classes are shown on figure 1. Defining special bands as shown in table1, we proposed that the signals could be fairly classified by calculating the trapped zone under the curve of each band as a feature.

Table 1. Sample bands for calculating trapped zone under the curve

Band No.	1	2	3	4	5
Sample Bands	1-50	60-74	75-90	93-105	110-150

We divided the dataset to train & test sets randomly. However, we have tried to select our test set among those recordings which had been recognized as the most severe cases by a physician. So we have selected 23 of 124 recordings as test and remaining as training set. Afterwards, we trained 18 most popular classifiers that were defined on PRTools. Considering this set of features made the error rates acceptably low for some of classifiers-especially for quadratic bayes classifier (second in the table2). The results including total error and the error that was made on diagnosing each class is shown in Table 2. For detailed description about each of the classifiers shown in Table 2, you can refer to [6].

Table 2. The results for different classifiers (Classification accuracy of each class and the total)

Classes:	Control	Huntington	Parkinson	ALS	Total Accuracy (%)
1 : Linear Bayes Normal Classifier	100	71.429	40	50	65.217
2 : Quadratic Bayes Normal Classifier	100	71.429	80	100	86.957
3 : Parzen Classifier	100	57.143	60	16.667	56.522
4 : BP Neural Classifier	80	71.429	0	83.333	60.87
5 : Karhunen-Loeve Mapping	0	0	0	0	0
6 : K-NN Classifier	100	42.857	60	66.667	65.217
7 : KL Bayes-Normal-1*	100	71.429	40	50	65.217
8 : Logistic Classifier	20	28.571	40	83.333	43.478

Table 2. (continued)

9 : Fisher	80	71.429	40	50	60.87
10 : Nearest Mean	100	0	60	33.333	43.478
11 : Scaled Nearest Mean	100	57.143	20	33.333	52.174
12 : Linear Perceptron Classifier	40	42.857	60	100	60.87
13 : Polynomial classifier	80	71.429	40	50	60.87
14 : Uncorrelated normal based quadratic Bayes classifier	100	42.857	40	100	69.565
15 : Mixture of Gaussian classifier	0	71.429	0	100	47.826
16: Decision Tree	100	85.714	40	83.333	78.261
17 : Naïve Bayes	100	57.143	40	33.333	56.522
18 : Levenberg- Marquardt trained feed- forward neural net classifier	40	71.429	40	83.333	60.87

* Linear classifier built on the KL expansion of the common covariance matrix.

3 Discussions

An automated method for classifying movement disorders prior to further assessment and diagnosis of the severity of diseases may help physicians to focus on the correct disease and to make better decisions. So a vast majority of research work in this area is

now directed toward finding automatic or semiautomatic diagnosis methods [7, 8, 9]. This may result in faster and more effective treatment of the disorders. In this study, we have tried to introduce an automated approach to diagnose three similar movement disorders through classifying the proposed features that are extracted from the gait signal of a patient. We proposed a feature set based on DCT of patients' gait signal and showed that using different classifiers we can get reasonable error rates. According to Table 2, we conclude that using a quadratic bayes classifier, we can classify the patients in the corresponding disease groups. Moreover, we proposed a novel feature that needs less computation for classifier to be trained and tested. Most of studies that focus on feature extraction using DCT only use 2D-DCT and is applied to images [10, 11, 12, 13]. However, it is not commonly used for signals like as gait that is one dimensional. We think that DCT may improve feature extraction on non-stationary signals (such as gait signal). The trapped zone under the DCT curve in special bands is introduced to have a high capability to be used in similar studies. This approach has the advantage that uses fewer features - only the trapped zone under the curve of the nine distinct bands of the discrete cosine transform (DCT) of the gait signal - to classify the disease of the subjects in a correct manner. This can be used to prepare an online approach to diagnose the disease's in a faster and more accurate way.

4 Future Considerations

In this study, we have focused on introducing a new automated approach on classifying the movement disorders in a correct manner. However, the design and implementing the classifier clinically may be considered in future.

Moreover, we had a limited access to clinical recordings needed for the train and test process. So, we could not get more accuracy. It was because the database we have used included only a limited number of subjects. Thus, training the classifier using more recordings might result in better results.

References

- [1] Haeri, M., Sarbaz, Y., Gharibzadeh, S.: Modeling the Parkinson's tremor and its treatments. *Journal of Theoretical Biology* 236(3), 311–322 (2005)
- [2] Banaie, M., Sarbaz, Y., Gharibzadeh, S., Towhidkhah, F.: Huntington's disease: Modeling the gait disorder and proposing novel treatments. *Journal of Theoretical Biology* (2008), doi:10.1016/j.jtbi.2008.05.023
- [3] Kandel, E.R., Schwartz, J.H., Jessell, T.M.: *Principles of Neural Science*, 4th edn. McGraw-Hill, New York (2000)
- [4] <http://en.wikipedia.org>
- [5] <http://www.physionet.org>
- [6] Duin, R.P.W., Juszczak, P., Paclik, P., Pekalska, E., de Ridder, D., Tax, D.M.J.: *PRTTools4. A Matlab Toolbox for Pattern Recognition*, Delft University of Technology (2004)
- [7] Jeon, B.-J., Ko, I.-Y.: Ontology-based Semi-automatic Construction of Bayesian Network Models for Diagnosing Diseases in E-health Applications. In: *Frontiers in the Convergence of Bioscience and Information Technologies* (2007)

- [8] Ross, N.E., Pritchard, C.J., Rubin, D.M., Duse, A.G.: Automated image processing method for the diagnosis and classification of malaria on thin blood smears. *Med. Biol. Eng. Comput.* 44, 427–436 (2006)
- [9] Hiroshi, K., Hisanori, T., Takanori, K., Koki, K.: Automatic Diagnosis of Mental Disease by Face Image Processing. *Nippon Kikai Gakkai Robotikusu, Mekatoronikusu Koenkai Koen Ronbunshu*, vol. 2000(Pt.4), page. 2P1.12.010(1)-2P1.12.010(2) (2000)
- [10] Kim, H.-J., Kim, W.-Y.: Computerized bone age assessment using DCT and LDA. In: Gagalowicz, A., Philips, W. (eds.) *MIRAGE 2007. LNCS*, vol. 4418, pp. 440–448. Springer, Heidelberg (2007)
- [11] Tachaphetpiboon, S., Amornraksa, T.: Fingerprint Features Extraction Using Curve-scanned DCT Coefficients. In: *Proceedings of Asia-Pacific Conference on Communications (2007)*
- [12] Yin, H., Fu, P., Qiao, J.: Face recognition based on DCT and 2DLDA. In: *Second International Conference on Innovative Computing, Information and Control, ICICIC 2007, September 5-7, p. 581 (2007)*
- [13] Kharat, G.U., Dudul, S.V.: Neural Network Classifier for Human Emotion Recognition from Facial Expressions Using Discrete Cosine Transform. In: *First International Conference on Emerging Trends in Engineering and Technology (2008)*

The Fault Diagnosis of Electric Railway Traction Substation with Model-Based Diagnosis of Integration of FDI and DX Approaches

Zhigang Liu, Zhiwei Han, and Wei Zhong

School of Elec. Eng., Southwest Jiaotong University, Chengdu, Sichuan, 610031, China
liuzg_cd@126.com

Abstract. Most diagnosis systems used in electric railway substation are expert systems based on experiences, which can't diagnose the faults beyond experiences and are very difficult for system transplanting and maintaining. In the paper, a method of model-based diagnosis is applied to diagnose the electric devices in railway substation. The system model knowledge can be obtained at the design phase of traction substations. Through the introduction of the minimal conflict candidate, the integration of FDI and DX is adopted for the main electrical devices diagnosis of traction substation. Finally, an example of actual AT traction substation is presented to verify the feasibility and validity of this method.

Keywords: Traction Substation, Fault Diagnosis, Model-based Diagnosis, Integration of FDI and DX.

1 Introduction

The traction power system is one of most important components in electric railway. The traction substation plays an important role as the power supply interface between general power system and electric railway system. In order to ensure the safe operation of traction power system, there are many different monitoring and controlling equipments for the electrical devices. In addition, some fault diagnosis systems are adopted as the assistant means for the device fault diagnosis in traction substation. The assistant systems are expert system based on experience in general. The expert system for relay protection diagnosis and breaker fault diagnosis is proposed based on monitoring and protection integrated and automated system in [1]. Some expert systems based experience knowledge is used to analyze and adjust the faults of electrical devices in traction substation [2-6]. To obtain the experience knowledge of diagnosed objects is a long-term course, although the fault diagnosis systems based on expert knowledge have been researched widely. The actual developing time for expert system is very long, which results into loss of synchronization between expert system and device update. Above all, the expert system based on experience knowledge can't correctly diagnose the faults beyond experiences. In addition, the system's transplanting and maintaining are very difficult.

To avoid the disadvantages of expert system, some scholars proposed the fault diagnosis system based on model-based diagnosis [7-11]. The main idea is illustrated in Fig. 1 below.

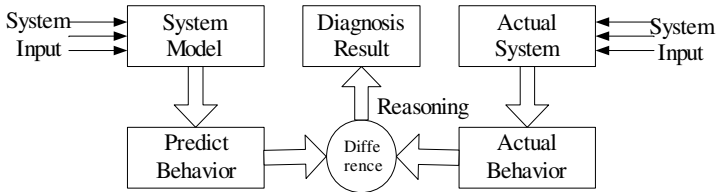


Fig. 1. The basic idea of model-based diagnosis

From Fig. 1 above, the knowledge of system model can be obtained at the design phase, which results into that the development of diagnosis system based on model-based diagnosis and system design can be processed at the same time. In addition, the diagnosis system based on model-based diagnosis can diagnose different faults with system design principle only. The faults that can't be predicted with experience can be diagnosed with the method. Recently, the research results about model-based diagnosis can be found in [8-10]. Furthermore, the actual diagnosis system with the method can developed and applied in actual engineering diagnosis system [12-13].

In general, the evolution of model-based diagnosis includes two communities separately work in parallel [14-15]. One is FDI (Fault Detection and Isolation) based on system analytical from automatic control field, the other is DX (Diagnostic) based on artificial intelligence from fault diagnosis field. In [14-15], the linking of the concepts and assumptions that underlie the FDI analytical redundancy approach and the DX consistency based logical approach is analyzed, and the potential minimal conflicts is defined. If the supposition condition is same, the FDI and DX are consistent. Namely, the same diagnosis can be obtained with them. Based on the conclusions in [14-15], the integration of FDI and DX approaches is adopted in the fault diagnosis of heat exchanger system [16-18].

In the paper, the idea of model-based diagnosis is applied in the fault diagnosis of traction substation. Through the introduction of basic theory of model-based diagnosis in DX field, the concept of potential minimal conflicts in [14-15] is adopted. The application feasibility and validity of integration of FDI and DX in traction substation fault diagnosis is analyzed and discussed. Finally, an actual AT traction substation is validated for the effect of model-based diagnosis method.

2 The Basic Theory of Model-Based Diagnosis

(1) The basic concept of model-based diagnosis

Reiter described the theory of model-based diagnosis with first-order logic language in [9]. Subsequently, he developed and improved some theories in [11]. Some concepts used in the paper are introduced below.

Definition 1: A system is a triple (SD, OBS, COMP) where: SD, the system description, is a set of first-order sentences. OBS, a set of observations, is a set of first-order sentences. COMP, the system component, is a finite set of constant.

Definition 2: Given two sets of component C_p and C_n , define $D(C_p, C_n)$ to be the conjunction:

$$[\bigwedge_{c \in C_p} AB(c)] \wedge [\bigwedge_{c \in C_n} \neg AB(c)] \tag{1}$$

A diagnosis is a sentence describing one possible state of the system, where this state is an assignment of the status normal or abnormal to each system.

Definition 3: A conflict set of (SD, OBS, COMP) is set of components $\{c_1, \dots, c_k\} \subseteq COMP$, which makes $SD \cup OBS \cup \{\neg AB(c_1), \dots, \neg AB(c_k)\}$ not been satisfied.

Definition 4: A diagnosis for (SD, OBS, COMP) is a minimal set $\Delta \subseteq COMP$ such that $SD \cup OBS \cup \{AB(c) | c \in \Delta\} \cup \{\neg AB(c) | c \in COMP - \Delta\}$ is consistent.

Definition 5: Suppose C is a collection of sets. A hitting set for C is a set $H \subseteq \bigcup_{S \in C} S$ such that $H \cap S \neq \emptyset$ for each $S \in C$. A hitting set for C minimal iff no proper subset of it is a hitting set for C .

Theorem 1: $\Delta \subseteq COMP$ is a diagnosis for (SD, OBS, COMP) iff Δ is a minimal hitting set for the collection of conflict sets for (SD, OBS, COMP).

(2) Conflicts versus analytical redundancy relations [14].

Definition 6: An ARR (analytical redundancy relations) is a constraint deduced from the system model which contains only observed variables, and which can therefore be evaluated from any OBS. It is noted $r=0$, where is called the residual of the ARR.

3 The Diagnosis Method of Traction Substation

Considering the location fixation of current and voltage transformers in traction substation, the model-based diagnosis of integration of FDI and DX is adopted for the fault diagnosis of main electrical devices. The main steps are listed below.

(1)Obtaining all analytical redundancy relations off-line and correlative minimal conflict candidate set of electrical main wiring in traction substation.

(2)Checking and adjusting all analytical redundancy relations of status message through current and voltage transformers in traction substation. If the analytical redundancy relations are not satisfied, a minimal conflict set should be chosen from minimal conflict candidate sets.

(3)After obtaining all minimal conflict sets, the hitting set of minimal conflict set can be solved with the method in DX field. Then, all minimal candidate diagnosis can be obtained.

(4) The minimal candidate diagnosis can be chosen through FDI field, and the final diagnosis conclusion is drawn.

The minimal conflict set can be adopted in the second step. The obtaining of minimal conflict set based on model-based diagnosis includes the obtaining of minimal candidate conflict set off-line and the affirming of minimal conflict set.

Based on the distributing conditions of observation for the traction substation system, all analytical redundancy relations and minimal support conditions can be established. The minimal support condition means minimal candidate conflict set. Then, when the diagnosis is being on-line, the observations of traction substation are introduced into analytical redundancy relations. If they can't be satisfied, the corresponding the minimal candidate conflict set is true minimal conflict set.

4 Traction Substation Modeling

(1) System Description

We take the AT (Autotransformer) traction substation in Hefei-Nanjing Passenger Link of China Railway as an example. The AT supply and V/X traction transformer wiring are adopted in traction substation system. The capability of traction transformer is $2 \times (25 + 20) \text{MVA}$. The two independent incoming lines and stand-by 220kV power supply are used, and the compensation device is not equipped in the traction substation. The main electric diagram is illustrated in Fig. 2 below.

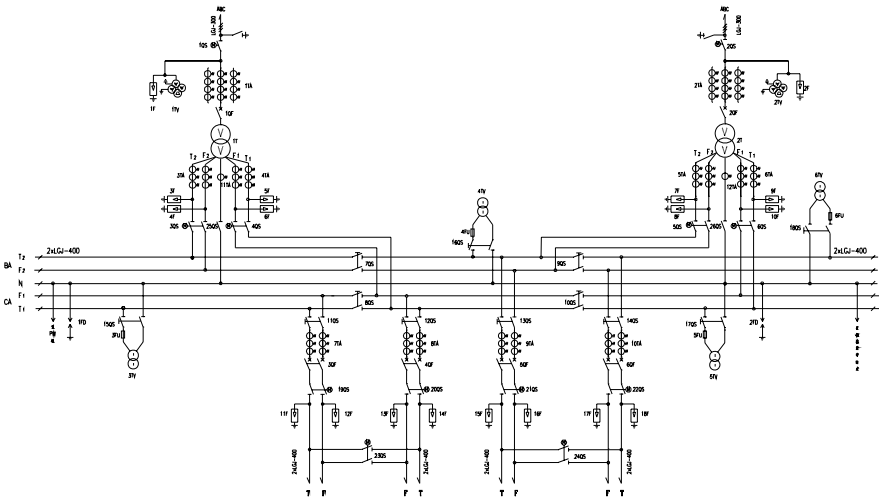


Fig. 2. Main electric diagram of AT traction substation in Hefei-Nanjing Passenger Link

(2) System Element

The main and standby operation mode of the transformer in traction substation is adopted. Namely, when 1T is used, 2T will be shut off through the breaker and disconnecting link. For the sake of simplifying problem, the correlative elements of 2T will not be considered during the course of analyzing actual example below. In

addition, the leads of each feeder in substation will not be considered, too. Based on the simplification of system, some system elements can be abstracted from Fig. 2.

Connecting Lines: L1_A, L1_B, L1_C (three-phase connecting lines from the power supply side to traction transformer 1T, including the breakers and the disconnectors).

Connecting Lines: L3_F1, L3_F2, L3_N, L3_T1, L3_T2 (all connecting lines from the traction side of traction transformer 1T to bus-bar side of substation, including the disconnectors).

Single-phase Transformers: T1_T1F1, T1_T2F2 (two independent single-phase transformers of traction transformer 1T)

Bus-bar: B_T1, B_F1, B_T2, B_F2, B_N (all bus-bars in the substation, including the disconnectors).

Rail and Ground: GND

(3) Element Parameters

The parameters and their value of T1_T1F1 are listed below:

$_0101_k=220/27.5,$
 $_0101_Zs21=j3.176 (\Omega),$
 $_0101_Zs31=j3.176 (\Omega),$
 $_0101_Zs23_1=j12.705 (\Omega),$
 $_0101_Rt=j7.0479 (\Omega),$
 $_0101_Ym=4.959e-7+j25.826e-7 (\Omega).$

The parameters and their value of T1_T2F2 are listed below:

$_0102_k=220/27.5,$
 $_0102_Zs21=j3.970 (\Omega),$
 $_0102_Zs31=j3.970 (\Omega),$
 $_0102_Zs23_1=j15.881 (\Omega),$
 $_0102_Rt=9.317 (\Omega),$
 $_0102_Ym=4.339e-7+j20.661e-7 (\Omega).$

(4) System Observation

The observations of current and voltage transformers for the corresponding systems in traction substation are listed below:

Current observations: I_1TA_A, I_1TA_B, I_1TA_C, I_3TA_F1, I_3TA_T1, I_11TA, I_4TA_F2, I_4TA_T2, I_7TA_F, I_7TA_T, I_8TA_F, I_8TA_T, I_9TA_F, I_9TA_T, I_10TA_F, I_10TA_T.

Voltage observations: V_1TV_A, V_1TV_B, V_1TV_C, V_3TV, V_4TV.

5 Experiment Results and Analysis

The analytical redundancy relations and the minimal conflict candidate sets of the substation can be obtained through the search off-line. The minimal conflict candidate sets are listed below:

MinCSC1 = [I_1TA_A+I_1TA_C+I_1TA_B=0, {L1_A, L1_B, L1_C, T1_T1F1, T1_T2F2}];

$$\begin{aligned}
 \text{MinCSC2} &= [-I_{3TA_F1}+I_{7TA_F}+I_{8TA_F}=0, \{B_F1\}]; \\
 \text{MinCSC3} &= [-I_{3TA_T1}+I_{7TA_T}+I_{8TA_T}=0, \{B_T1\}]; \\
 \text{MinCSC4} &= [-I_{4TA_F2}+I_{9TA_F}+I_{10TA_F}=0, \{B_F2\}]; \\
 \text{MinCSC5} &= [-I_{4TA_T2}+I_{9TA_T}+I_{10TA_T}=0, \{B_T2\}]; \\
 \text{MinCSC6} &= [I_{3TA_F1}+I_{3TA_T1}+I_{4TA_F2}+I_{4TA_T2}+I_{11TA}=0, \\
 &\quad \{L3_F1, L3_F2, L3_N, L3_T1, L3_T2, T1_T1F1, T1_T2F2\}]; \\
 \text{MinCSC7} &= [-_0101_Ym*_0101_k*V_{1TV_C}+_0101_Ym*_0101_k*V_{1TV_A}+ \\
 &\quad _0101_k*I_{1TA_C}-I_{3TA_T1}+I_{3TA_F1}=0, \{L1_A, L1_C, L3_F1, \\
 &\quad L3_T1, T1_T1F1\}]; \\
 \text{MinCSC8} &= [-_0102_Ym*_0102_k*V_{1TV_B}+_0102_Ym*_0102_k*V_{1TV_A}+ \\
 &\quad _0102_k*I_{1TA_B}-I_{4TA_T2}+I_{4TA_F2}, \{L1_A, L1_B, L3_F2, \\
 &\quad L3_T2, T1_T2F2\}]; \\
 \text{MinCSC9} &= [-2*V_{3TV}*_0101_Ym*_0101_k-2*_0101_Zs21*_0101_Ym* \\
 &\quad _0101_k*I_{3TA_T1}-_0101_Zs21*_0101_Ym*_0101_k*I_{3TA_F1}- \\
 &\quad 2*I_{3TA_T1}+_0101_Ym*_0101_k*I_{3TA_F1}*_0101_Zs23_1+ \\
 &\quad 2*_0101_Ym*_0101_Rt*I_{3TA_F1}-2*_0101_Ym*_0101_Rt* \\
 &\quad I_{3TA_T1}+2*_0101_k*I_{1TA_C}+2*I_{3TA_F1}-_0101_Ym* \\
 &\quad _0101_k*I_{3TA_F1}*_0101_Zs31=0, \{B_N, B_T1, L1_C, L3_F1, \\
 &\quad L3_N, L3_T1, T1_T1F1\}]; \\
 \text{MinCSC10} &= [-2*V_{4TV}*_0102_Ym*_0102_k-2*_0102_Zs21*_0102_Ym* \\
 &\quad _0102_k*I_{4TA_T2}-_0102_Zs21*_0102_Ym*_0102_k* \\
 &\quad I_{4TA_F2}-2*I_{4TA_T2}+_0102_Ym*_0102_k*I_{4TA_F2}* \\
 &\quad _0102_Zs23_1+2*_0102_Ym*_0102_Rt*I_{4TA_F2}-2*_0102_Ym* \\
 &\quad _0102_Rt*I_{4TA_T2}+2*_0102_k*I_{1TA_B}+2*I_{4TA_F2}- \\
 &\quad _0102_Ym*_0102_k*I_{4TA_F2}*_0102_Zs31=0, \{B_N, B_T2, L1_B, \\
 &\quad L3_F2, L3_N, L3_T2, T1_T2F2\}].
 \end{aligned}$$

Supposing the short circuit earthing of bus-bar B_F1 and connecting line L1_C, the measuring values of all current and voltage transformers in the traction substation are listed in Table 1.

Table 1. The measure value of all transformers during fault conditions in railway substation

Observations	Amplitude	Phase Angle	Observations	Amplitude	Phase Angle
V_1TV_A	1.2216e+005	-0.98°	V_3TV	1.3797e+004	178.62°
V_1TV_C	2.6006e+001	37.02°	V_4TV	1.3089e+004	-149.20°
V_1TV_B	1.2294e+005	-118.65°	I_3TA_F1	2.4688e+002	-85.67°
I_1TA_A	1.0871e+003	-59.62°	I_3TA_T1	2.4688e+002	94.33°
I_1TA_B	1.0318e+003	121.89°	I_4TA_F2	8.2464e+003	-58.11°
I_1TA_C	2.6040e+004	37.14°	I_4TA_T2	5.8559e+000	126.48°
I_7TA_F	1.1255e+002	-85.67°	I_9TA_F	2.9278e+000	-53.53°
I_7TA_T	1.1255e+002	94.33°	I_9TA_T	2.9279e+000	126.48°
I_8TA_F	1.3433e+002	-85.67°	I_10TA_F	2.9277e+000	-53.53°
I_8TA_T	1.3433e+002	94.33°	I_10TA_T	2.9280e+000	126.48°
I_11TA	8.2406e+003	121.88°			

If the measuring values are introduced in the restrictions of analytical redundancy relations of each minimal conflict candidate sets, the residual values of each analytical redundancy relation can be calculated in Table 2.

Let the relative error be 0.01, the minimal conflict sets can be chosen from the minimal conflict candidate sets whose relative residual value $r' > 0.01$ in Table 2.

- <B_F1>, <L1_A, L1_B, L1_C, T1_T1F1, T1_T2F2>
- <L1_A, L1_C, L3_F1, L3_T1, T1_T1F1>
- <B_N, B_T1, L1_C, L3_F1, L3_N, L3_T1, T1_T1F1>

Table 2. The residual values of analytical redundancy relationships

Minimal Conflict Candidate Set	Absolute Residual Value (r)	Maximum	Relative Residual Value (r')
MinCSC1 :	2.600e+004 ;	2.604e+004	0.998e+000
MinCSC2 :	1.000e-008 ;	2.469e+002	0.405e-010
MinCSC3 :	0.000 ;	2.469e+002	0.000
MinCSC4 :	8.240e+003 ;	8.246e+003	0.999e+000
MinCSC5 :	0.000 ;	5.856e+000	0.000
MinCSC6 :	9.703e-001 ;	8.246e+003	0.117e-003
MinCSC7 :	2.081e+005 ;	2.083e+005	0.998e+000
MinCSC8 :	5.755e+000 ;	8.254e+003	0.697e-003
MinCSC9 :	4.161e+005 ;	4.166e+005	0.998e+000
MinCSC10 :	5.847e+000 ;	1.650e+004	0.354e-003

The minimal hitting set can be calculated from the 4 minimal conflict sets, and 11 minimal candidate diagnoses can be obtained.

- [B_F2, L1_C], [B_F2, T1_T1F1], [B_N, B_F2, L1_A],
- [B_F2, B_T1, L1_A], [B_F2, L1_A, L3_F1], [B_F2, L1_A, L3_N],
- [B_F2, L1_A, L3_T1], [B_F2, L1_B, L3_F1], [B_F2, L1_B, L3_T1],
- [B_F2, L3_F1, T1_T2F2], [B_F2, L3_T1, T1_T2F2].

Thus it can be seen, there are much candidate diagnosis. In order to choose the final diagnosis from candidate diagnosis, some methods should be adopted. If the method of DX field is used, more observation message will be needed. If the method of FDI field is used, namely supposing $r > 0$, the elements in minimal support conditions are normal. Then, a normal element set can be obtained.

- {B_N, B_F1, B_T1, B_T2, L1_A, L1_B, L3_F1,
- L3_F2, L3_N, L3_T1, L3_T2, T1_T1F1, T1_T2F2}

If the elements in the normal element set are included in candidate diagnosis, the candidate diagnosis is not the final diagnosis. Based on the supposition, the 10 latter minimal candidate diagnoses can be excluded from the 11 minimal candidate diagnoses. The last candidate diagnosis [B_F1, L1_C] is just final diagnosis of the whole system.

In addition, if the method of FDI field is used to restrict the minimal conflict set, the 4 minimal conflict sets can be shrunk gradually. Namely, the considered normal elements can be obtained from the minimal conflict sets, and the less conflict sets can be calculated.

$\langle B_F1 \rangle, \langle L1_C \rangle, \langle L1_C \rangle, \langle L1_C \rangle$

Because the three latter minimal conflict sets are same, they can be combined into one set. The two less minimal conflict sets $\langle B_F1 \rangle, \langle L1_C \rangle$ can be obtained. There is only one minimal hitting set of them, so $[B_F1, L1_C]$ can be confirmed, which is just the final diagnosis conclusion: the short circuit earthing faults of bus-bar B_F1 and connecting line L1_C has occurred in the traction substation.

6 Discussion

From the actual example above, the faults of electrical devices in the traction substation can be diagnosed with the method of model-based diagnosis, which only need using the normal models of electrical devices. The model-based diagnosis of DX field needs obtaining the minimal conflict set firstly based on system observation message. It will cost much time to obtain the system minimal conflict set, if the course is operated on-line. Because the status monitoring values of electrical devices in traction substation are generally invariable, the minimal conflict candidate sets can be obtained off-line during the course of minimal conflict set recognition with the analytical redundancy relations from DX field. When a set of system observation message is given, the minimal conflict set can be obtained from the minimal conflict candidate that can't satisfy the analytical redundancy relations.

7 Conclusion

As a fault diagnosis method that can obtained system model knowledge during the course of design, the model-based diagnosis has been paid attention by the researcher gradually. In the paper, through the introduction of the concept of minimal conflict candidate, the model-based diagnosis with integration of FDI and DX is used for the fault diagnosis of electrical devices in electrical railway traction substation. The modeling and experiment of the method is processed, and the results show that this method is feasible and valide for the fault diagnosis of the actual traction substation.

Acknowledgments. The work was supported by New Century Excellent Talents in University and Sichuan Province Distinguished Scholars Fund (No. 07ZQ026-012).

References

1. Chen, W.J., Chen, X.C., Gao, S.B.: Study on Integrated Automation and Application of Expert System for Traction Substations. *Journal of the China Railway Society* 18, 6–10 (1996)
2. Guo, K.: A DFR Data Based Expert System for Traction Substation Fault Analysis. Southwest Jiaotong University Master Degree Thesis (2004)

3. Li, X.Q., Bao, L.D., Fan, J.S.: An Expert System for the Fault Diagnosis of Electrified Railway Substations. *Journal of Shi Jia Zhuang Railway Institute* 8, 65–69 (1995)
4. Xie, S.F., Li, Q.Z.: Application of Expert System Based on Mixing Reasoning in Traction Substation Fault Diagnosis. In: *Proc. the 2nd International Workshop on Autonomous Decentralized System*, pp. 229–232. IEEE Press, Chengdu (2002)
5. Liu, F.: A Visual Prolog Based Expert System for Traction Substation Fault Analysis. Southwest Jiaotong University Master Degree Thesis (2005)
6. Zhuang, H.M., Chen, X.C.: Expert System Design on Analysis of Relay Protection's Action at Substation. *Relay* 31, 41–42, 63 (2003)
7. Kleer, J.: *Local Methods for Localizing Faults in Electronic Circuits*. MIT Artificial Intelligence Laboratory (1976)
8. Davis, R.: Diagnostic Reasoning Based on Structure and Behavior. *Artificial intelligence* 24, 347–410 (1984)
9. Reiter, R.: A Theory of Diagnosis from First Principles. *Artificial Intelligence* 32, 57–95 (1987)
10. Kleer, J., Williams, B.C.: Diagnosing Multiple Faults. *Artificial Intelligence* 32, 97–130 (1987)
11. Kleer, J., Mackworth, A.K., Reiter, R.: Characterizing Diagnoses and Systems. *Artificial Intelligence* 56, 197–222 (1992)
12. Baoyin, H.: *The Application of Model-based Diagnosis Technologies in the Satellite*. Harbin Institute of Technology Doctoral Degree Dissertation (1999)
13. Zheng, W.: *The Fault Diagnosis Method Research of Liquid-propellant Rocket Engine Based Aualitative Model-based Diagnosis*. National University of Defense Technology Master Degree Thesis (2002)
14. Cordier, M.O., Dague, P., Lévy, F., et al.: Conflicts Versus Analytical Redundancy Relations: A Comparative Analysis of the Model Based Diagnosis Approach From the Artificial Intelligence and Automatic Control Perspectives. *IEEE Transactions on Systems, Man, and Cybernetics, Part B* 34, 2163–2177 (2004)
15. Cordier, M.O., Dague, P., et al.: A Comparative Analysis of AI and Control Theory Approaches to Model-based Diagnosis. In: *14th European Conference on Artificial Intelligence*, pp. 36–140 (2000)
16. Gómez-López, M.T., Ceballos, R., Gasca, R.M., Pozo, S.: Determination of possible minimal conflict sets using constraint databases technology and clustering. In: Lemaître, C., Reyes, C.A., González, J.A. (eds.) *IBERAMIA 2004*. LNCS, vol. 3315, pp. 942–952. Springer, Heidelberg (2004)
17. Gómez López, M.T., Ceballos Guerrero, R., Martínez Gasca, R., del Valle Sevilla, C.: Applying constraint databases in the determination of potential minimal conflicts to polynomial model-based diagnosis. In: Kuijpers, B., Revesz, P.Z. (eds.) *CDB 2004*. LNCS, vol. 3074, pp. 74–87. Springer, Heidelberg (2004)
18. Ceballos, R., Pozo, S., Del Valle, C., Gasca, R.M.: An Integration of FDI and DX Techniques for Determining the Minimal Diagnosis in an Automatic Way. In: Gelbukh, A., de Albornoz, Á., Terashima-Marín, H. (eds.) *MICAI 2005*. LNCS (LNAI), vol. 3789, pp. 1082–1092. Springer, Heidelberg (2005)

A Single-Hop Active Clustering Algorithm for Wireless Sensor Networks

Fengjun Shang

College of Computer Science and Technology
Chongqing University of Posts and Telecommunications, Chongqing 400065, China
shangfj@cqupt.edu.cn

Abstract. In this paper, a Single-Hop Active Clustering (SHAC) algorithm is proposed for wireless sensor networks. The core of SHAC has three parts. Firstly, when tentative clusterheads are selected, an average energy factor is introduced to balance the residual energy of the whole network nodes which improves the network energy efficiency. Secondly, a cost function is proposed to balance energy-efficient of each node. Finally, an active clustering algorithm is proposed for single-hop homogeneous networks. Through both theoretical analysis and numerical results, it is shown that SHAC prolongs the network lifetime significantly against the other clustering protocols such as LEACH-C and EECS. Under general instance, SHAC may prolong the lifetime by up to 50% against EECS.

Keywords: wireless sensor network, active cluster, cost function, homogeneous.

1 Introduction

The rapid developments and technological advances in MEMS(Micro Electromechanical System) and wireless communication, has made possible the development and deployment of large scale wireless sensor networks. Wireless sensor network consists of hundreds to several thousands of small sensor nodes scattered throughout an area of interest. The potential applications of sensor networks are highly varied, such as environmental monitoring, target tracking, and battlefield surveillance. Sensors in such a network are equipped with sensing, data processing and radio transmission units. Distinguished from traditional wireless networks, sensor networks are characterized by severe power, computation, and memory constraints. Due to the strict energy constraints, energy efficiency for extending network lifetime is one of the most important topics. Sensor nodes are likely to be battery powered, and it is often very difficult to change or recharge batteries for these nodes. Prolonging network lifetime for these nodes is a critical issue. Therefore, all aspects of the node, from hardware to the protocols, must be designed to be extremely energy efficient.

Wireless sensor networking is a broad research area, and many researchers have done research in the area of power efficiency to extend network lifetime. In order to achieve high energy efficiency and increase the network scalability, sensor nodes can be organized into clusters. The high density of the network may lead to multiple

adjacent sensors generating redundant sensed data, thus data aggregation can be used to eliminate the data redundancy and reduce the communication load [1]. Hierarchical protocols aim at clustering the nodes so that cluster heads can do some aggregation and reduction of data in order to save energy.

In this paper we assume that the sink is not energy limited (at least in comparison with the energy of other sensor nodes) and that the coordinates of the sink and the dimensions of the field are known. We also assume that the nodes are uniformly distributed over the field and they are not mobile. Under this model, we propose a new energy-efficient adaptive clustering algorithm. Our contributions have four parts. Firstly, an average energy-based threshold is introduced to produce tentative clusterheads so that our algorithm may prolong network lifetime. Secondly, an estimated average energy method is introduced to avoid additional communication between BS and clusterhead. Thirdly, a cost function is proposed to balance energy-efficient of each node. Last but not least, an active clustering algorithm is proposed in single-hop homogeneous network. Through both theoretical analysis and numerical results, it is shown that SHAC prolongs the network lifetime significantly against the other clustering protocols such as LEACH-C and EECS.

Owing to constraining the resource of sensor node, clustering algorithm aiming at ad hoc networking can not be used directly, especially, the energy of WSN is limited, so new clustering algorithm must be researched. LEACH (low-energy adaptive clustering hierarchy)^[1] is first proposed as clustering routing protocol in WSN. Its clustering idea is used in many clustering routing protocol, for example, TEEN(threshold sensitive energy efficient sensor network protocol)^[2], HEED(hybrid energy-efficient distributed clustering)^[3] etc.. At the same time, there are some independent designing clustering routing protocol, for example, ACE (algorithm for cluster establishment)^[4], LSCP(lightweight sensing and communication protocols)^[5] etc..

The paper is organized as follows. In Section 2, related work is discussed. Section 3, describes our proposed clustering routing algorithm. In section 4, simulation results are presented while Section 5 concludes the paper.

2 Related Works

LEACH is one of the most popular hierarchical routing algorithms for sensor networks. The idea is to form clusters based on the received signal strength and use local cluster heads as routers to the sink. This is shown to save energy since the transmissions will only be done by such cluster heads rather than all sensor nodes. All the data processing such as data fusion and aggregation are local to the cluster. Cluster heads change randomly over time in order to balance the energy dissipation of nodes. This decision is made by the node choosing a random number between 0 and 1.

In recent years, a number of modifications have been proposed for the LEACH algorithm, for example, EECS^[6], LEACH-B^[7] etc.. In EECS, in order to cluster, nodes in selecting clusterhead do not only the closest clusterhead but also the closest distant from clusterhead to BS to balance the load of network. But the EECS only balance energy in the area of clusterhead and it can not balance the energy in whole network.

In [8], it is proposed to select clusterhead according to the residual energy of node. The main disadvantage of this algorithm requires the energy information of all nodes of the network not to be distributed realize. SEP^[9] mainly aims at two-level heterogeneous network, that is, in this network, its initial energy has two kinds of level. DEEC algorithm^[10] aim at the general multi-level heterogeneous networks, But it can also adapted to operate in homogeneous sensor network. In [11], DCHS algorithm is proposed. In this algorithm, in order to extend the lifetime, a parameter $E_{n_current} / E_{n_max}$ is introduced. Furthermore, introducing a factor r_s is a further modification of the threshold equation so that this may improve the performance of algorithm.

The clustering algorithm for single-hop networks have little delay and been well suited for time critical applications. However, its energy consumption is much higher between BS and node. The clustering algorithm for multi-hop network is complex and difficult to implement. In this paper, we propose a single-hop clustering algorithm which prolongs the lifetime of network.

In order to conserve node energy and prolong lifetime of the network, the previous research have been mainly focused on balancing energy consumption among cluster members and they do not consider energy consumption among clusterheads. In this paper, we propose the SHAC algorithm for homogeneous and single-hop sensor network. According to the residual energy of node, SHAC algorithm selects tentative cluster-heads in order to improve the clustering idea of LEACH. At the same time, SHAC algorithm keeps the distributed characteristic of algorithm and it does not require location information of all nodes of the network.

3 SHAC Routing Algorithm

In this paper, a novel clustering idea is proposed called active clustering. Generally speaking, clusterheads are first selected based on the corresponding rule and then nodes are passive adding to that clusterhead, for example, LEACH selects them according to threshold, etc.. In our idea, nodes select actively clusterheads according to cost function so that it can balance energy well. Our idea includes several parts as follows.

In selecting tentative clusterheads phase, an average energy factor is introduced. Its aim is rational selecting clusterheads according to residual energy. The high residual energy is high probability selected clusterhead so that this may balance the whole network energy. According to above idea, a clustering algorithm is proposed based on the average energy of whole network. This algorithm is similar to LEACH-C^[8], but it avoids transmitting residual energy from nodes to BS. An estimation algorithm must be introduced so that this algorithm may avoid above energy consumption problem. After tentative clusterheads are selected, according to the cost function, the tentative clusterheads select final clusterheads according to prior, that is, the number of nodes adding to that clusterhead and then each tentative clusterhead knows final clusterhead. Lastly, the final clusterhead broadcasts information around nodes, because selecting clusterheads is minimal cost so that it may prolong the lifetime of network.

3.1 Network Model

Let us consider a sensor network consisting of N sensor nodes uniformly deployed over a vast field to continuously monitor the environment. We denote the i -th sensor by s_i and the corresponding sensor node set $Node = \{n_1, n_2, \dots, n_N\}$, where $|Node| = N$. We make some assumptions about the sensor nodes and the underlying network model:

- 1) There is a base-station (*i.e.*, data sink) located far away from the square sensing field. Sensors and the base-station are all stationary after deployment.
- 2) All nodes are homogeneous and have the same capabilities. Each node is assigned a unique identifier (ID).
- 3) Nodes have no location information.
- 4) All nodes are able to reach BS in one hop.
- 5) Nodes can use power control to vary the amount of transmission power which depends on the distance to the receiver.
- 6) Links are symmetric. A node can compute the approximate distance to another node based on the received signal strength, if the transmitting power is given^[12].

We use a simplified model shown in figure 1 for the radio hardware energy dissipation. Both the free space (d^2 power loss) and the multi-path fading (d^4 power loss) channel models are used in the model, depending on the distance between the transmitter and receiver. Transmission (E_{Tx}) and receiving costs (E_{Rx}) are calculated as follows[8]:

$$E_{Tx}(l, d) = \begin{cases} lE_{elec} + l\epsilon_{fs}d^2, & d < d_o \\ lE_{elec} + l\epsilon_{mp}d^4, & d > d_o \end{cases} \quad (1)$$

Where d is the distance between the transmitter and the receiver.

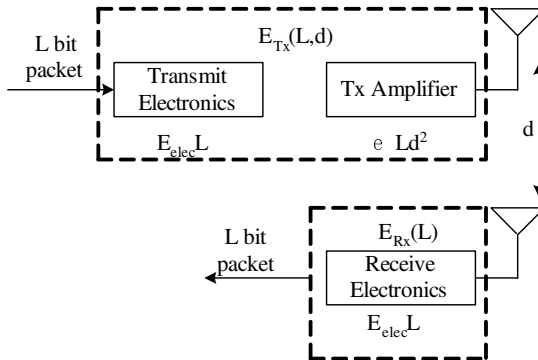


Fig. 1. Radio Energy Dissipation Model

To receive this message, the energy used by the radio can be expressed following:

$$E_{Rx}(l) = E_{elec}l \tag{2}$$

with l as the length of the message in bits, d as the distance between transmitter and receiver node. A sensor node also consumes E_{DA} ($nJ/bit/signal$) amount of energy for data aggregation. It is also assumed that the sensed information is highly correlated, thus the clusterhead can always aggregate the data gathered from its members into a single fixed length packet.

3.2 SHAC Algorithm

In the network deployment stage, the base-station broadcasts a “hello” message to all nodes at a certain power level. Using this approach, each node can compute the approximate distance to the base-station based on the received signal strength. It not only helps nodes to select the proper power level to communicate with the base-station, but also helps us to produce clusters of unequal size. Figure 2 gives an overview of the SHAC mechanism, where the anomalous polygon of unequal size represent our clusters of unequal size and the traffic among cluster heads illustrates our single-hop forwarding method.

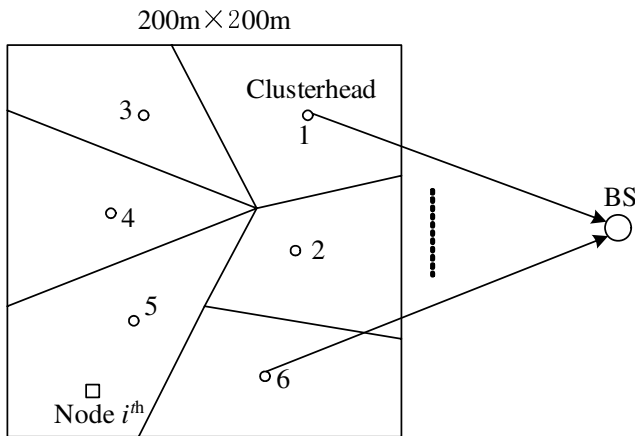


Fig. 2. An overview of the SHAC mechanism

From figure 2, The SHAC algorithm makes nodes cost maximum to have a lower chance of becoming a cluster-head than nodes cost minimum in order to reduce energy consumption, at the same time, in each area of clusterhead overlay, the highest residual energy is selected as final clusterhead. The process of SHAC algorithm is as follows. It first starts by selecting a tentative cluster-head. This decision is made by the node choosing a random number between 0 and 1. If random number is below the threshold, the node is selected as tentative cluster-head. Thus each tentative clusterhead receives information from node adding its clusterhead and calculates the number of node. Lastly, in selecting final clusterheads phase, each tentative clusterhead selects the

final clusterhead according to the prior so as to acquire final clusterheads. Once final clusterhead is selected, it broadcasts the information to the neighbouring nodes. In forming cluster phase, each node adds the selected cluster-head according to cost function and then each node returns the information to selected clusterhead. In data transmitting phase, cluster member transmits data to clusterhead according to TDMA slot and then clusterhead converges the data and transmits to BS. Once the above process is completed, the algorithm begins to prepare next round work.

3.3 Selecting Tentative Clusterhead

Selecting tentative cluster-head is the basis for creating clusters. When SHAC selects tentative clusterhead, this decision is made by the node choosing a random number between 0 and 1. If random number is below the threshold $T(n)$, n is number of nodes, the node is selected as tentative clusterhead. The clusterhead broadcasts itself to become tentative. Because LEACH does not consider residual energy and distance of nodes, an average energy factor for LEACH is introduced and make the node with the highest level of energy to be first tentative clusterhead. The key parameter is as follows.

$$Energy = \frac{E(i)_{residual}}{\overline{E(r)}} \quad (3)$$

where $E(i)_{residual}$ is the residual energy of the i -th node, $\overline{E_r}$ is the average energy of the node and r is the current round number. *Energy* factor is used to balance the whole network energy.

Definition. The number of rounds from first round to round which first node dies is called *lifetime*.

By above analysis, introducing *Energy* factor can balance the whole network energy and prolong *lifetime* of network. The new threshold formula is as follows.

$$T(n)_{new} = \begin{cases} \frac{p}{1 - p \times [r \bmod (\frac{1}{p})]} \times Energy, n \in G \\ 0, otherwise \end{cases} \quad (4)$$

where p is the desired percentage of cluster heads (e.g. $p=0.1$), r is the current round, and G is the set of nodes that have not been cluster-heads in the last $1/p$ rounds.

The election probability of nodes $s \in G$ to become cluster heads increases in each round in the same epoch and becomes equal to 1 in the last round of the epoch. Note that by round we define a time interval where all clusters members have to transmit to the cluster head once.

3.4 Active Selecting Clusterheads

Clustering a wireless sensor network means partitioning its nodes into groups, each one with a cluster head and some ordinary nodes as its members. The task of being a

cluster head is rotated among sensors in each data gathering round to distribute the energy consumption across the network. SHAC is a distributed cluster heads competitive algorithm, where cluster head is selected primarily based on the residual energy of each node.

Firstly, several tentative clusterheads are selected to compete for final cluster heads. Each node becomes a tentative cluster head with the same probability *Threshold* which is a predefined threshold. Secondly, each tentative clusterhead broadcasts a *COMPETE_HEAD_MSG*, which includes residual energy(*RE*), distance from BS(*DBS*), broadcast radius *R* and *ID* of that tentative clusterhead and each node adds the clusterhead according to the cost function $f(i,j)$. Thirdly, each tentative clusterhead broadcasts *RECEIVE_NODE_MSG*, which includes the number of nodes adding this clusterhead and each tentative clusterhead receives the information and selects the final clusterhead according to prior knowledge, that is, selects the clusterhead accoring to number of nodes adding to that clusterhead, we select them about eight clusterheads. Lastly, the final clusterheads broadcast *FINAL_HEAD_MSG*.

3.5 Balancing Cluster Member Energy

After cluster-heads are selected, the key problem is assign each node to particular cluster-head. It is important to balance energy consumption in area around the cluster-head, for example, node *i* shown in figure 2 can use clusterhead 5 or 6, which must be decided by the cost function. Intuitively, node *i* add clusterhead 5, because it is close to clusterhead, but, it is not beneficial to balance the network energy consumption. The proposed cost function $f(i, j)$ is as follows.

$$f(i, j) = w \times \frac{c(n_i, CH_j)}{E_i} + (1 - w) \times \frac{g(CH_j, BS)}{E_{CH_j}}$$

The condition is that cost function $f(i, j)$ is minimum, if node *i* uses to CH_j , where E_i is current energy of the *i*-th node, E_{CH_j} denotes current energy of the *j*-th clusterhead and

$$c(n_i, CH_j) = \frac{d^2(n_i, CH_j)}{d_{n_CH}^2}$$

$$g(n_i, CH_j) = \frac{d^4(CH_j, BS)}{d_{CH_max}^4}$$

where $d(CH_j, BS)$ denotes the distance from *j*-th clusterhead to BS, E_{CH_j} denotes residual energy of *j*-th clusterhead, $d_{CH_max} = \max\{d(CH_j, BS)\}$, $d(n_i, CH_j)$ denotes the distance from *i*-th node to *j*-th clusterhead, $d_{n_CH} = \max\{d(n_i, CH_j)\}$.

4 Simulations and Analysis

We select a scenario to simulate our algorithm using MATLAB and the parameter set is shown in Table 1.

Table 1. Simulation parameters

Parameter	Value
Network coverage	(0,0)~(200,200)
Base station location	(100,350)
N	1000
Initial energy	1J
E_{elec}	50nJ/bit
ϵ_{fs}	10pJ/bit/m ²
ϵ_{mp}	0.0013pJ/bit/m ⁴
d_o	87m
E_{DA}	5nJ/bit/signal
Data packet size	4000 bits

The key problem is how to select parameter w in SHAC algorithm, because w is weight value using balancing between clusterhead and cluster member. In this paper we select $w = 0.6$.

We compare between LEACH-C, EECS and SHAC. Figure 3 shows lifetime of network over the simulation time, where SHAC is the longest. SHAC prolongs the network lifetime significantly against the other clustering protocols such as LEACH-C and EECS. Under general instance, SHAC may prolong the lifetime at least 30%. Additionally, it may prolongs the lifetime by up to 50% against EECS. By observing the number of dead nodes, it can be seen that there are no dead nodes in 1,000 rounds of SHAC. In 1,000 rounds of EECS, there are 30 nodes at least, which is 3% of total number of node, there are 100 nodes at least, which is 10% of total number of node in 1,000 rounds in LEACH-C. The dead number of node shows balance network energy consumption. The less there are dead nodes, the better we can do balance network energy. SHAC both prolongs lifetime of network and reduces the number of dead nodes. Hence, SHAC more efficiency balances the energy consumption of network compared to the other two strategies. EECS introduces a cost function, but its performance is not better than SHAC, the reasons have three points. Firstly, it does not consider energy consumption of node in selecting tentative clusterhead. Secondly, it only considers the distance factor and omits residual energy of node and clusterhead. Lastly, it does not consider the cost of each node and clusterhead. It has considered the residual energy of node in LEACH-C, but overhead about transmitting information is bigger, so it has the worst performance.

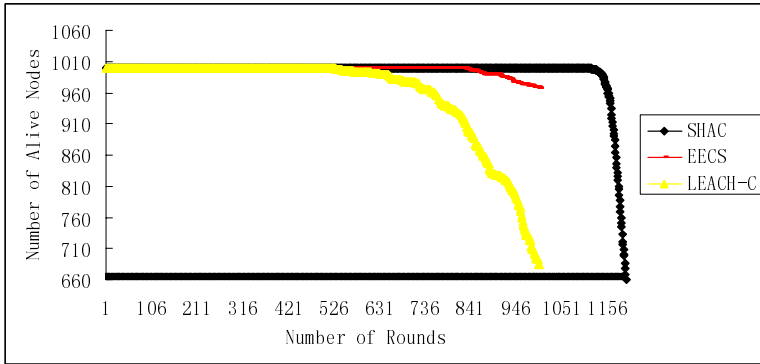


Fig. 3. Lifetime of network over simulation time

5 Conclusions and Future Works

In this paper, a Single-Hop Active Clustering (SHAC) algorithm is proposed about wireless sensor networks by research current routing algorithms. The core of SHAC has three parts. Firstly, when tentative clusterheads are selected, an average energy factor is introduced to balance the residual energy of the whole network nodes so that it may improve the network energy-efficient. Secondly, a cost function is proposed to balance energy-efficient of each node. Last but not least, an active clustering algorithm is proposed in single-hop homogeneous network. Through both theoretical analysis and numerical results, it is shown that SHAC prolongs the network lifetime significantly against the other clustering protocols such as LEACH-C and EECS. Under general instance, SHAC may prolong the lifetime 30% at least, especially, it may prolong the lifetime up to 50% against EECS.

In future research, we will consider NS2 simulation platform using event-driven mechanism to simulate performance of the SHAC algorithm. In LEACH-C, EECS and SHAC, we assume that data are transmitted at any moment, but for event-driven network, in no events, nodes do not consume energy and keep sleeping status. Once there is a event, the node is waked to collect data and communicate, so this can improve energy-efficient of sensor network so that this make SHAC is better to apply in real condition.

Acknowledgment

The author would like to thank the Science and Technology Research Project of Chongqing Municipal Education Commission of China under Grant No. 080526 and Chongqing Natural Science Foundation under Grant No. 2009BB2081. The author would also like to thank to MATLAB software.

References

1. Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: Energy-efficient routing protocols for wireless microsensor networks. In: Proc. 33rd Hawaii International Conference on System Sciences, vol. 8, pp. 8020–8030 (2000)
2. Manjeshwar, A., Grawal, D.P.: TEEN: A protocol for enhanced efficiency in wireless sensor networks. In: Proc. of the 15th Parallel and Distributed Processing Symp., pp. 2009–2015. IEEE Computer Society, San Francisco (2001)
3. Younis, O., Heed, F.S.: A hybrid, energy-efficient, distributed clustering approach for ad-hoc sensor networks. *IEEE Trans. on Mobile Computing* 3(4), 660–669 (2004)
4. Chan, H., Perrig, A.: ACE: An emergent algorithm for highly uniform cluster formation. In: Karl, H., Wolisz, A., Willig, A. (eds.) EWSN 2004. LNCS, vol. 2920, pp. 154–171. Springer, Heidelberg (2004)
5. Fang, Q., Zhao, F., Guibas, L.J.: Lightweight sensing and communication protocols for target enumeration and aggregation. In: Proc. of the 4th ACM Int'l Symp. on Mobile Ad Hoc Networking & Computing, pp. 165–176. ACM Press, New York (2003)
6. Ye, M., Li, C., Chen, G., Wu, J.: EECS: An energy efficient cluster scheme in wireless sensor networks. In: Dahlberg, T., Oliver, R., Sen, A., Xue, G.L. (eds.) Proc. of the IEEE IPCCC 2005, pp. 535–540. IEEE Press, New York (2005)
7. Depedri, A., Zanella, A., Verdone, R.: An energy efficient protocol for wireless sensor networks. In: Proc. of the AINS 2003, Menlo Park, pp. 1–6 (2003)
8. Heinzelman, W.R., Chandrakasan, A.P., Balakrishnan, H.: An application-specific protocol architecture for wireless microsensor networks. *IEEE Trans. on Wireless Communications* 1(4), 660–670 (2002)
9. Smaragdakis, G., Matta, I., Bestavros, A.: SEP: A stable election protocol for clustered heterogeneous wireless sensor networks. In: Proc. of the Int'l Workshop on SANPA, Boston, vol. 4, pp. 660–670 (2004)
10. Li, Q., Zhou, Q.-i., Wang, i.-W.: A Distributed Energy-Efficient Clustering Algorithm for Heterogeneous Wireless Sensor Networks. *Chinese Journal of Software* 17(3), 481–489 (2006)
11. Handy, M.J., Haase, M., Timmermann, D.: Low energy adaptive clustering hierarchy with deterministic cluster-head selection. In: Proc. of the 4th IEEE Conf. on Mobile and Wireless Communications Networks, pp. 368–372. IEEE Communications Society, Stockholm (2002)
12. Li, C.-F., Chen, G.-H., Ye, m., et al.: An uneven cluster-based routing protocol for wireless sensor network. *Chinese Journal of Computers* 30(1), 27–36 (2007)

A Transelevator Moving Inside of an Automatic Warehouse in Virtual Reality

Jose de Jesus Rubio, Enrique Garcia, and Jaime Pacheco

Instituto Politecnico Nacional - ESIME Azcapotzalco
Seccion de Estudios de Posgrado e Investigacion
Av. de las Granjas 682, Col. Sta. Catarina. Azcapotzalco, México D.F.
jrubioa@ipn.mx

Abstract. In this research, it is simulated the computed movements of a transelevator inside of a warehouse in virtual reality. This transelevator can be used to move some load from the floor to the deposit, and from the deposit to the floor, or can be used move the load from one place of the deposit to another one. The virtual reality is simulated using the graphic designer Quest3D. It is presented the simulation of the system.

1 Introduction

The automatic warehouse are elements used to make easy the works of moving loads from one side to another place in an automatic way. In the market, it is necessary to provide some tools to help to interact many systems. One of the tools used in the some complex applications is the virtual reality because in the virtual reality the designer can make many proofs in a simulator avoiding the necessity to prove it in the real system.

In the recent years, there are some researches related with the virtual reality as are [1], [2], [3], [4], [5], [6], [7], [8], [9].

In [2], they presented the projection based on a virtual reality and the use of a cave. In [3], they comment that the training on patients has raised concern among the profession and the public about how physicians will acquire sufficient skill to safely perform new, potentially high-risk, endovascular procedures such as carotid stenting. In [4], the authors present the virtual reality of a surgery applied to a patient. In [6], VR exposure (VRE) is proposed as an alternative to typical imaginal exposure treatment for Vietnam combat veterans with posttraumatic stress disorder (PTSD). In [5] and [7], they determine the relative efficacy of exposure to a virtual airplane and standard exposure to an actual airplane at the airport compared with wait list control in the treatment of fear of flying. In [8], they present a virtual reality for a electromagnetism laboratory. In [9], they discuss two experimental studies to assess the impact on presence of this method in comparison to the usual hand-pointing method of navigation in virtual reality, their studies sauteed that subjective rating of presence is enhanced by the walking method provided that participants associate subjectively with the virtual body provided in the environment. This authors presented the virtual

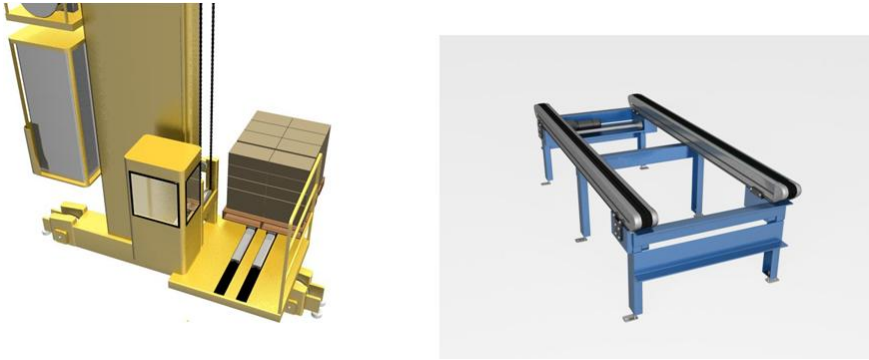


Fig. 1. Some 3D objects used in the simulation

reality applied to the medicine, but none of them applied the virtual reality for arm robots. In [1], they present the main characteristics of a virtual laboratory and some practices used to teach students the robotic.

In this paper it is presented the process for the build of an virtual reality in real time for an evaluation of an automatic warehouse (AS/RS) using the graphic designer Quest3D which is commonly used for the design of video games, but now is used for applications used in factories. In this case are presented the movements of a transelevator, this program is capable to interact with its environment which in this case is an automatic warehouse. Some of the characteristics of this system are: it can analyze the behavior of the warehouse, it can reduce the time used to develop the real application, it can make easy to see the department store in 3D. It is given a simulation of the system.

2 Building of the Physical Model of the Department Store

The elements considered in the warehouse are made in the Autodesk 3DStudio Max. It is important that was used the method called "lowpoly" to show in the screen the least number of polygons, and was used the method called "Texture Baker" to map the 3D object with only one map of bits, and including shadows and channels "alpha" (cut of a 3D object with an image). In Fig. 1 are given some 3D objects used in the simulation.

3 Building of the Logical Model

The logical model is used to compute the time employed by the works made for the transelevator, the simple and the combined works are considered. For the transelevator moving in the department store considers some assumptions, it produces some data written as mathematical equations. For a simple cycle of work are considered the assumptions given in the Fig. 2.

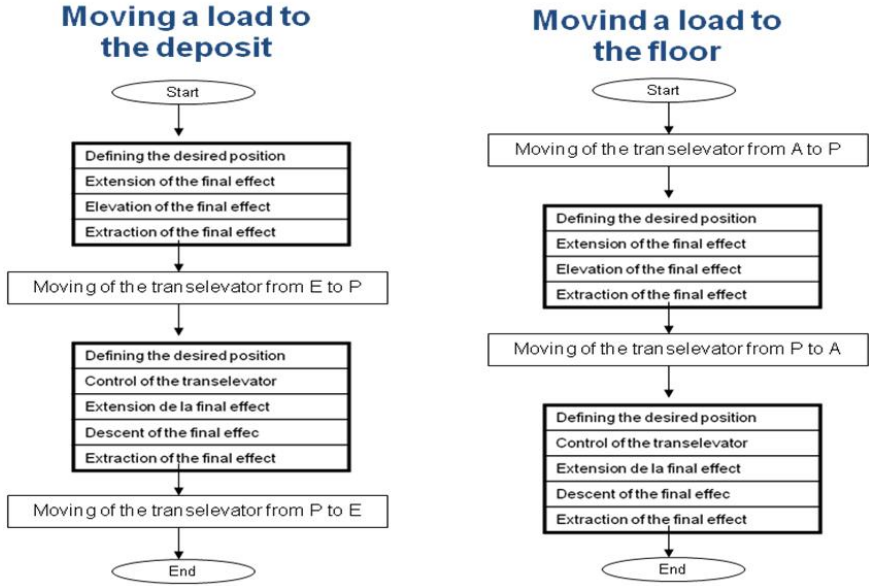


Fig. 2. Assumptions considered for a simple cycle of work

The most frequent works are:

- Case 1: Putting the load and taking from one inferior place.
- Case 2 : Putting the load at the start of the deposit and taking the load at the final of it.
- Case 3 : Putting and taking in the elevation.
- Case 4 : Putting and taking considering the moving in the X direction.
- Case 5 : Putting in an bottom place and taking from a top place.
- Case 6 : Putting in at the top place with the taking in a bottom place.

The reference points (desired points) of putting and taking are defined P_1 and P_2 as:

$$\begin{aligned} P_1 &= \left(\frac{1}{3}L, \frac{2}{3}H\right) \\ P_2 &= \left(\frac{2}{3}L, \frac{1}{3}H\right) \end{aligned} \tag{1}$$

Considering points P_1 and P_2 will be built the model to obtain the time employed in the simple cycles using the cases of before.

3.1 Case1: Putting the Load and Taking from One Inferior Place

The time used to the movement to one point will be the maximum of the vertical and the horizontal because when the transelevator moves in a longitudinal way, it moves in a vertical way, too. At this way, to compete the move to the two points proposed by the norm of P_1 and P_2

3.2 Case 2: Putting from the Point of Angle E. Taking from the Point of Angle A

It will be the same as the Case1, i.e., it will be computed the movement until P_1 and the time for the movement until P_2 , then will be considered the average. At this value will be added twice the transfer time and it will be the average of time of one cycle of putting.

3.3 Case 3: Putting and Taking in Elevation from One Point

To compute the time employed for a simple cycle will be the same as in the Case 1, but taken in count that in this case the points A and E are points with coordinates different of $(0, 0)$.

3.4 Case 4: Putting and Taking Considering Movements in X Direction, E=A

To compute the time employed for a simple cycle will be the same as in the Case 1, but taken in count that in this case the points A and E are points with coordinates different of $(0, 0)$.

3.5 Case 5: Putting from a Point of Angle E. Taking the Load in Elevation in de Y Direction

It will be same as the Case1, i.e., it will be computed the movement until P_1 and the time for the movement until P_2 , then will be considered the average. At this value will be added twice the transfer time and it will be the average of time of one cycle of putting.

3.6 Case 6: Putting the Load in Elevation in Y Direction. Taking the Load from One Point of Angle A

To compute the time employed for a simple cycle will be the same as in the Case 1, but taken in count that in this case the points A is a point with coordinates different of $(0, 0)$.

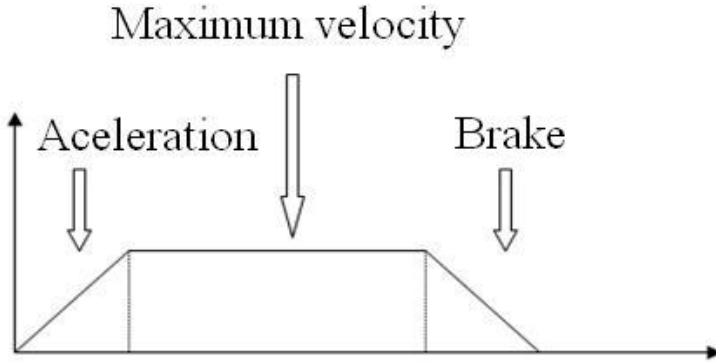
4 Example of a Case

Technical data:

$$\begin{aligned} L &= 70m & H &= 25m \\ V_x &= 150m/\text{min} & V_y &= 60m/\text{min} \\ A_x &= 0.5m/s^2 & A_y &= 0.5m/s^2 \end{aligned}$$

Reference Time= 10 sec.

1. Take the load in the input (E).
2. Time from E to P_1 with coordinates $(14, 16.67)$



To get the time employed to be at P_1 will be considered the following: The time of acceleration denoted as T_{acel} , the movement with constant velocity denoted as $T_{ctelevel}$, and the time to brake denoted as T_{brake} .

In X :

$$T(E \rightarrow P_1) = T_{acel} + T_{ctelevel} + T_{brake} = 10.60 \text{ sec}$$

In Y :

$$T(E \rightarrow P_1) = T_{acel} + T_{ctelevel} + T_{brake} = 18.67 \text{ sec}$$

Then:

$$T(E \rightarrow P_1) = \max(18.67, 10.60) = 18.67 \text{ sec}$$

3. Putting the load in P_1

4. Time employed for the movement at the input (A)

$$T(P_1 \rightarrow A) = 18.67 \text{ sec}$$

After, the time employed for a simple cycle is:

$$C_s(P_1) = 10 + 18.67 * 2 + 10 = 57.34 \text{ sec}$$

5 Simulation

The program used to simulate the construction of the virtual reality of the translevator and the warehouse is one variant of the DirectX called Quest3D.

Quest3D is the union of a graphic designer with a platform of develop, it is generally used for architecture, video games and for simulators. the CAD models are exported to Quest3D where are used for the simulation of 3D applications in real time.

One of the most important characteristics of the Quest3D is the method of programing. It is like Visual Basic, it uses blocks and some programing for this blocks. The programmer can change the program while it is running. There is not compilation in this program.

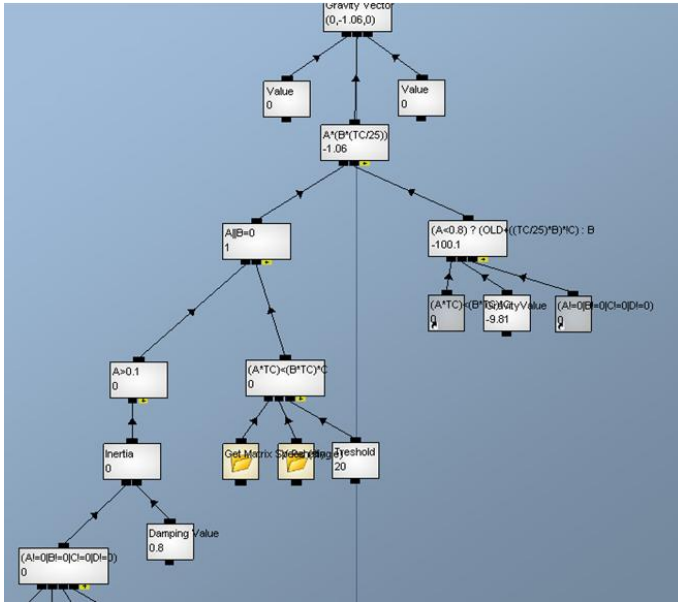


Fig. 3. Simulation of the gravity effect

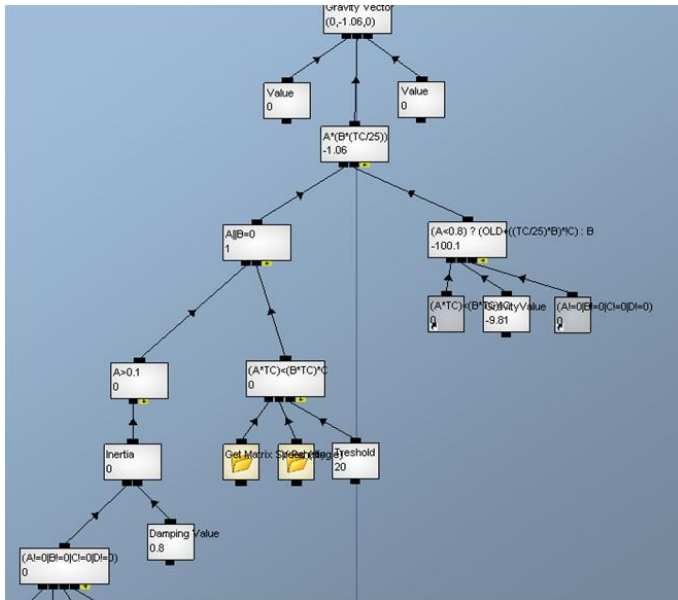


Fig. 4. Simulation of the translevator

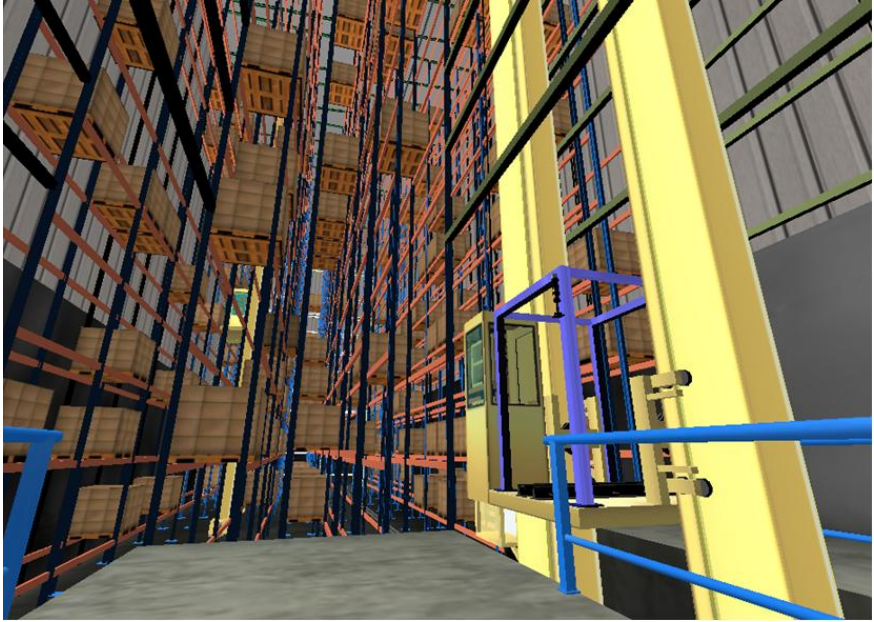


Fig. 5. A translevator in the right side



Fig. 6. A translevator taking a load

The application in Ques3D are made connecting functional components called channels. The channels form a structure of tree.

Fig. 3 shows the simulation of the gravity effect in the deposit. Fig 4 shows the simulation of the transelevator. Fig. 5 shows a transelevator in the right side, it moves from movements programed previously. It can be controlled in real time. Fig. 6 shows a transelevator taking a load.

6 Conclusion

In this research, it was simulated the computed movements of a transelevator inside of a warehouse in virtual reality. This transelevator can be used to move some load from the floor to the deposit, and from the deposit to the floor, or can be used move the load from one place of the deposit to another one. The virtual reality was simulated using the graphic designer Quest3D. In this paper the movements of the transelevator are programed previously. As a future research, the control of the transelevator will be considered.

Acknowledgment. The authors are thankful with the editor to invite them to be part of the committee.

References

1. Candelas, F.A., Torres, F., Gill, P., Ortiz, F., Puente, S., Pomares, J.: Laboratorio virtual remoto para robotica y evaluacion de su impacto en la doscencia. *Revista Iberoamericana de Automática e Informática Industrial* 1(2), 49–57 (2004)
2. Cruz, C., Sandin, D.J., DeFanti, T.A.: Surround-Screen Projection- Based Virtual Reality: the design and implementation of the cave, pp. 135–142. ACM, New York (1993)
3. Gallegher, A.G., Cates, C.U.: Approval for virtual reality training for carotid stending. *IEEE Transactions on Automatic Control* 37, 1234–1237 (1992)
4. Navarro, A.A., Veléz, J.A., Múnera, L.E., Bernabé, G.: Arquitectura de software basada en la internet basada en la simulacion virtualde cirugia endoscópica otorrinolingologica, *Sistemas y Telemática*, 35–45 (1998)
5. Oslasov, B., Hodges, L., Watson, B.A., Drew, G., Opdyke, D.: Reality exposure therapy in the treatment of fear of flying: a case report. *Behav. Res. Ther.* 34(56), 447–481 (1996)
6. Oslasov, B., Hodges, L., Alarcon, R., Ready, D., Shaha, F., Graap, K.: Virtual reality exposure Therapy for PTSD Vietnam veterans: A case study. *International Society for traumatic stress studies*, 263–271 (1999)
7. Oslasov, B., Hodges, L., Watson, B.A., Drew, G., Opdyke, D.: A Controlled Study of Virtual Reality Exposure Therapy for the Fear of Flying. *Journal of Consulting and Clinical Psychology* 68(6), 1020–1026 (2000)
8. Rosado, L., Herreros, J.L.: Nuevas aportaciones didácticas de los laboratorios virtuales y remotos en la enseñanza de la Física. In: *International Conference on Multimedia and ICT in Education*, pp. 1–5 (2005)
9. Slater, M., Usoh, A.S.: Taking Steps: The Influence of a Walking Technique on Presence in Virtual Reality. *ACM Transactions on Computer-Human Interaction* 2(3), 201–219 (1995)

Improved AFT and Background-Mesh Generation for FEM with Applications in Microwave

Xin Li, Changying Wu, Jie Li, and Jiadong Xu

School of Electronics and Information, Northwestern Polytechnical University
Xi'an, 710072, China
xxingg@hotmail.com

Abstract. A hybrid method for the automatic generation of finite element meshes is presented. The original domain awaiting to be meshed is divided into two areas, the squares of the interior part are split into isosceles right triangles, which results in uniform meshes with excellent internal triangles like the Almost Regular Triangulations. The region near boundary is meshed by advancing front technique (AFT). Compared with other methods, it's capable of automatically and adaptively generating meshes with reasonable quality. It is also easy to implement in program, and save the CPU time greatly. The method is proved to be effective by application to calculate the RCS of sphere and cylinder.

Keywords: Adaptive, Advancing Front Technique, FEM, Mesh generation, RCS.

1 Introduction

The finite-element method has proven its worth in both microwave and low-frequency electrical engineering applications include circuits and devices, as well as scattering and radiation problems. It has been more than 30 years since this method was used for solving electromagnetic problems. Relative growth of the literature on this subject appears to have stabilized, suggesting that the method has matured.

Electromagnetic analysis with the finite-element method can be thought of as a three-step process: preprocessing (mesh generation), analysis, and then postprocessing. The preprocessing step is often incorporated into the analysis code, or considered to be part of the mesh-generation step. For large, complex objects (especially in three dimensions), this task can be very cumbersome, and should be considered as a separate process. According to some statistics, it indicates that the three parts, respectively occupy 40%-50%, 5%, 50%-55% of the total computation time. Along with the maturing of computing method, the pre- and post-processing received more and more attention. Therefore, after continually developing for over 30 years, mesh generation, the most important part of pre-processing, is still active in the domain of engineering science and computation science.

Since the last century, a lot of papers and method are presented during the research of mesh generation, developed some business software successfully. Mesh generation

methods are classified by Ho-Le [1, 2, 8], are categorized as mapping methods, grid-based methods (include finite quadtree/octree methods) [3], Delaunay triangulation [4] and Advancing Front Technique (AFT) method [5, 6]. The necessity of fine meshes with good quality is well known as the first step for solving problems using FEM. Generally, it involves the generation of an initial mesh, which is then refined accordingly to ensure accurate finite element solutions.

2 Method

The algorithm initiates with the generation of a mesh template made up only of squares; then, this grid is superimposed on the domain. The squares that fall outside the domain are removed, the squares that fall inside the domain boundary are split into triangles; the squares that intersect the domain boundary are adjusted to triangles. The first part of the algorithm is very easy, but in the second part, which adjusts the triangles that intersect the boundary, the algorithm is innovative and results in high quality triangles.

2.1 Two Dimensional Mesh Generation Procedure

Our approach for mesh generation is composed of four steps. The first two steps are similar to the approaches described in [7].

Step1. Generation of the Square Mesh

Starting from the original domain (e.g. a circle), calculate the smallest values of the x and y coordinates on the boundary, x_{min} and y_{min} and then their largest values, x_{max} and y_{max} , find the corresponding points: $A(x_{min}, y)$, $B(x, y_{min})$, $C(x_{max}, y)$, $D(x, y_{max})$, as shown in Fig. 1.

Then square meshes are generated, the squares having the same size, where h (the length of the edge) is wavelength divided by 10. It is a default value, but other choices are also possible when the mesh need to be dense. The four vertex's coordinates of one mesh are:

$$A_{ij}(x_{min}+i*h, y_{min}+j*h), B_{ij}(x_{min}+(i+1)*h, y_{min}+j*h),$$

$$C_{ij}(x_{min}+(i+1)*h, y_{min}+(j+1)*h), D_{ij}(x_{min}+i*h, y_{min}+(i+1)*h).$$

It's shown in Fig. 2.

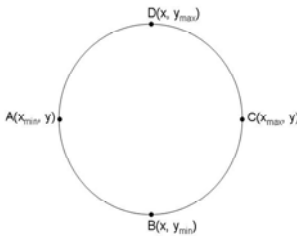


Fig. 1. Original domain awaiting to be meshed

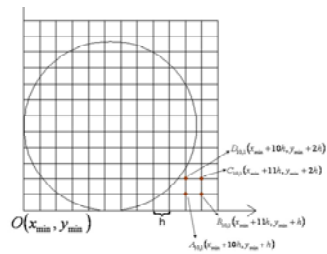


Fig. 2. Original domain meshed by squares

If the nodes on the zigzag boundary that are located at a distance less than $L/5$ (L is the length of the square) of the domain boundary, replace the domain boundary with mesh.

Step2. Generation of the Triangular Mesh

All the squares are split into triangles.

- a). If all the vertexes aren't in the domain to be meshed, the mesh is removed, as shown in Fig. 3.
- b). If its four nodes are located in the domain, the square is kept in the mesh, connect the two cross vertexes, then the square is split into two isosceles right triangle, as shown in Fig. 4.

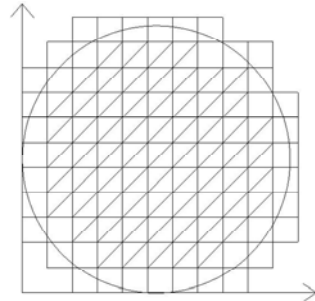
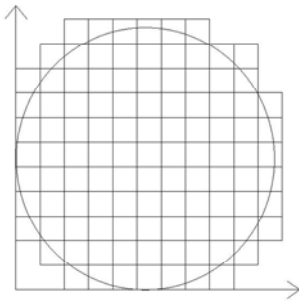


Fig. 3. Squares in the domain are split into triangles

Fig. 4. Squares out of the domain are removed

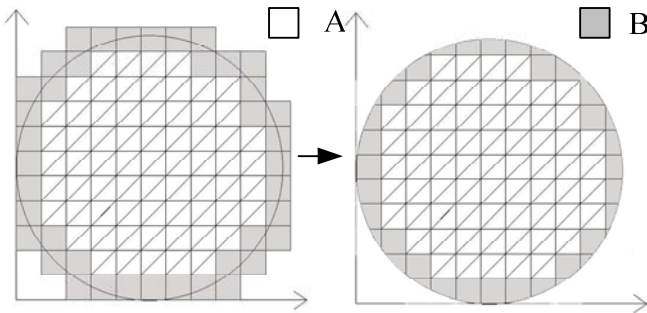


Fig. 5. Domain is divided into two areas

Step3. Mesh Generation of Region B

The boundary of region A and B is used as the front of AFT method, node is used as the discrete points of AFT, then the mesh of region B is generated by AFT.

Step4. Improve the Quality of the Triangular Mesh

Function Q is defined to measure the quality of the triangle

$$Q = L_{\max} / L_{\min} \quad (1)$$

L_{\max} and L_{\min} is the length of the largest side and shortest side of the triangle, $Q=1$ means equilateral triangle is the optimal element. $Q<2$ is eligible.

If the result isn't satisfied, it has badly shaped elements on the boundary. Enlarge region B, as shown in Fig 6, then mesh it again. The result will be much better.

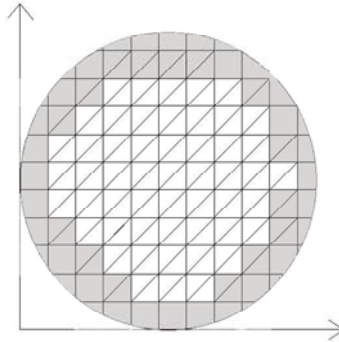


Fig. 6. Enlarge region B

2.2 Three Dimensional Mesh Generation Procedure

The construction of the mesh is similar to the steps of the two-dimension:

Step1. Generation of the Square Cube

Starting from the original domain, calculate the smallest values of the x, y and z coordinates on the boundary, x_{\min} , y_{\min} , z_{\min} and then their largest values, x_{\max} , y_{\max} , z_{\max} , find the corresponding points: $A(x_{\min}, y, z)$, $B(x, y_{\min}, z)$, $C(x_{\max}, y, z)$, $D(x, y_{\max}, z)$, $E(x, y, z_{\min})$, $F(x, y, z_{\max})$. Then square cubes are generated, the cubs having the same size, where the length of the edge, h, can choose the smallest of the values: $(x_{\max} - x_{\min})$, $(y_{\max} - y_{\min})$ and $(z_{\max} - z_{\min})$ divided by 10, or wavelength divided by 10. This is a default value, but other choices are also possible when the mesh need to be dense. The eight vertex's coordinates of one mesh are:

$$\begin{aligned} &A_{ij}(x_{\min}+i*h, y_{\min}+j*h, z_{\min}+k*h), B_{ij}(x_{\min}+(i+1)*h, y_{\min}+j*h, z_{\min}+k*h), \\ &C_{ij}(x_{\min}+(i+1)*h, y_{\min}+(j+1)*h, z_{\min}+k*h), D_{ij}(x_{\min}+i*h, y_{\min}+(i+1)*h, z_{\min}+k*h), \\ &E_{ij}(x_{\min}+i*h, y_{\min}+j*h, z_{\min}+(k+1)*h), F_{ij}(x_{\min}+(i+1)*h, y_{\min}+j*h, z_{\min}+(k+1)*h), \\ &H_{ij}(x_{\min}+(i+1)*h, y_{\min}+(j+1)*h, z_{\min}+(k+1)*h), \\ &I_{ij}(x_{\min}+i*h, y_{\min}+(i+1)*h, z_{\min}+(k+1)*h). \end{aligned}$$

Step2. Generation of the Tetrahedron

The square cube is split into tetrahedron.

a). If all the vertexes of the square cube locate in the domain to be meshed, connect the two diagonal, then the square cube is split into three tetrahedron.

- b). If all the vertexes of the square aren't in the domain to be meshed, delete this mesh.
- c). The square cube intersect the domain boundary, the remainder region is meshed by AFT [6].

Step3. Improve the Quality of the Tetrahedron

Enlarge the region adjacent to the boundary, and then mesh it again. The result will be much better.

3 Examples

3.1 Examples of Two Dimensional Meshes Generated

Giving some problem to be meshed, we can see the result using the new adaptive mesh generation, the element have good shapes, as shown in Fig 7, 8. By comparison with FDTD mesh shown in Fig 9, the mesh of Fig 7 is more conformal.

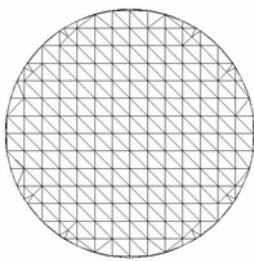


Fig. 7. Mesh of a circle

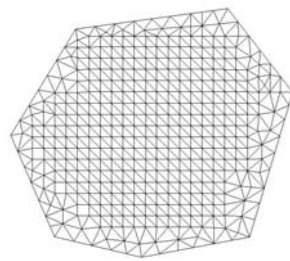


Fig. 8. Mesh of a polygonal

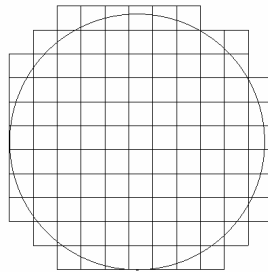


Fig. 9. Mesh of a circle in FDTD

3.2 RCS of Sphere Calculated with FEM

In the example, we present initial tests of the PML formulation for scattering from a conducting sphere with the radius 0.2λ . The thickness of the PML region is 0.2λ and the improved AFT and background-mesh method in the paper is used to discretize it.

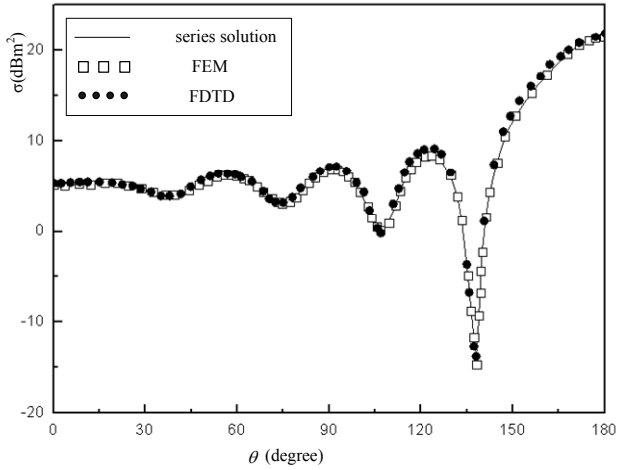


Fig. 10. RCS of sphere

3.3 RCS of Cylinder Calculated with FEM

In the example, we present initial tests of the PML formulation for scattering from a dielectric circular cylinder with five coating layers, thickness of the layer is 0.1λ , radius of the cylinder is 0.1λ .

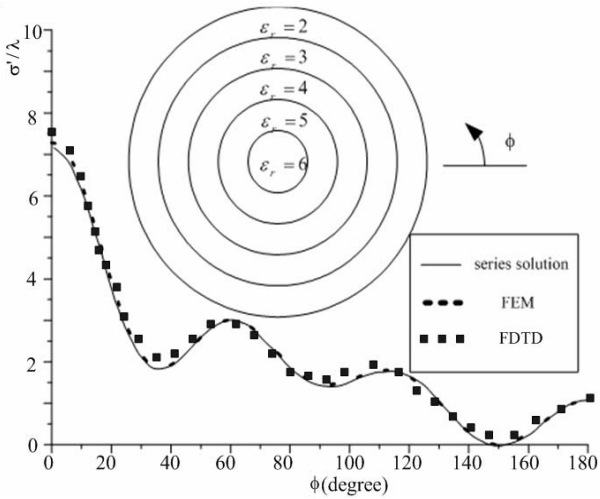


Fig. 11. RCS of cylinder

4 Conclusion

In this paper, a hybrid FE mesh generation method from triangular mesh using AFT and grid-based method was proposed. It is easy to realize in program in two dimension and three dimension. RCS calculation using FEM with the mesh method is approximate to the series solution than FDTD, which is shown in Figure 10, 11.

Compared with other methods, this algorithm does not require a search process and, will save the CPU time greatly. It's capable of automatically and adaptively generating meshes with reasonable quality, the examples here clearly demonstrate the effectiveness and usefulness of the algorithm. The generated meshes have good quality and ensure a smooth transition between large and small elements.

Our future work is to extend our algorithm to more complicated models, improve the quality of the mesh in three-dimension.

Acknowledgments. The work was supported by Science and Technology Innovation Fund of Northwestern Polytechnical University under the grant No. 2006CR11 and the Natural Science Basic Research Plan in Shaanxi Province of China under the grant No. 2006F15.

References

1. Ho-Le, K.: Finite Element Mesh Generation Methods: A Review and Classification. *Computer-Aided Design* 20(1), 27–38 (1988)
2. Coccioli, R., Itoh, T., Pelosi, G., Silvester, P.P.: Finite-element Methods in Microwaves: a Selected Bibliography. *IEEE Antennas and Propagation Magazine* 38(6), 34–48 (1996)
3. Yerry, M.A., Shephard, M.S.: A Modified Quadtree Approach to Finite Element Mesh Generation. *IEEE Computer Graphics & Applications* 3(1), 39–46 (1983)
4. Shenton, D.N., Cendes, Z.J.: Three Dimensional Finite Element Mesh Generation Using Delaunay Tessellation. *IEEE Transactions on Magnetics* 21(6), 2535–2538 (1985)
5. Lohner, R.: Progress in Grid Generation via the Advancing Front Technique. *Engineering with Computers* 12(3-4), 186–210 (1996)
6. Lo, S.H.: Volume Discretization into Tetrahedra II. 3D Triangulation by Advancing Front Approach. *Computers and Structures* 39(5), 501–511 (1991)
7. Li, X., Wu, C.Y., Li, J., Xu, J.D.: New Algorithm of Automatic FEM Mesh Generation for Electromagnetic Scattering Calculation. *ICEMI*, 2547–2550 (2007)
8. Guan, Z.Q., Song, C., Gu, Y.X., et al.: Recent Advances of Research on Finite Mesh Generation Methods. *Journal of Computer-Aided Design & Computer Graphics* 15(1), 1–14 (2003)

Application of Non-redundant Association Rules in University Library

Xiukuan Chen, Xiangjun Dong, Yuming Ma, and Runian Geng

School of Information Science and Technology, Shandong Institute of Light Industry,
Jinan 250353, P.R. China
xiukuan8410@163.com, dxj@sdili.edu.cn, mym@sdili.edu.cn,
gengrnn@163.com

Abstract. In view of the number of the association rules to the libraries' book circulation data are too large to analyze, this paper introduced the definition and the theorem of redundant rules, then proposed the redundancy theorem in negative association rules. Combined with correlation, the paper gave us an algorithm and apply it to the library's information services, the experiment show that it can mine the implied discipline, so that it can better service for library management, research and teaching.

1 Introduction

There are large amount of data at the books' circulation process, and the reader's loan information is one of the most important data. According to readers' long-term lending data, we will find there is correlation between the readers' lending literature, and there also exist correlation between different disciplines. However the number of the association rules produced by traditional algorithms is too large to analyze, therefore we should mine non-redundant association rules which useful for decision-making, then it can rational allocation of resources and improve resource utilization, thereby improve the level of library services.

Mining association rules is an important research field of data mining, it reflects the interesting association or correlation between a great deal of itemsets. The research has important actual significance and application value, However, we usually get a lot of association rules, which contain a large number of redundant rules. It's difficult for users to analyze and use these rules. Therefore it is necessary to prune redundant rules. At present, algorithms reference to the deletion of redundant association rules are not a lot, they consider the algorithm from different points of view. In [1] the authors proposed a new algorithm MVNR (Mining Valid and non-Redundant Association Rules Algorithm), which primly solved above problems by using the minimal subset of frequent itemset; In [2] the authors presented a general algorithm for mining non-redundant rules from the largest frequent itemsets using the redundant relationship of rules. The algorithm eliminates the redundancy between the rules and reduces the number of rules exponentially; they are both relatively rapid and efficient methods; Literature [3] mentioned an algorithm is a variant of Apriori algorithm which can also reduce the number of rules. [4] presents an algorithm which

can calculate the support and confidence of negative association rules, then presents a PNARC model which can mine positive and negative association rules. The model can delete the contradictory rules and get correct positive and negative association rules. In [5] the authors presented the notions and algorithms of generating basis for exact rules and the proper basis for conditional rules of redescription database are presented using operations of formal concept analysis and closure operator of Galois connection in the paper. In literature [6], the concept of minimal association rules is introduced from the aspect of application. Minimal rule set, which includes rules with single item as consequent and the minimal number of items as the antecedent, can be used to derive the same decisions as other association rules without information loss, while the number of minimal rules is much less than of all rules. However, they are not pruning redundancy rules in negative association rules entirely and effectively. This paper introduced the definition and the theorem of redundant rules, and proposed the redundancy theorem in negative association rules. Combined with correlation, the paper gave us an algorithm and apply it to the library's information services, the experiment show that it can mine the implied discipline, so that it can better service for library management, research and teaching.

The main contributions of this paper are as follows:

1. We proposed the redundancy theorems in negative association rules.
2. Combined with correlation, we proposed an algorithm MNTP which can mine non-redundant association rules and apply it to the library's information services.

The rest of this paper is organized as follows. Section 2 is the improved methods of pruning redundant rules. The description and analysis of the algorithm is presented in Section 3. Section 4 is experimental results and Section 5 is conclusions.

2 Improved Methods of Pruning Redundant Rules

2.1 Related Concepts

Sequence is a schedule of itemsets in order.

Assumed that all items of the itemsets are instead by the integer in a row, then a itemsets of i can be expressed as (i_1, i_2, \dots, i_m) , i_j represents a item. s is a sequence can be expressed as $\langle s_1, s_2, \dots, s_n \rangle$, s_j represents a itemsets.

Assume that given us two sequences $a \langle a_1, a_2, \dots, a_n \rangle$ and $b \langle b_1, b_2, \dots, b_m \rangle$, if there is a whole number $i_1 < i_2 < \dots < i_n$ and a_{i_1} included in b_{i_1} , a_2 included in the $b_{i_2} \dots, a_n$ contained in the b_{i_n} , then we said that the sequence a is included in the sequence b . In a sequences s , if the sequence does not contain any other sequence, then we said that sequence s is the largest item sequence.

Sequence rules can be in the form of $A \Rightarrow B$, A and B are the sequences. A negative association rule is an implication as $X \Rightarrow \neg Y$ (or $\neg X \Rightarrow Y$), in which $X, Y \subseteq T$ and $X \cap Y = \emptyset$.

Conditions for the establishment of the sequence rule: Given sequences A and B , their connection can be expressed as $C = \langle A, B \rangle$, the support of the rule is defined as $C.\text{sup}$, expressed the possibility of the new sequence which comes from the connection of A and B . The confidence of the rule is defined as $C.\text{sup}/A.\text{sup}$.

Support refers to the frequency that the rule appears, and the confidence mainly refers to the strength of the rule. *Minconf* order for the confidence threshold, express the minimum reliability of the rule. When the support and confidence of a rule is not less than the threshold, then we said the rule is a strong rule.

By the same token, for the sequences X and Y , The negative rule $X \Rightarrow \neg Y$ holds in the transaction set D with confidence $\text{conf}(X \Rightarrow \neg Y) = \frac{\text{supp}(X \cup \neg Y)}{\text{supp}(X)}$. A rule $X \Rightarrow \neg Y$ is a valid rule if its support and confidence meet user specified minimum support (*minsupp*) and minimum confidence (*minconf*) thresholds.

2.2 The Definition of Redundancy Rules

For the different point of view on the algorithm, researchers present different definitions of redundancy rules. There are some commonly used definitions and theorems [7] as follows:

1. Given the rules of $A \Rightarrow B$ and $C \Rightarrow D$, if the support and confidence of $C \Rightarrow D$ are greater than or equal to the support and confidence of $A \Rightarrow B$, then we say the rule $C \Rightarrow D$ is the redundant rule of $A \Rightarrow B$.

2. A rule is considered necessary if and only if it is neither simple nor strict redundant in relation to any other rules.

Briefly redundancy: Suppose we have two rules, $A \Rightarrow B$ and $C \Rightarrow D$, they meet the conditions of $A \cup B = C \cup D = I_k$, if $A \subseteq C$, then we say the rule $C \Rightarrow D$ is the simple redundancy Rule of $A \Rightarrow B$.

Strictly redundancy: Suppose the rule $A \Rightarrow B$ is generated by the itemsets I_i , that is, $A \cup B = I_i$; the rule $C \Rightarrow D$ is set by the itemsets I_j ($I_j \subseteq I_i$), that is, $C \cup D = I_j$; if the $A \subseteq C$, then we say the rule $C \Rightarrow D$ is the strictly redundancy rule of $A \Rightarrow B$.

3. Given the rules of $X \Rightarrow Y$ and $A \Rightarrow B$, if we can obtain the rule $A \Rightarrow B$ from $X \Rightarrow Y$, then we say the rule $A \Rightarrow B$ is the redundant rule of $X \Rightarrow Y$

4. Given the rules of $X \Rightarrow Y$ and $A \Rightarrow B$, if they meet $(A \cup B) \subseteq (X \cup Y)$, and $X \subseteq A$, then (1) we say the rule $A \Rightarrow B$ is the redundant rule of $X \Rightarrow Y$; (2) The total number of redundant rules is $(3^{|X|} - 2^{|X|} - 1) \cdot |Y|$ represent the number of items included in itemsets Y .

2.3 Theorems and Prove of Redundancy in Negative Association Rules

When we mine negative association rules, we will find the theorem and the definition of redundancy in association rules are not fully applicable in the negative association rules. For example, we can't derived from $\neg A \Rightarrow BC$ that $\neg A \Rightarrow B$ and $\neg A \Rightarrow C$. Therefore, we propose the redundancy theorem in negative association rules:

Theorem 1. association rules like $\neg A \Rightarrow B$:

Given the rules of $\neg A \Rightarrow B$ and $\neg A \Rightarrow C$, if $B \subseteq C$, then we say the rule $\neg A \Rightarrow B$ is the redundant rule of $\neg A \Rightarrow C$.

For example, the negative rule $\neg A \Rightarrow B$ is the redundant rule of $\neg A \Rightarrow BC$.

Theorem 2. association rules like $A \Rightarrow \neg B$: Given the rules of $A \Rightarrow \neg B$ and $A \Rightarrow \neg C$, if $C \subseteq B$, then we say the rule $A \Rightarrow \neg B$ is the redundant rule of $A \Rightarrow \neg C$.

For example, the negative rule $A \Rightarrow \neg BC$ is the redundant rule of $A \Rightarrow \neg B$.

Theorem 3. association rules like $\neg A \Rightarrow \neg B$:

Given the rules of $\neg A \Rightarrow \neg B$ and $\neg C \Rightarrow \neg D$, if $C \subseteq A$ and $C \cup D \subseteq A \cup B$, then we say the rule $\neg A \Rightarrow \neg B$ is the redundant rule of $\neg A \Rightarrow \neg C$.

For example, the negative rule $\neg A \Rightarrow \neg BC$ is the redundant rule of $\neg AB \Rightarrow \neg CD$.

Then we prove Theorem 1 as follows:

As $\neg A \Rightarrow C$ is a valid association rules, so it meet $supp(\neg A \Rightarrow C) \geq minsupp, conf(\neg A \Rightarrow C) \geq minconf$

Because

$$conf(\neg A \Rightarrow C) = supp(\neg A \Rightarrow C) / supp(\neg A) \geq minconf$$

$$conf(\neg A \Rightarrow B) = supp(\neg A \cup B) / supp(\neg A)$$

Set by the subset support quality of the itemsets, we know

$$supp(\neg A \cup B) \geq supp(\neg A \cup C)$$

$$\text{Therefore, } conf(\neg A \Rightarrow B) \geq conf(\neg A \cup C) \geq minconf$$

That is, we can derive $\neg A \Rightarrow B$ by association rules $\neg A \cup C$. Therefore, the rule $\neg A \Rightarrow B$ is the redundant rule of $\neg A \cup C$.

Other proof of theorem is similar to the above.

2.4 The Related Concepts of Correlation

When we mining positive and negative association rules at the same time, we should also consider another problem is correlation. [8] gave us the definition of itemsets correlation, they used the related knowledge of probability, used the itemsets A and B as two random events, the support of itemsets A and B , expressed as $supp(A), supp(B)$ are the probability of their occurrence. If $p(AB) = p(A)p(B)$, we incident that A, B are independent, otherwise, they are correlated. Similarly, if $p(ABC) = p(A)p(B)p(C)$, then that the events A, B, C are independent, otherwise, they are correlated. Correlation has an important property called closed up. If an itemset X is correlated, then all supersets of X are correlated. This can be proved by apagoge. Assume that A, B are correlated, and the A, B, C are not, then $p(AB) = p(ABC) + p(AB\neg C) = p(A)p(B)p(C) + p(A)p(B)p(\neg C) = p(A)p(B)$, that means A, B are independent, which conflicts with the assumption. So A, B, C are correlated.

Literature [9] proposed that the correlation of A and B can be measured by following formulation: $corr_{A,B} = p(A \cup B) / p(A)p(B) = supp(A \cup B) / supp(A)supp(B)$, of which, $p(A) \neq 0, p(B) \neq 0$.

$Corr_{A,B}$ have 3 possibilities:

(1) If $corr_{A,B} > 1$, then A and B is positively correlated, the more events A occur, the more events B do.

(2) If $corr_{A,B} < 1$, then A and B is negatively correlated, the more events A occur, the less events B do.

(3) If $corr_{A,B} = 1$, then A and B is independent, the occurrence of event A has nothing to do with the occurrence of event B .

In case (2), $corr_{A,B} = 1$, that is, $supp(A \cup B) = supp(A) \cdot supp(B)$, it is possible, then A and B are independent of each other, the event B has nothing to do with the emergence of A .

Although in theory we could obtain the rule $A \Rightarrow B$ from the rule $A \Rightarrow BC$, but in practice it is meaningless. In case (3), $corr_{A,B} < 1$, that is, $supp(A \cup B) < supp(A) \cdot supp(B)$, which is possible, then A and B negative correlation, $A \Rightarrow BC$ arrive at $A \Rightarrow B$ is wrong, there may be such a negative association rule as $\neg A \Rightarrow B$ or $A \Rightarrow \neg B$.

3 The Description and Analysis of the Algorithm

At present, there are many algorithms reference to sequential patterns and sequential rules, they consider them from different points of view. [10] introduced event-oriented negative sequential rules and have designed an efficient algorithm to discover such rules. An event-oriented negative sequential rule is a rule in the forms of $P \rightarrow \neg e$, $\neg P \rightarrow e$ and $\neg P \rightarrow \neg e$, where P is a positive sequential pattern and e denotes a single event. Ouyang and Huang proposed negative sequences as $(A, \neg B)$, $(\neg A, B)$ and $(\neg A, \neg B)$ [11]. Their idea is generating frequent itemsets first, based on which both frequent and infrequent sequences are found, and then negative sequential patterns are derived from infrequent sequences. A drawback of their algorithm is that both frequent and infrequent sequences have to be found at the first stage, which demands a large amount of space. Lin et al. designed an algorithm NSPM (Negative Sequential Patterns Mining) for mining negative sequential patterns [12]. In their designed negative patterns, only the last element can be negative, and all other elements are positive. It is actually a special case of general negative sequential patterns.

In this paper, when mining positive and negative association rules from sequential patterns, we have not only to consider the sequence mode but also all of its subsets, because it's subset may min rules which can't get from itself.

As we mining non-redundant association rules based on sequential patterns, so when we consider correlation, we should also affiliate the time factor. In the expression of correlation, $supp(A \cup B)$ means the support that B arises after A . We mining positive and negative association rules based on sequential patterns combine with correlation, the algorithm MNTP is as follows:

Input: FS : the sequence and its subsets; $minconf$: the minimum confidence;

Output: PAR : association rules sets; $NAR1$: negative association rules sets like $\neg A \Rightarrow \neg B$; $NAR2$: negative association rules sets like $A \Rightarrow \neg B$; $NAR3$: negative association rules sets like $\neg A \Rightarrow B$;

(1) $PAR = \emptyset$; $NAR1 = \emptyset$; $NAR2 = \emptyset$; $NAR3 = \emptyset$;

(2) // Mining positive and negative association rules from all sequential patterns for any itemset X in FS do begin

for any itemset $A \cup B = X$ and $A \cap B = \emptyset$ do begin

$corr = supp(A \cup B) / (supp(A) \cdot supp(B))$;

if $corr > 1$ then begin

(2.1) // Mining rules like $A \Rightarrow B$ and $A \Rightarrow \neg B$

```

if X in FS and  $\text{conf}(A \Rightarrow B) \geq \text{minconf}$  then
  PAR = PAR  $\cup$  {  $A \Rightarrow B$  };
if  $\text{conf}(\neg A \Rightarrow \neg B) \geq \text{minconf}$  then
  NAR1 = NAR1  $\cup$  {  $\neg A \Rightarrow \neg B$  };
end;
(3) return PAR, NAR1, NAR2, NAR3;

```

The following give us the technology of pruning positive and negative redundant rules.

Input: *PAR*: positive association rules ; *NAR1*: negative association rules like $\neg A \Rightarrow \neg B$; *NAR2*: negative association rules like $A \Rightarrow \neg B$; *NAR3*: negative association rules like $\neg A \Rightarrow B$;

Output: *PAR*: non-redundant association rules sets ; *NAR1*: non-redundant negative association rules sets like $\neg A \Rightarrow \neg B$; *NAR2*: non-redundant negative association rules sets like $A \Rightarrow \neg B$; *NAR3*: non-redundant negative association rules sets like $\neg A \Rightarrow B$;

```

(1) par =  $\Phi$ ; nar1 =  $\Phi$ ; nar2 =  $\Phi$ ; nar3 =  $\Phi$ ;
(2) //Mining four forms of non-redundant association rules
(3) //Mining non-redundant negative association rules
for each rule (  $X \Rightarrow Y$  )  $\in$  PAR do
  par = par  $\cup$  {  $X \Rightarrow Y$  };
for each rule (  $X \Rightarrow Y$  )  $\in$  par do
  for each rule (  $A \Rightarrow B$  )  $\in$  PAR do
    if ( (  $A \cup B \subseteq X \cup Y$  ) and  $X \subseteq A$  ) then
      PAR = PAR - {  $A \Rightarrow B$  };
(4) //Mining non-redundant negative association rules
like  $\neg A \Rightarrow \neg B$ 
for each rule (  $\neg X \Rightarrow \neg Y$  )  $\in$  NAR1 do
  nar1 = nar1  $\cup$  {  $\neg X \Rightarrow \neg Y$  };
for each rule (  $\neg X \Rightarrow \neg Y$  )  $\in$  nar1 do
  for each rule (  $\neg A \Rightarrow \neg B$  )  $\in$  NAR1 do
    if (  $X \subseteq A$  and  $X \cup Y \subseteq A \cup B$  ) then
      NAR1 = NAR1 - {  $\neg A \Rightarrow \neg B$  };
(5) //Mining non-redundant negative association rules
like  $A \Rightarrow \neg B$ 
for each rule (  $A \Rightarrow \neg Y$  )  $\in$  NAR2 do
  nar2 = nar2  $\cup$  {  $A \Rightarrow \neg Y$  };
for each rule (  $A \Rightarrow \neg Y$  )  $\in$  nar2 do
  for each rule (  $A \Rightarrow \neg B$  )  $\in$  NAR2 do
    if (  $B \subseteq Y$  ) then
      NAR2 = NAR2 - {  $A \Rightarrow \neg B$  };
(6) //Mining non-redundant negative association rules like  $\neg A \Rightarrow B$ 
for each rule (  $\neg A \Rightarrow Y$  )  $\in$  NAR3 do
  par = par  $\cup$  {  $\neg A \Rightarrow Y$  };

```

```

for each rule  $(\neg A \Rightarrow Y) \in \text{nar3}$  do
  for each rule  $(\neg A \Rightarrow B) \in \text{NAR3}$  do
    if  $(Y \subseteq B)$  then
       $\text{NAR3} = \text{NAR3} - \{\neg A \Rightarrow B\}$  ;
return PAR, NAR1, NAR2, NAR3 ;

```

4 The Experimental Results

The library's GDLIS record the details of the readers' lending information. Here we did an experiment on a simple list of readers which showed in Table 1. (All sequences are composed by one order itemsets.) We use $\{1,2,3,4,5,6\}$ to represent $\{\text{Computer field, English field, Medical field, Exercises field, Economics field, Legal field}\}$.

Table 1. Database Service

Database Service				
Stu dy No.	Library list	Stud y No.	Library list	
00	< 1,2,4 >	006	< 2,4,6 >	
1				
002	< 1,2,3,4 >	007	< 1,5,6 >	
The did in athlon64 MHZ, 512 WIN xp,	00			experiment
	< 2,4 >	008	< 3,6 >	AMD
	3			3000
	00			RAM,
	< 2,3,4,5 >	009	< 2,3,6 >	
	4			
	00			
	< 1,3,5 >	010	< 1,2,3,4,6 >	
	5			

VisualBasic environment. Set $\text{mins_FS} = 0.1$, $\text{minconf} = 0.40$. In Table 2, we can see that our algorithm can mine 4 forms positive and negative association rules from frequent itemsets, we can also see that the rule $\neg 2 \Rightarrow \neg 4$ which min from the algorithm of mining non-redundant association rules can make the same decision with the rules $\{\neg 1, 2 \Rightarrow \neg 4; \neg 2 \Rightarrow \neg 3, 4; \neg 2, 3 \Rightarrow \neg 4; \neg 2 \Rightarrow \neg 4\}$. This has fully demonstrated the effectiveness of our algorithm, that is, the reduction of rules did not missing the number of information.

Table 2. The experimental results

		Type	rules	After the pruning
5.	Number of association rules (<i>mins_</i> <i>FS</i> = 0.1, Use <i>Minconf</i> relevance to prune =0.4)	$A \Rightarrow B$	36	8
		$A \Rightarrow \neg B$	69	6
		$\neg A \Rightarrow B$	57	12
		$\neg A \Rightarrow \neg B$	83	8
			12	
		<i>Subtotal</i>	257	34

5 Conclusion

For the number of association rules are too difficult to be effective use by policy-makers, we proposed the redundancy theorem in negative association rules. Combined with correlation, the paper gave us an algorithm which can min non-redundant association rules from sequences and their subsequences. The experiment proved that association rules derived from our algorithm can help us to make decision more effective.

Acknowledgements

This work was partially supported by Natural Science Foundation of Shandong Province of China under Grant No.Y2007G25 and No.Y2008G26; Excellent Young Scientist Foundation of Shandong Province of China under Grant No. 2006BS01017.

References

- [1] Agrawal, R., Srikant, R.: Mining sequential patterns. In: Yu, P.S., Chen, A.S.P. (eds.) Eleventh International Conference on Data Engineering, Taipei, Taiwan, pp. 3–14. IEEE Computer Society Press, Los Alamitos (1995)
- [2] Juguang, L., Xueyuan, N., Zeming, J., Zhaowei, W.: Detailed development of the ARM application system. Tsinghua University Press, Beijing (2003)

- [3] Feng, G., Jianying, X.: Algorithm for Generating Non- Redundant Association Rules. *Journal of Shanghai Jiaotong University* 35(2), 2562258 (2001)
- [4] Weiping, W., Fu, L., Guiming, H.: An Efficient Algorithm for Rapidly Mining Non-redundant Rules. *Computer Engineering* 29(9), 90291 (2003)
- [5] Wei-chuan, Y., Yuan-yuan, W.: Non-redundant Rules for Redescription Datasets Mining Based on FCA. *Computer Engineering* (2008) 04-0066-05
- [6] Li, J., Xu, Y., Wang, Y., Wang, Y.: Minimal Association Rules and Mining Algorithm. *Computer Engineering* (2007) 13-046-03
- [7] Xiangjun, D., Shujing, W., Hantao, S.: Approach for Mining Positive & Negative Association Rules Based on 2-level Support. *Computer Engineering* (2005)
- [8] Podilchuk, C.I., Wenjun, Z.: Image-adaptive Watermarking Using Visual Models. *IEEE Journal on Selected Areas in Communications* 16(4), 525 (1998)
- [9] Han, J.W., Kamber, M.: *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, San Francisco (2006)
- [10] Zhao, Y., Zhang, H., Cao, L., Zhang, C., Bohlscheid, H.: Efficient mining of event-oriented negative sequential rules. In: To appear in Proc. of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2008 (2008)
- [11] Ouyang, W.-M., Huang, Q.-H.: Mining negative sequential patterns in transaction databases. In: Proc. of 2007 International Conference on Machine Learning and Cybernetics, Hong Kong, China, pp. 830–834 (2007)
- [12] Lin, N.P., Chen, H.-J., Hao, W.-H.: Mining negative sequential patterns. In: Proc. of the 6th WSEAS International Conference on Applied Computer Science, Hangzhou, China, pp. 654–658 (2007)

Global Structure Constraint: A Fast Pre-location for Active Appearance Model

Jie Ma, Dongwei Guo, Miao Liu, Kangping Wang, and Chun-Guang Zhou

College of Computer Science and Technology, Jilin University, China 130012
guodw@jlu.edu.cn

Abstract. In this paper, a global structure constraint model, GSC, is described. This model represents the target pattern as a set of landmark points reserving both geometric relationships (shape model) and color information (color model). Each patch with small color variations in the target pattern is denoted by one or several points, differing with Active Shape Model, ASM, and Active Appearance Model, AAM, which model the local boundary information of target patterns. With the information of global distributions, GSC model can be exploited to estimate the initial status of objects in target images in a more generalized way, for its consequent models, such as ASM and AAM. Low dimension representation and the flexible selection of transformation parameters lead to the rapid match to the complex models. The experiment results show the models are effective.

Keywords: Global Structure Constraint, Active Shape Model, Active Appearance Model, Object Localization.

1 Introduction

The interpretation of images of variable objects is an essential task for computer vision. To 'understand' the images is to recover image structure and know what it means. Model-based methods offering prior knowledge of problems can offer potential solutions to object alignment, recognition and synthesis [1]. Active shape models, ASM [2] and active appearance models, AAM [3], introduced by Cootes et al, both belong to the category of model-based methods.

ASM models the boundary shape of objects and the local grey-level appearance along profiles normal to the boundary and searches along profiles about the current model point positions to update the current estimate of the shape of the object. AAM, based on the shape model of ASM, warps the image patch enclosed by each training shape into a "shape free patch" enclosed by the mean shape to model the statistical texture of objects. Then, it samples the image data under the current instance and uses the difference between model and sample to update the appearance model parameters [4].

ASM only use data around the model points, and do not take advantage of all the grey-level information available across an object as the AAM does. AAM must generate a synthesized hypothesis with the current model parameters. Compared with AAM, ASM tends to have a larger capture range, runs

significantly faster, and achieves more accurate feature point location. The other way, AAM can build a convincing model with a relatively small number of landmarks, explicitly minimize texture errors to give a better match to the image texture and be more robust than ASM. However, both ASM and AAM need a reasonable initial starting approximation, cannot correct for large displacements from the correct position, and may either diverge to infinity, or converge to an incorrect solution [4] [5].

Many alternatives and extensions to ASM and AAM have been presented. Cootes et al. [6] describe variations of the basic AAM aimed at improving the speed and robustness of search. These include subsampling and using image residuals to drive the shape rather than full appearance model. Cristinacce et al. [7] present a Boosted Regression Active Shape Models, which use non-linear boosted features trained using GentleBoost [8] and investigate local feature detection using boosted features and also boosted regression. Hou et al. [1] propose direct appearance model (DAM) which uses texture information directly in the prediction of the shape and in the estimation of position and appearance, without combining from shape and texture as in AAM and prove that DAM subspaces includes admissible appearances previously unseen in AAM, the convergence and accuracy are improved and the memory requirement is cut down to a large extent. Many methods to optimize the parameters of these two models are presented [9]. In 3D, ASM are also suitable [10] [11] [12].

In this paper, we present a global structure constrained model (GSC), which examines the statistics of the positions and color information of the labeled landmark points under a "Point Distribution Model". Landmark points are labeled at main patches with small texture variations. Then, the pattern of the target objects can be represented with the shape and typical colors of the patches around landmarks, such as the grey-levels. GSC can use fewer landmarks to model target pattern than ASM and AAM. GSC retains the globally inherent constrained relation (both spatial and optical) of the target pattern. It can be used to search the candidate initial status in new images and supervise the improper convergence, such as in ASM and AAM, with low computational cost.

2 Global Structure Constraint Model

2.1 Patches and Landmarks

The patches with small texture variations can be segmented by statistical means. However, in practice we can line out the functional areas of the target object and then mark several key points on each area to depict its main color structure. As shown in Fig. 1, a human face is composed of seven areas, which are forehead, eyes, left cheek, right cheek, nose, mouth and jaw, and 28 landmark points are chosen totally.

2.2 Shape Model and Transformation Parameters

Given a training set consisting of m labeled 2-d images, in which n key landmark points $\{p_1, p_2, \dots, p_n\}$ are marked on each samples. $\mathbf{s} = \{x_1, \dots, x_n, y_1, \dots, y_n\}$

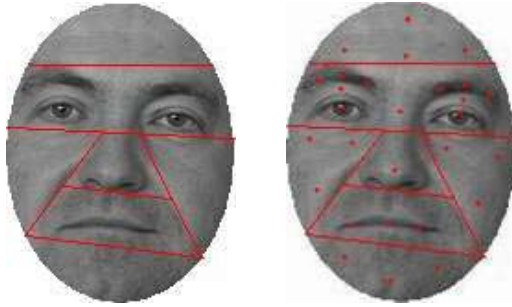


Fig. 1. Illustration of areas and landmarks

denotes the displacement of landmark points and describes the global distribution of that object. To build a statistical model of GSC, we normalize all the shape vectors of the training set to a 'model' coordinates independent to image coordinates. Then, the shape model of GSC, \mathbf{S}_{GSC} , can be designated as the mean shape vector $\bar{\mathbf{s}}$.

Given an arbitrary transformation matrix \mathbf{M} , the shape model \mathbf{S}_{GSC} can be transformed to the aimed coordinates to get a hypothesis sample \mathbf{S}_{sample} as

$$\mathbf{S}_{sample} = \mathbf{M}\mathbf{S}_{GSC} \quad (1)$$

The matrix \mathbf{M} can be composed of several kinds, which are translation, T_x and T_y , rotation, θ , scale, S_x and S_y . Most of the pixel-wise transformations can be used, such as shearing, projective and polynomial if necessary.

2.3 Color Information Model

The simplest way of building the color model is using the absolute color value, such as the grey-level or RGB. Given the set of landmark points on the i th sample image, the color information vector $\mathbf{c}_i = (c_{i1}, \dots, c_{in})^T$ is extracted by computing the representative color value around each landmark statistically. Then, we can normalize \mathbf{c}_i to reduce the effects of the changes while imaging by subscribing, $\mathbf{c}_i = \mathbf{c}_i - \frac{\sum_{j=1}^n c_{ij}}{n} \mathbf{I}$, or dividing, $\mathbf{c}_i = \frac{\mathbf{c}_i}{\frac{\sum_{j=1}^n c_{ij}}{n}}$, the average.

In the real world, there are still many circumstances which cannot be represented using only the color value. For example, cats are cats whatever colors they are and an image and its negative both illustrate the same contents. These issues can be recognized by human very easy. So, the relationships of the color values in the image are as important as the values. While building the color model, the number of the color categories N_{GSC} and the class tag for all the landmark points, $C_{GSC} = \{cl_j\}, cl_j \in [1, \dots, N_{GSC}]$, should be designated. At the same time, when an absolute color value must be satisfied at a specific landmark, it should be reserved as a part of the color model. Then, the color model of GSC can be described as $C_{GSC} = \{cl_j, ac_j\}, cl_j \in [1, \dots, N_{GSC}]$ in which ac_j is the absolute color value for the j th landmark.

2.4 Search and Measure

Given a group of transformation parameters, we can locate an instance shape, \mathbf{S}_{inst} , coming from the GSC model in the new image. The color information, \mathbf{C}_{inst} , at \mathbf{S}_{inst} in this image can be sampled with the same rule while the GSC model is built. Then, the problem to locate the best match between the model and the new image is converted into an optimization problem to minimize or maximize the measure function in the search space of the transformation parameters. The dimension of the parameter space is low enough intrinsically, and the direct optimization using many standard methods is possible.

We can cluster the color information \mathbf{C}_{inst} and reserve the absolute color information which is required in the color model $C_{inst} = \{cl_u, ac_u\}, cl_u \in [1, \dots, N_{inst}]$. The measure function can be defined using the mutual information in information theory [13].

$$f(C_{GSC}, C_{inst}) = \frac{1}{n} \sum_{k=1}^{N_{GSC}} \sum_{u=1}^{N_{inst}} n_{ku} \log \frac{n_{ku}n}{n_k n_u} \tag{2}$$

where n_k and n_u is the number of points in the k th class of C_{GSC} and the u th class of C_{inst} respectively. n_{ku} is the number of the common points in these two class.

If there are some specific landmarks with absolute color requirement as described above, another item should be added to the measure function.

$$f(C_{GSC}, C_{inst}) = \frac{k_1}{n} \sum_{k=1}^{N_{GSC}} \sum_{u=1}^{N_{inst}} n_{ku} \log \frac{n_{ku}n}{n_k n_u} + k_2 \sum_i \frac{|ac_{i,GSC} - ac_{i,inst}|}{ac_{Span}} \tag{3}$$

where $k_1 + k_2 = 1$ and ac_{Span} is the color span of the model.

3 Working Together with ASM and AAM

ASM and AAM are two kinds of object representation models, which have been applied in many fields of computer vision, such as image registration [14] and synthesis [15], segmentation [16] [11] [12], localization [17] [18] and recognition [19]. However, their essential precondition is a reasonable starting approximation. ASM is a kind of local model, which searches several pixels either side of each model point along the profile normal to the boundary. Compared with ASM, AAM can be categorized as a global model in spite of being based on the shape model of ASM. However, in each search step a synthetic example must be generated to be matched with the new image and this leads to a low search speed, such as an average of 4.1 seconds on a Sun Ultra to search one time [3]. This means that using AAM to search globally in a new image is unsuitable. The generalized model GSC, which converts the global searching in images into an optimization problem in the searching space with small dimension, can be used before ASM and AAM to search an initial status.

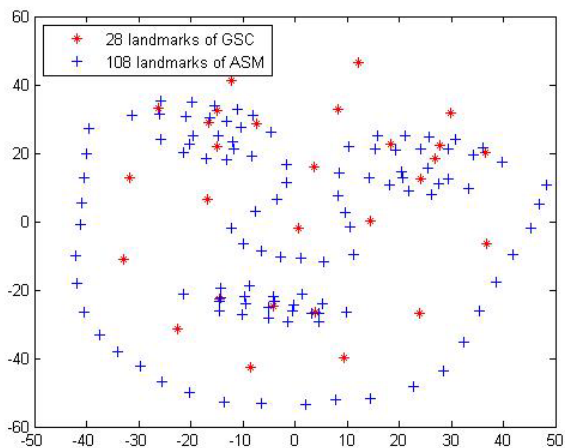


Fig. 2. Landmarks of the GSC and ASM models. 28 points marked with '*' and 108 with '+' are for GSC and ASM/AAM respectively.

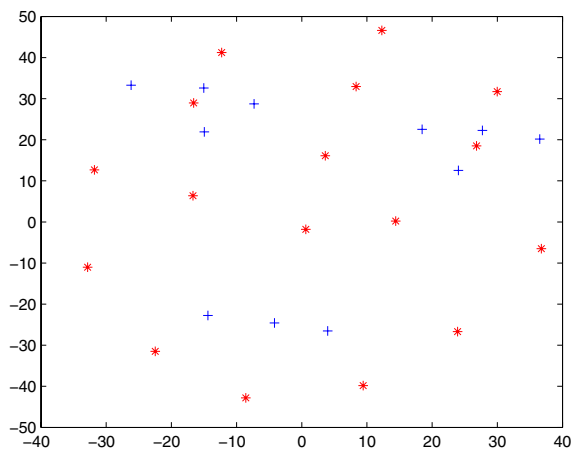
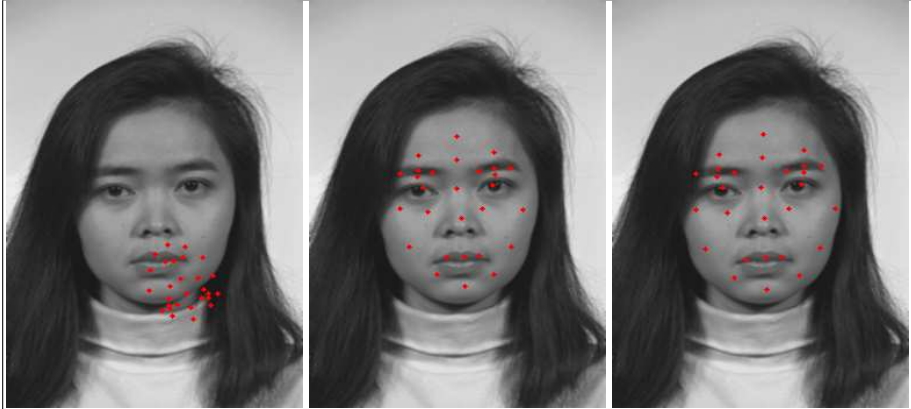


Fig. 3. Landmarks of GSC with different class tags. '+' and '*' are for the class with dark and light color respectively.

Table 1. The statistical errors of searching results with GSC model

Unit: Pixels	minimum	maximum	average
MPPE on X axis	2.372	19.632	10.5447
MPPE on Y axis	5.232	21.677	13.527
MPPE on $X + Y$ axis	8.202	40.508	24.072

Table 2. Some images with the searching results displayed

Iter. 10	Iter. 30	Iter. 50
		
Parameters:(119.706, 208.275, 2.075, 2.088, -1.819)		
		
Parameters:(128.875, 202.588, 1.588, 1.694, -1.835)		
		
Parameters:(116.955, 185.115, 1.692, 1.709, -1.817)		

The shape model of ASM/AAM focuses on the boundary of the target, where the texture (or intensity) changes strongly. In contrast, the shape model of GSC lays the landmarks at the patches with small texture variations. These two sets of landmarks are not equal, even have no common components. During the phase of building models, mark the landmarks of ASM/AAM and GSC in the training images together and transform them into the same model coordinates. Then, when the first step, searching for the initial status search using GSC, ends, the result of transformation parameters for GSC is the initial status of ASM/AAM.

4 Experiment Results

To obtain a quantitative evaluation of the performance we carried an experiment similar to [3], to train a model on 88 hand-labeled face images, and test it on a different set of 100 labeled images. Each image was about 256*384 pixels coming from the FERET database [20]. Fig. 2 shows the shape of GSC and ASM/AAM in an independent model coordinates. Fig. 3 shows the categories of the color information at all the landmarks marked for GSC. The K-Mean algorithm [21] was chosen to cluster the color information while searching in the new image.

On each test image, we use the GSC model to locate the initial status and AAM to run a local multi-resolution search, starting with the mean appearance model. Generic algorithm is adopted for searching with the population size 50, the probability of crossover 0.7 and mutation 0.3. The dimension of parameter spaces is five, with translation and scale in X and Y and rotation,

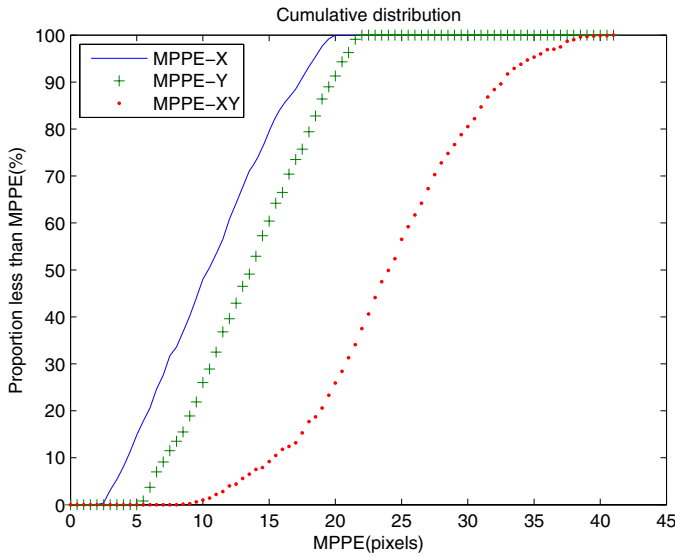


Fig. 4. The proportion less than different MPPE. ‘.’, ‘.’ and ‘+’ depict X , Y and $X+Y$ respectively.

$(T_x, T_y, S_x, S_y, \theta)$. Also, 1000 searches were run in total and the pre-location of GSC took an average of 0.6 second on a 2.3GHz PC. The convergence results are compared with the labeled positions to calculate the mean point position error (MPPE) per point. In Table 1 are the minimum, maximum and average, along X , Y and $X + Y$ axis of all the 1000 times searches respectively. If the search results lay right on the exact position of the labeled, the MPPE of X , Y and $X + Y$ will be zero. Fig. 4 shows the proportion of the searching results less than each MPPE. Table 2 shows some results images and parameters of GSC.

The local search using AAM is run beginning at the initial status coming from the search result using the GSC model. Of those 1000 times of searches, 203 (20.3%) failed to converge to a satisfactory result (the average point position error was greater than 7.5 pixels per point). This is much smaller than the results 519 from 2700 (19%) reported in [3], in which the starting positions are systematically displaced from the true position by ± 15 pixels in x and y , and changed its scale by $\pm 10\%$.

5 Discussion and Conclusion

In this paper, we present a kind of generalized point distribution model, called GSC. Using this model, we can localize the approximate position of the target in the new images. Compared with some other methods, GSC has its advantages.

Generalized. A generalized method which can be used in many applications.

Simple. Through this model, the complicated problem, image localization, can be converted into an optimization problem and the dimension of the searching space is small usually.

Fast. Searching in the space with a small dimension will not consume long time.

Also, this kind of method can work as a precondition process for some methods aimed for registration mainly, such as ASM and AAM.

References

1. Hou, X.W., Li, S.Z., Zhang, H.J., Cheng, Q.S.: Direct Appearance Models. In: Computer Vision and Pattern Recognition, vol. 1, pp. 828–833. IEEE Press, Hawaii (2001)
2. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active Shape Models - Their Training and Application. *Computer Vision and Image Understanding* 61(1), 38–59 (1995)
3. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active Appearance Models. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, pp. 484–498. Springer, Heidelberg (1998)
4. Cootes, T.F., Edwards, G., Taylor, C.J.: Comparing Active Shape Models with Active Appearance Models. In: 10th British Machine Vision Conference, vol. 1, pp. 173–182. BMVA Press, Nottingham (1999)

5. Cootes, T.F., Taylor, C.J.: Statistical Models of Appearance for Medical Image Analysis and Computer Vision. In: Proceedings of SPIE - Medical Imaging, San Diego, pp. 236–248 (2001)
6. Cootes, T.F., Edwards, G., Taylor, C.J.: A Comparative Evaluation of Active Appearance Model Algorithms. In: 9th British Machine Vision Conference, Southampton, vol. 2, pp. 680–689 (1998)
7. Cristinacce, D., Cootes, T.F.: Boosted Regression Active Shape Models. In: 18th British Machine Vision Conference, Coventry, vol. 2, pp. 880–889 (2007)
8. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. *The Annals of Statistics* 28, 337–407 (2000)
9. Li, Y.Z.: Shape Parameter Optimization for Adaboosted Active Shape Model. In: 10th IEEE International Conference on Computer Vision, Beijing, vol. 1, pp. 251–258 (2005)
10. Heimann, T., Wolf, I., Williams, T., Meinzer, H.-P.: 3D Active Shape Models Using Gradient Descent Optimization of Description Length. In: Christensen, G.E., Sonka, M. (eds.) IPMI 2005. LNCS, vol. 3565, pp. 566–577. Springer, Heidelberg (2005)
11. Rousson, M., Paragios, N., Deriche, R.: Implicit Active Shape Models for 3D Segmentation in MR Imaging. In: 7th International Conference on Medical Image Computing and Computer Assisted Intervention, Saint-Malo, pp. 209–216 (2004)
12. Hansegård, J., Orderud, F., Rabben, S.I.: Real-time active shape models for segmentation of 3D cardiac ultrasound. In: Kropatsch, W.G., Kampel, M., Hanbury, A. (eds.) CAIP 2007. LNCS, vol. 4673, pp. 157–164. Springer, Heidelberg (2007)
13. Cover, T.M., Thomas, J.A.: Elements of Information Theory. John Wiley & Sons Inc., Hoboken (1991)
14. Cootes, T.F., Hill, A., Taylor, C.J., Haslam, J.: Use of Active Shape Models for Locating Structures in Medical Imaging. *Image Vision and Computing* 12, 355–366 (1994)
15. Bettinger, F., Cootes, T.F.: A Model of Facial Behavior. In: International Conference on Face and Gesture Recognition, pp. 123–128 (2004)
16. Duta, N., Sonka, M.: Segmentation and Interpretation of MR Brain Images Using an Improved Knowledge-based Active Shape Model. In: Duncan, J.S., Gindi, G. (eds.) IPMI 1997. LNCS, vol. 1230, pp. 375–380. Springer, Heidelberg (1997)
17. Gui, Z.H., Zhang, C.: Robust active shape model construction and fitting for facial feature localization. In: Kanade, T., Jain, A., Ratha, N.K. (eds.) AVBPA 2005. LNCS, vol. 3546, pp. 1029–1038. Springer, Heidelberg (2005)
18. Beichel, R.R., Bischof, H., Leberl, F., Sonka, M.: Robust active appearance model matching. In: Christensen, G.E., Sonka, M. (eds.) IPMI 2005. LNCS, vol. 3565, pp. 114–125. Springer, Heidelberg (2005)
19. Edwards, G.J., Cootes, T.F., Taylor, C.J.: Face Recognition Using Active Appearance Models. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, pp. 581–595. Springer, Heidelberg (1998)
20. Phillips, P.J., Wechsler, H., Huang, J., Rauss, P.: The FERET database and evaluation procedure for face recognition algorithms. *Image and Vision Computing* 16(5), 295–306 (1998)
21. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297. University of California Press, Berkeley (1967)

Bio-inspired Architecture for Visual Recognition of Humans Walking

Pedro Luis Sánchez Orellana¹, Claudio Castellanos Sánchez¹,
Edgar del Angel-Guerrero², and Tomás Martínez-Arenas²

¹ Laboratory of Information Technology of the Centre for Research and Advanced Studies, LIT Cinvestav - Tamaulipas, Ciudad Victoria, Tamaulipas, México
{psanchez, castellanos}@tamps.cinvestav.mx

² Superior Technological Institute of Tantoyuca, ITSTA-Veracruz

Abstract. In this paper we propose a bio-inspired architecture for visual recognition of humans at walking and objects that can be humans but do not describe a gait like humans at walking, based on the behaviour of simple cells in the human primary visual cortex. This architecture was tested with real sequences of images acquired in natural environments. The results show the flexibility of our propose since it helps to distinguish between these two types of moving objects, even in unknown scene conditions (bright, or background motion).

1 Introduction

The visual recognition of human/non-human while moving is a complicated task that involves several sub-problems, the main one is the detection of the objects in motion in unknown environments. In this context several methodologies [2,3,13] have been proposed in the computer vision area to overcome the problem, often without very good results when the problem is taken to real life conditions. The reason is that these methodologies often make assumptions about the structure of the object.

More recently a different approach to this problem has come to light. It consist in modelling the ability of the human brain to recognise and analyze objects in motion [7], independent of the composition or structure of the pedestrian parts and the complexity of the environment. Even though the details of how this processing is achieved is not very clear some clues about it has been found. For example, it is known that the capability of our brain to visually recognise moving objects relies on the visual cortex, which is divided in two pathways, the dorsal pathway (primary visual cortex -V1-, middle temporal -MT-, middle superior temporal -MST-, etc.) specialises on the detection of motion [15,6] and the ventral pathway (V1, V2, V3, infero-temporal -IT-, etc.) which processes characteristics related to the form of the visual information. The available information about the functioning of these areas in the brain has been taken to create the so called bio-inspired algorithms, where the main goal is to take advantage of the capability of the human brain to recognise objects in real life scenes, and isolate them from other moving ones to analyze them.

In this work we will use the bio-inspiration to detect the difference between humans from non-humans in motion. So we will first review some related works on the bio-inspiration, then we will mention some biological foundations for this problem. Next we will introduce our architecture for the detection of objects in motion, based on the primary visual cortex of the human brain, to then analyze the neuronal activation patterns to finally discuss the results and our conclusions.

Also, it is important to mention that in this work we focus on the objective of distinguishing between humans walking motion, and non human objects or even humans but that are moving on other type of vehicle like bicycle or on a skateboard. So from here on we will refer to these two classes of objects just like humans/non-humans objects in motion.

2 Related Works

An important amount of effort has been put on the creation of bio-inspired models that not only achieve good results for the recognition tasks, but also represents more accurately the way our brain realises the mentioned task. From this investigation on bio-inspiration different points of view have been developed, basically differentiated by the way they conceive the processing in both pathways: dorsal and ventral.

Some researchers have focused on finding the optimal set of characteristics and the type of motion pattern necessary for recognising a person. For example, Laxmin et al [9] present a neural network model based on the capabilities of the human beings to recognise motion of other human beings based on only partial information (cluttered information or sketched with light points). Lange et al [7] presented experiments that showed the importance of considering both the spatial and temporal structure of the shape for the recognition process, it means, it is important to keep on with a specific motion pattern.

Finally, others focused their effort on modelling the real functioning of the processing of the information in both pathways, for example, Giese and Poggio [5] consider general biological models where the brain activity is represented by a continuous scalar variable which is a valid assumption at this level. This last set of researchers approach more to what we could expect from a bio-inspired model since we are focused more on the internal representation (specifically in the intermediate/higher stages of processing) of the objects. However, this works have not considered the primary stages (V1) of processing more than a black box, where the information extracted from them is used to feed the more specialised areas (MT, MST, IT, etc).

In this context Castellanos [1] used the known information in V1 and all dorsal pathway to create a model (called CONEPVIM) to estimate the movement of objects from sequences of images. This model is composed by two stages, the first one models the simple cells in V1 to extract the spatial-temporal characteristics of the images. The responses of the first stage feeds the second one, that considers the neurons in MT and MST to estimate velocity, direction and whether the motion in the scene was caused by the motion of the observer or by motion of objects.

For our architecture we work with the methodology of Castellanos, by using the first stage of his model to extract the objects in motion in a scene. However we will try to discover what is the best representation for modelling the pedestrian detection? and what really happens in the intermediate stages in the brain?, in other words, how does the brain complete the information in each stage? For this we will start describing some biological foundation for our proposed bio-inspired architecture in the behaviour of the simple cells in V1.

3 Biological Foundations

Recent research on computational neuroscience has provided an improved understanding of human brain functionality and bio-inspired models have been proposed to mimic the computational abilities of the brain for motion perception and understanding. For example, there are some research inspired on the functioning of (V1) with a strong neural cooperative-competitive interactions that converge to a local, distributed and oriented auto-organization [4,8,11]. Some others are inspired by the middle temporal area (MT) with the cooperative-competitive interactions between V1 and MT and an influence range [16,10]. And the others are inspired by the middle superior area (MST) for the coherent motion and egomotion [14,17].

In this work, our analysis will be focused on the responses of the neurons in V1, modelled by means of a causal spatial-temporal Gabor-like filters. The filter has the capability to extract the motion information in a local and distributed form, by measuring the changes through the images sequence. This capability of the filter can be used to characterize the motion of different objects (articulated/non-articulated ones), by measuring the global activations of neurons. The basis for this last asseveration is that different objects have different motion composition patterns, so an articulated object produces different global neurons activations compared to the responses to a non articulated ones.

To explain the last idea we take into account the work realised by Murray [12], where the composition of the human walking was described: there are several pieces that form the stepping of the humans. For instance, one whole step is composed by three major movements, one forward left step, forward right step and finally another forward left step, but the interesting thing comes in the middle of this movements, while one foot is moving the other one remains static. This type of motion in humans, in terms of the responses of the neurons, produces a different patterns of activations which are unique for the articulated objects. Mainly because the non articulated ones never change their structure while moving. To try to solve the articulated/non-articulated objects differentiation problem we propose an architecture (described in the following section), that takes into account the foundations mentioned in the last paragraph.

4 Architecture

The proposed architecture for visual detection of moving objects is divided basically into three stages (see figure 1). The first stage is the spatial treatment,

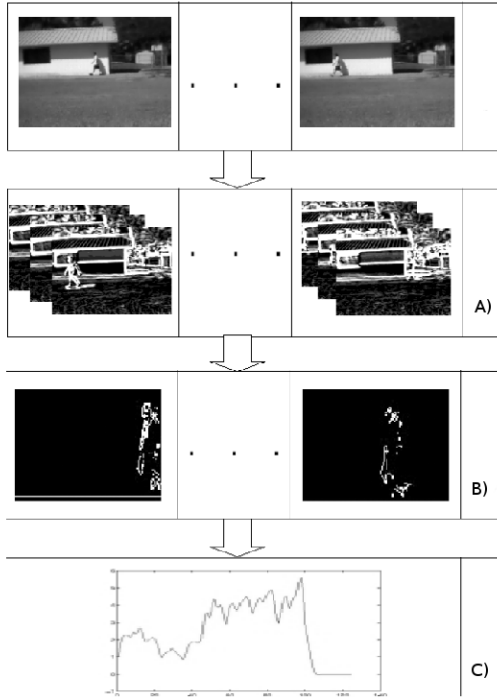


Fig. 1. Proposed architecture : (A) Spatial treatment, (B) temporal treatment using a WTA architecture, and (C) analysis of the responses

a convolution with our Gabor-like oriented filters. Next, a temporal treatment by computing the difference between the spatial responses and combining the temporal responses into set of maximum responses by means of a Winner Takes All strategy (WTA). This is because our objective is to keep on with the higher responses to a specific condition in the images. Besides this reduces our problem dimensionality from a eight-dimensional set of responses for each image in the sequence to a one-dimensional set of responses. Finally, this resultant reduced set is taken to measure, in ranked way, the responses of the neurons in V1, by considering that the responses are related not only to the portion of the objects in motion during the sequence, but also to the periodicity of this responses.

It is important to mention that the used images were taken in outdoor environments, with uncontrolled conditions (neither illumination, background motion nor the automatic contrast adjustment of the camera), and they were converted from RGB format to grey scale.

4.1 First Stage (A)

Our architecture begins by applying our Gabor-like oriented filters that modelled the responses of the simple cells in V1, this filtering ensures the capability to detect the local motion in a simple and local way, defined as follows.

Let $I(x, y, t)$ be an image sequence representing the shape of intensity in the time-varying image, assuming that every point has an invariant brightness. By applying an oriented Gabor filter, $G_\theta(x, y)$, we obtain:

$$\begin{aligned}
 D_\theta(t) &= \int \int_{t=0} \frac{dI(x, y, t)}{dt} * G_\theta(\hat{x}, \hat{y}) \, dx \, dy \\
 &= \frac{d \int \int_{t=0} I(x, y, t) * G_\theta(\hat{x}, \hat{y}) \, dx \, dy}{dt}
 \end{aligned}
 \tag{1}$$

where D is the result of the convolution between the Gabor functions and the image, $*$ is the convolution function, \hat{x} , \hat{y} the rotational values and $G_\theta(\hat{x}, \hat{y})$ is computed in a standard way:

$$G_\theta(\hat{x}, \hat{y}) = \frac{1}{2\pi\sigma_x\sigma_y} e^{\left(-\frac{x^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2}\right)} e^{(2\pi i \frac{\hat{x}}{\lambda} + \phi)}
 \tag{2}$$

This is our Gabor-like filter model the simple cells in V1 where γ is the eccentricity of the receptive field, $\sigma_x\sigma_y$ its dimensions, λ the wavelength and ϕ the phase. For simplicity $0 \leq \theta = n\pi/4 < 2 \cdot \pi$ and the other parameters in the filter were set by experimentation and considering the parameters described by Castellanos [1]. The result of this stage is a set of oriented responses $D_\theta(t)$ which contain the preferred responses of the simple cells in V1. These responses are used to estimate motion in the sequences by using a temporal treatment and a WTA strategy as it is shown in the following subsection.

4.2 Second Stage (B)

In order to integrate the responses from $D_\theta(t)$ that are considered as a good responses we established a threshold . This limit serves for two objectives, the first one is to establish a WTA tournament, where the responses in $D_\theta(t)$ compete to determine which of them is the maximum response for a specific position $d_\theta(x, y, t)$ in the D. The second objective is to leave aside the errors obtained in the capture stage such as automatic contrast or auto focus characteristics. For this first the θ responses compete to establish which of the (x, y, t) has the highest response as follows

$$E(x, y, t) = \max_\theta(d_\theta(x, y, t))
 \tag{3}$$

These results in $E(x, y, t)$ have been estimated for the same time in the sequences but with all the θ responses obtained in the spatial treatment, so the next step is to compare the responses for all t images, to measure the changes. For this comparison we use a threshold (ϵ) to determine which of the responses are strong enough to produce a movement rather than by an error in the capturing stage. This is estimated as follows if $F = |E(x, y, t) - E(x, y, t - 1)|$:

$$H(x, y, t) = \begin{cases} F & \text{if } F \geq \epsilon \\ 0 & \text{otherwise} \end{cases}
 \tag{4}$$

with ε as the threshold. The final stage is to analyze the responses obtained by the temporal treatment to determine the behaviour of the responses.

4.3 Third Stage (C)

Since we are searching for different patterns of activations of the neurons, depending on the type of movement that is presented, our theory is considering a way an articulated object moves (might be human being or an animal) it is possible to discriminate it from other type of moving objects that are not articulated (vehicles). For this we counted the responses and estimated the number of activation of the cells in $H(t)$ by splitting them into 40 parts of responses, this value was determined empirically, and the number of responses was multiplied by the information that it represented. This was to rate the information, since not all the responses can be treated in the same way, in fact, it depends on the amount of activations rather than the magnitude of them. Besides it was divided into the proportion of the maximum object that for us is possible to recognise, lets say that a human can occupy a maximum area in the image of 33.3% of the total space, for a bigger value we would not be able to recognise the body because of cluttering. The function of this maximum size of the object is to show a proportion of activations in the neurons based on the maximum expected value as follow:

$$winner(t) = \frac{\sum_{i=0}^n H_i(x, y, t) P_i}{Q} \quad (5)$$

where $H(t)$ is the set of the active neurons in the i -esim percentile, P_i is the percentile, and Q is the maximum supposed size of an object in the image I . This last stage allow us to measure the behaviour of neuron responses when exposed to different motion patterns.

5 Results

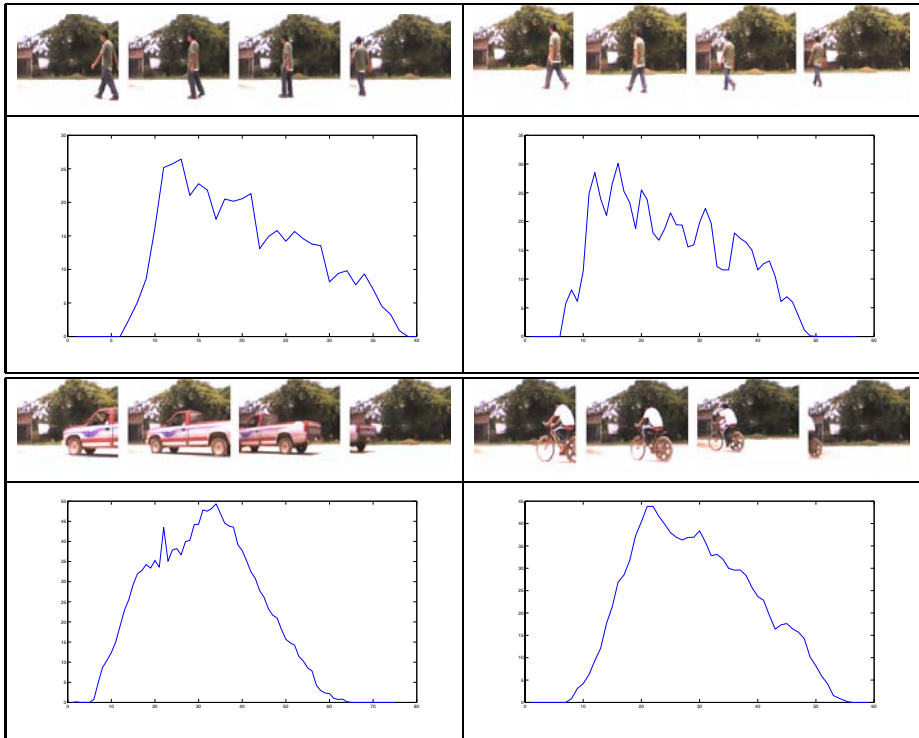
For the experiments we used several sequences of real images taken with a webcam in an outside environment, to test the methodology even with the lowest conditions for the detection, mainly due to uncontrolled illumination conditions and the function of self-adjustment of contrast in the webcam. The images were taken at a very low rate, 15 fps, which decreases the accuracy of the measurements.

To test the architecture we decided to record a single object per sequence in order to prove that global activation patterns in V1 neurons in fact describe the type of motion of an object. The first step is to compare the responses that are obtained with our architecture with sequences from articulated objects (humans) and non-articulated objects (cars or human skating). So we separated the experimentations into two main categories: articulated objects and non-articulated ones, the sub categories are described in the table [II](#).

Table 1. Categories for objects recorded in the sequences

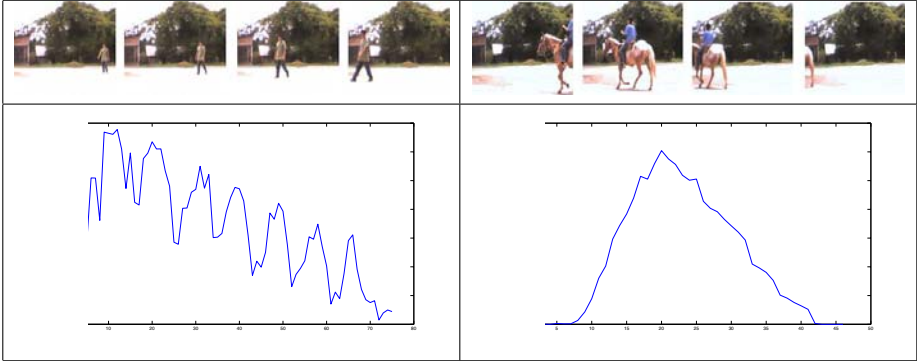
Category	Sub category	Motion angle in the sequences
Articulated	Healthy person	0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°
	Unhealthy person	
	Horses	
Non-Articulated	Person skating	0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°
	Person on a bicycle	
	Vehicles	

Table 2. Table Neuronal activation patterns with sequences of articulated objects. In the graphs the *X – axis* represent the magnitude of the global responses and *Y – axis* represent the time for the sequence. Sequences taken at 45° of orientation with respect to the camera.



Since there are many objects that have articulated locomotion we separated the objects in two main categories to know whether our methodology is capable to distinguish between the objects that are not articulated. So by analysing the results we could confirm our theory about different activation patterns in neurons. The results of the experiments with sequences of articulated/non-articulated objects can be seen in the figures shown in the table 2, where we show different patterns of neurons activation that are different in the case of the

Table 3. Table Neuronal activation patterns with sequences of articulated objects. In the graphs the $X - axis$ represent the magnitude of the global responses and $Y - axis$ represent the time for the sequence. Sequences taken at 45° of orientation with respect to the camera.



articulated objects set, also it can be seen that the activation patterns in the case of sequences of humans are different compared to the ones produced with sequence of a horse.

Besides, in the case of rigid objects like a truck, the bicycle or the skateboard the responses produced are more uniform. In the case of the human, it can be seen that the activations of the neurons along the time ($Xaxis$) are not in a uniform growing-up-style, which is caused by multiple moving articulations at different order. Also it can be explained by comparing the proportion of extremities in the human to the ones in the horse; the legs and arms in the case of the human are bigger compared to the horse in proportion with the body.

If we compare the global activation values ($Yaxis$) between human beings and non-human beings (see figures in table 3), we can see that the responses of non-human objects grow more than uniformly that the responses of human beings. This is due to the fact that some extremities of the articulated bodies move while the others remain static.

6 Conclusions and Future Work

Summarising, we consider important to mention that our proposed architecture is tolerant to the changes in contrast and also to the motion in the background. It is completely based in the responses of simple cells in V1. Besides the methodology was designed in such a way that neither the estimation of the *winner* nor the slants depends on a specific size of image. However, despite of this flexibility not all the movements can be analysed, it also depends on the resolution and size of the object. However, the responses shown in graphs in table 3 allowed us to conclude that neuronal activations patterns describe different behaviours, which correspond to our theory mentioned in the section of Biological foundations. What resulted more interesting is that we can differentiate even between

objects that have an articulated locomotion. In spite of these results we also consider that it is important to keep on working with the methodology to avoid the dependence on the global responses, so it will be possible not only to recognise a single person in the scene, but also to independently track objects in motion.

Acknowledgment

This research was partially funded by project number 51623 from “Fondo Mixto Conacyt-Gobierno del Estado de Tamaulipas” and the project number 78885 from “Ciencia Básica 2007” of Conacyt.

References

1. Sánchez, C.C.: Neuromimetic indicators for visual perception of motion. In: 2nd International Symposium on Brain, Vision and Artificial Intelligence, vol. 103, pp. 134–143 (2007)
2. Gross, R., Collins, R., Shi, J.: Silhouette-based human identification from body shape and gait. In: Int. Conference on Face and Gesture Recognition, pp. 351–356 (2002)
3. Cunado, D., Nixon, M., Carter, J.: Automatic extraction and description of human gait models for recognition purposes. *Comput. Vis. Image Underst.* 90, 1–41 (2003)
4. Fellez, W.A., Taylor, J.G.: Establishing retinotopy by lateral-inhibition type homogeneous neural fields. *Neurocomputing* 48, 313–322 (2002)
5. Giese, M., Poggio, T.: Neural mechanisms for the recognition of biological movements and acti on. *Nature Reviews Neuroscience* 4, 179–192 (2003)
6. Grossman, E., Donnelly, M., Price, R., Pickens, D., Morgan, V., Neighbor, G., Blake, R.: Brain areas involved in perception of biological motion. *J. Cognitive Neuroscience* 12(5), 711–720 (2000)
7. Lappe, M., Lange, J.: The role of spatial and temporal information in biological motion perception. *Advances in Cognitive Psychology* 3, 419–428 (2007)
8. Latham, P.E., Nirenberg, S.: Computing and stability in cortical networks. *Neural Computation*, 1385–1412 (2004)
9. Laxmi, V., Carter, J.N., Damper, R.I.: Biologically-inspired human motion detection. In: 10th European Symposium on Artificial Neural Networks, pp. 95–100 (2002)
10. Mingolla, E.: Neural models of motion integration and segmentation. *Neural Networks* 16, 939–945 (2003)
11. Moga, S.: Apprendre par imitation: une nouvelle voie d’apprentissage pour les robots autonomes. PhD thesis, Université de Cergy-Pontoise, Cergy-Pontoise, France (September 2000)
12. Murray, M.: Gait as a total pattern of movement. *American Journal in Physics and Medicine* 46, 290–332 (1967)
13. Nikolaos, V., Zhiwei, X.: Gait recognition using radon transform and linear discriminant analysis. *IEEE Image Processing* 16, 731–740 (2007)
14. Pack, C., Grossberg, S., Mingolla, E.: A neural model of smooth pursuit control and motion perception by cortical area mst. Technical Report CAS/CNR-TR-99-023, Boston University, Department of Cognitive and Neural Systems and Center for Adaptive Systems, 677 Beacon St, Boston, MA 02215 (September 2000)

15. Serre, T., Kouh, M., Cadieu, C., Knoblich, U., Kreiman, G., Poggio, T.: A theory of object recognition: Computations and circuits in the feedforward path of the ventral stream in primate visual cortex. Technical report, Massachusetts Institute of Technology (2005)
16. Simoncelli, E.P., Heeger, D.J.: A model of neural responses in visual area mt. *Vision Research* 38(5), 743–761 (1998)
17. Zemel, R.S., Sejnowski, T.J.: A model for encoding multiple object motions and self-motion in area mst of primate visual cortex. *The Journal of Neurosciences* 18(1), 531–547 (1998)

Computational Model for Electric Fault Diagnosis in Induction Motors

Rodrigo López-Cárdenas¹, Luis Pastor Sánchez-Fernández²,
and Sergio Suárez-Guerra²

¹ Higher School of Mechanical and Electrical Engineering

² Centre for Computer Research

National Polytechnic Institute, Mexico City, 07738 Mexico

Abstract. This article describes a novel *computational model* for electric fault diagnostic in induction motors. The essential concept is that a minimum electric fault, like inter-turn short circuit, produces a slight variation that can be identified in current and rotor speed signals. This model uses motor data catalogue to calculate constant parameters that are handled in an original mathematical algorithm that employs varying parameters as function of motor slip. The model performs electric fault simulation and with them, are obtained operation characteristics that build relative and absolute patterns for normal and fault operation. These patterns train a neural network that accomplish the diagnostic in its phase implementation.

Index terms: Computational model, fault diagnosis, induction motors.

1 Introduction

Three-phase squirrel cage induction motors are essential components in most of today's manufacturing and production industries. Safety, reliability, efficiency and performance are some of the major concerns and needs for motor system applications [1] [2], the goal of this research is describe an original computational model for electric fault diagnostic in induction motors.

2 Computational Model

The Fig 1 shows the computational model for its training and execution phases used in this paper to diagnose electric faults in induction motors.

2.1 Training Phase

Mathematical models of induction motors like [3] use construction motor parameters, that only are accessible to manufacturers or extracted from theory literature. Simulate squirrel cage induction motors with these parameters implies to use the rotor resistance and motor inductances like a constants, assuming that

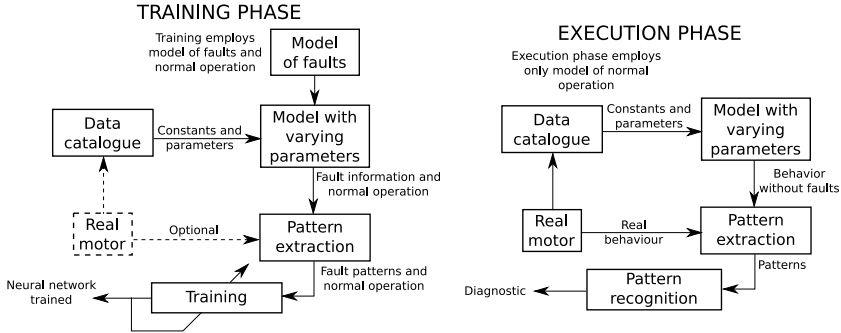


Fig. 1. Computational model block diagram for electric faults diagnostic in training and executions phases

squirrel cage motor maintains these values without change during all start-up process, ignoring that rotor resistance vary with time as function of motor slip [4] [5]. This is the reason that is used a novel mathematical model that varies its parameters as function of rotor slip. From the Fig.1, the block named “Model with varying parameters” takes as input the information from “Data catalogue” block, this block use a original method to calculate motor parameters from its nominal values [6].

This mathematical model simulate in “Model of faults” block, among with normal operation, electric faults like stator short circuit, rotor broken bar and unbalanced input voltage, with four intensities for each one. Although exist methods that can diagnose electric faults [7] [8], all of them use a physical motor to extract its parameters, and therefore, their methodology is difficult to generalize. By using nominal information, from “Data catalogue” block, was possible to build a data base of motors between 1kW to 50kW with normal and faults conditions.

Information from real motors with know faults or exclusively from simulation are used to calculate operational characteristics that establish relative patterns to normal operation and also absolute patterns that are extracted from motor specification, the block responsible is “Pattern extraction”. These patterns are used to train an Artificial Neural Network to diagnose the normal or fault motor operation in “Training” block.

2.2 Execution Phase

The motor nominal information under diagnostic is used for obtain its internal parameters by “Data catalogue” block and trough the varying parameter model simulate its normal behaviour and compare it with its real behaviour in the “Pattern extraction” block. The relative patterns to normal operation among with absolute patterns are the input to “Pattern recognition” block (trained in above phase), and the output delivery the diagnostic, that could be normal condition or some of the three faults cited above with four intensities each one, that is, one of the thirteen possible options.

In the subsequent section the specific functions of each block are explained in detail.

3 Mathematical Model of Induction Motor

This section describes the foundations of motor mathematical model, the method used to calculate the varying parameters with respect to motor slip and, the faults injection used in motor simulation. These mathematical developments are utilized in “Model with varying parameters” and “Data catalogue” blocks at training and execution phase, and “Model of faults” block at training phase.

The following equations have been transformed from tri-phase system to bi-phase system, $abc - \alpha\beta$, in order to simplify mathematical operations [9]. The stator equations as function of flux linkages are:

$$u_{s\alpha} = \frac{R_s}{L_{\sigma R}}\varphi_{s\alpha} - \frac{R_s}{L_{\sigma M}}\varphi_{r\alpha} + \frac{d\varphi_{s\alpha}}{dt} \tag{1}$$

$$u_{s\beta} = \frac{R_s}{L_{\sigma R}}\varphi_{s\beta} - \frac{R_s}{L_{\sigma M}}\varphi_{r\beta} + \frac{d\varphi_{s\beta}}{dt} \tag{2}$$

since motor is squirrel cage, then rotor voltage is zero:

$$0 = -\frac{R_r}{L_{\sigma M}}\varphi_{s\alpha} + \frac{R_r}{L_{\sigma S}}\varphi_{r\alpha} + \frac{d\varphi_{r\alpha}}{dt} + \frac{2}{P}\omega_r\varphi_{r\beta} \tag{3}$$

$$0 = -\frac{R_r}{L_{\sigma M}}\varphi_{s\beta} + \frac{R_r}{L_{\sigma S}}\varphi_{r\beta} + \frac{d\varphi_{r\beta}}{dt} + \frac{2}{P}\omega_r\varphi_{r\alpha} \tag{4}$$

the mechanical equation is:

$$\frac{3}{2} \frac{P}{2} (\varphi_{s\alpha}i_{s\beta} - \varphi_{s\beta}i_{s\alpha}) = J \frac{2}{P} \frac{d\omega_r}{dt} + m_{mec}(\omega_r) \tag{5}$$

In these equations $u_{s\alpha}$ and $u_{s\beta}$ are transformed input voltages, R_s and R_r are stator and rotor resistances respectively, ω_r is rotor speed, P is poles number, J is motor inertia, m_{mec} is the mechanical moment, φ_{sx} and φ_{rx} are stator and rotor flux linkages and, $L_{\sigma s}$, $L_{\sigma r}$ and $L_{\sigma M}$ are coupling inductances obtained from total dispersion inductance.

3.1 Novel Mathematical Model

Equations (1) to (5) suppose constant the resistive and inductive parameters from model, its not really so, rotor resistance and dispersion reactance depend strongly from motor slip, thus, is necessary a new model that includes the variation of parameters as function of motor slip. The following equations represents a mathematical model, expressed in state space, of squirrel cage induction motor that involves rotor resistance variation and inductances [10]. Not confuse the subindex in the stator parameter (R_s) with variable s , associated to rotor resistance, that denotes motor slip.

$$\dot{\varphi}_{s\alpha} = u_{s\alpha} - \frac{R_s}{L_{\sigma R}}\varphi_{s\alpha} + \frac{R_s}{L_{\sigma M}}\varphi_{r\alpha} \tag{6}$$

$$\dot{\varphi}_{s\beta} = u_{s\beta} - \frac{R_s}{L_{\sigma R}}\varphi_{s\beta} + \frac{R_s}{L_{\sigma M}}\varphi_{r\beta} \tag{7}$$

$$\dot{\varphi}_{r\alpha} = \frac{R_r(s)}{L_{\sigma M}}\varphi_{s\alpha} - \frac{R_r(s)}{L_{\sigma S}}\varphi_{r\alpha} - \frac{2}{P}\omega_r\varphi_{r\beta} \tag{8}$$

$$\dot{\varphi}_{r\beta} = \frac{R_r(s)}{L_{\sigma M}}\varphi_{r\beta} - \frac{2}{P}\omega_r\varphi_{r\alpha} - \frac{R_r(s)}{L_{\sigma S}}\varphi_{r\beta} \tag{9}$$

$$\begin{aligned} \dot{\omega}_r = & -m_{mec} + \frac{3P^2}{8J} \left[\varphi_{s\alpha} \left(\frac{1}{L_{\sigma R}}\varphi_{s\beta} - \frac{1}{L_{\sigma M}}\varphi_{r\beta} \right) \right. \\ & \left. - \varphi_{s\beta} \left(\frac{1}{L_{\sigma R}}\varphi_{s\alpha} - \frac{1}{L_{\sigma M}}\varphi_{r\alpha} \right) \right] \end{aligned} \tag{10}$$

3.2 Calculation of Varying Parameters for Mathematical Model Proposed

Information necessary to calculate parameters present in motor brochure are: rated power P_n , rated speed n_n , rated torque M_n , rated voltage U_n , pole number p , full load current I_n , rated frequency f_n , power factor f_{pn} , locked rotor current I_a , locked rotor torque M_a , breakdown torque M_{max} and moment of inertia J_n . From this data are obtained constant resistance K_1 and K_2 that are used to calculate varying rotor resistance, constant inductance K_3 and K_4 are used to calculate dispersion and magnetizing reactances [6].

The ratio between motor synchronous speed and rotor speed is commonly named motor slip denoted as s , and is expressed as a fraction of synchronous speed: $s = \frac{n_s - n_n}{n_s}$, where $n_s = \frac{120f_n}{p}$ is the synchronous speed. Hence, the motor slip can take values from one at machine start-up (in that moment rotor speed is zero) to decimals at rated speed. During this process, rotor resistance and inductances vary with respect motor slip.

In the rotor occurs a physical phenomenon called “skin effect” that only happens in rotor bars but not in rings, therefore, rotor resistance expression has two components: a constant that corresponds to rings and another, function of slip, that corresponds to bars. According to [5], this component is proportional to square root of frequency motor, and therefore, proportional to square root of slip:

$$R_r(s) = K_1 + K_2\sqrt{s} \tag{11}$$

the skin effect influences total dispersion reactance too, X_{cc} (that is, total dispersion inductance multiplied by $2\pi f_n$) but in counter way:

$$X_{cc} = K_3 + \frac{K_4}{\sqrt{s}} \tag{12}$$

to calculate constant resistance is necessary determine rotor resistance at nominal condition R_{rn} and at start-up R_{rk} , then, constant K_1 and K_2 are obtained from:

$$K_1 = \frac{R_{rn} - \sqrt{s_n}R_{rk}}{1 - \sqrt{s_n}} \quad ; \quad K_2 = \frac{R_{rk} - R_{rn}}{1 - \sqrt{s_n}} \quad (13)$$

where the rotor resistance at different operation conditions are obtained from:

$$R_{rn} = \frac{s_n P_n}{3(1 - s_n)(I_n f_{pn})^2} \quad ; \quad R_{rk} = R_{rn} \frac{M_a}{I_a^2 s_n} \quad (14)$$

The total dispersion reactance is composed by stator and rotor linkage, $X_{cc} = X_{fs} + X_{fr}$ and, in general proposed motors usually considered 40% for the rotor and 60% for the stator [4]. These constants are obtained from:

$$K_3 = \frac{X_{cck} - \sqrt{s_m}X_{ccsm}}{1 - \sqrt{s_m}} \quad ; \quad K_4 = \frac{\sqrt{s_m}(X_{ccsm} - X_{cck})}{1 - \sqrt{s_m}} \quad (15)$$

X_{ccsm} , X_{cck} , s_m are dispersion reactance at maximum slip, dispersion reactance at start-up and maximum slip respectively, given by:

$$X_{ccsm} = \frac{R_{rsm}}{s_m} \quad (16)$$

$$X_{cck} = \sqrt{\frac{1}{I_a} - (R_s + R_{rk})_{pu}^2 Z_b} \quad (17)$$

$$s_m = s_n \frac{M_{max} + \sqrt{M_{max}^2 - (1 + 2s_n - 2M_{max}s_n)}}{1 + 2s_s - 2M_{max}s_n} \quad (18)$$

where $R_{sm} = K_1 + K_2\sqrt{s_m}$ is the rotor resistance at maximum slip and $Z_b = \frac{U_{nf}}{I_n}$ is base impedance obtained from the relation between phase voltage and rated current.

3.3 Fault Injection

Unbalanced voltage is produced altering input voltage, is chosen phase-A because it coincide with phase- α of transformed system, that is, the bi-phase system would be transformed to:

$$u_{s\alpha} = \frac{2}{3} \left[(u_{as} + u_{des}) - \frac{1}{2}u_{bs} - \frac{1}{2}u_{cs} \right] \quad ; \quad u_{s\beta} = \frac{\sqrt{3}}{2} (u_{bs} - u_{cs}) \quad (19)$$

where u_{des} is unbalanced voltage.

Stator short circuit suppose a diminution of stator resistance proportional to the turns in short circuit, is chosen phase-A because it coincide with phase- α , therefore, stator resistance is modified to:

$$R_{s\alpha} = (1 - e_{cc})R_s \quad (20)$$

where e_{cc} is the ratio of turns in short circuit.

Broken bar or bars in the rotor cause the resistance in one phase is greater than the other, such phase- α match with the symmetry axis of the rotor causing that rotor resistance in this axis be greater that phase- β :

$$R_{r\alpha} > R_{r\beta} \quad (21)$$

because of this, rotor resistance is modified to:

$$R_{r\alpha} = K_1 + K_{rb}K_2\sqrt{s} \quad ; \quad R_{r\beta} = K_1 + K_2\sqrt{s} \quad (22)$$

where K_{rb} is the increasing resistance due to the broken bar, this value is called broken bar constant.

With this mathematical model and employing Matlab Simulink[®], motors with different power and pole number were simulated under conditions cited above.

4 Methodology to Transform Temporal Response in Patterns

Temporal response consists on different conditions: normal operation; unbalanced input voltage with 5V, 10V, 20V and 30V; stator short-circuit with 5%, 10%, 20% and 30% of inter-turn short circuit at phase-A and; rotor broken bar constants of 1.05, 1.1, 1.2 and 1.3. Figure 2 shows operational curves in normal operation and with 30% of inter-turn short circuit, 1.3 of broken bar constant and 30V of unbalanced voltage. As we can see from the figure a slight variation of the speed curve with fault and without it. The current signal is a combination of the two transformed current, $i_{s\alpha}$ and $i_{s\beta}$, and can be observed a current ripple and a noticeable displacement with respect normal condition.

Some authors [11] [12] perform a spectral analysis especially from signal current, but in this research we proposed an alternative way to resolve this kind of fault in induction motors.

4.1 Pattern Extraction

The first step from pattern extraction is ripple elimination by means digital processing that consists in decimate and signal smooth. Using heuristic analysis that suggests differences between normal and fault operation, we proposed calculate time differences between current and speed signals. Then, the difference

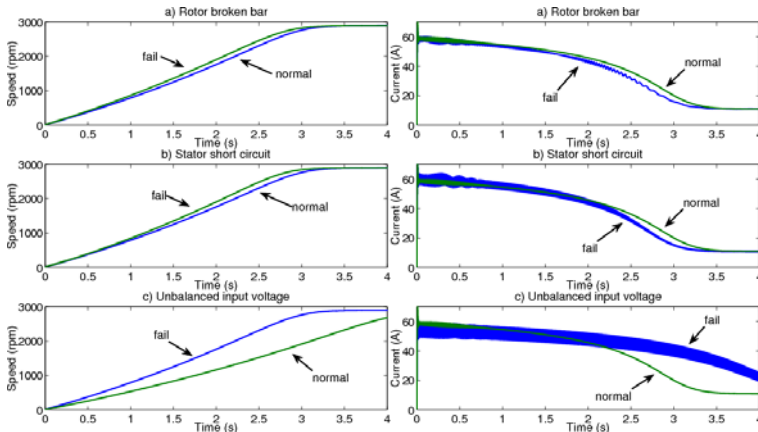


Fig. 2. Operational curves in normal operation and under different faults conditions

between normal condition versus one with fault is presented in Fig. 3 that shows stator short circuit and unbalanced input voltage, where is noticeable a significant difference between faults. In this graphic, the abscissa represents the speed differences and ordinate represents the current differences, that is:

$$X_{axis} = Speed_{without\ fault} - Speed_{with\ fault} \tag{23}$$

$$Y_{axis} = Current_{without\ fault} - Current_{with\ fault} \tag{24}$$

From behaviour in speed and current signals is now presented the particular characteristics extraction.

Relative Unidimensional Characteristics. These characteristics are composed by differences respect time of normal condition versus abnormal operation, hence its name. Sampling time differences between current magnitudes with fault and those without fault are calculated, the new signal conformed by this difference is analysed to extract: 1) maximum peak generated by difference between current without fault and with fault, b) occurrence time of maximum peak relative to establishment time and c) pulse width at 70% of maximum amplitude. It was necessary the application of scalar function to eliminate the transient pulse generated by abrupt raising stator current at start engine moment and limiting the window analysis to twice the establishment time. The Fig. 4 shows a graphic representation of relative unidimensional characteristics.

Relative Bidimensional Characteristics. Since exists a direct relationship between currents and motor speed, the transient time during motor start up provide important information to diagnostic. Therefore are chosen ten current

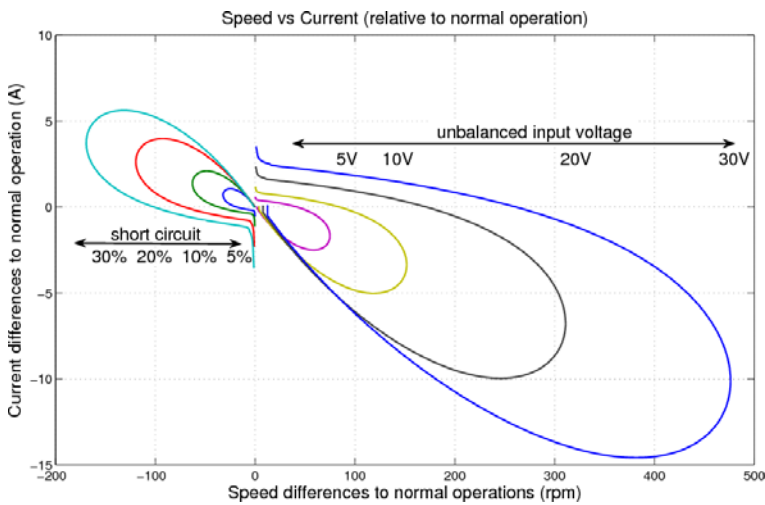


Fig. 3. Speed difference versus current difference for same motor and different abnormal conditions

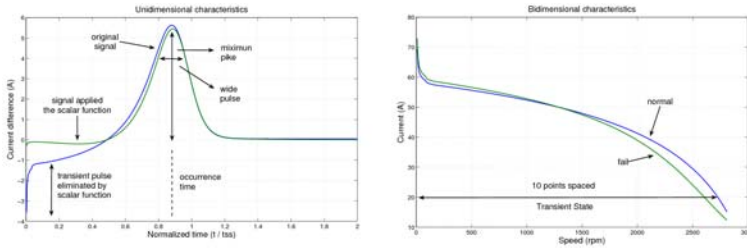


Fig. 4. Graphic representation of unidimensional and bidimensional characteristics

points during transient time taken to equidistant speed values, but these values are normalized respect value without fault, that is:

$$Value_{point} = \frac{Value_{with\ fault}}{Value_{without\ fault}} \tag{25}$$

thus, for normal operation these ten points will have values of one, for abnormal conditions, these points will have values greater than or less than one depending on which type of fault. The Fig. 4 shows the transient start up of speed versus current and illustrates curves with fault and without it.

Absolute Characteristics. These characteristics are taken from motor nominal data, in this case are elected the rated output in kW and motor pole number.

5 Training and Execution of Artificial Neural Network

5.1 Training of Neural Network

The first step was the build of motor database that consists in pattern generation by means of methodology above described. In this paper were selected motors between range of 1kW to 50kW with distinct rated input and pole number. The total number of patterns used from training were 468, extracted from simulated motors through data catalogue.

Each training pattern consist of fifteen characteristic: three from relative unidimensional, ten from relative bidimensional and last two from absolute characteristics, hence, the input neural network layer has fifteen neurons. The output layer has thirteen neurons, recognizing normal operation and each intensity of three faults described, that is, thirteen neuron activating with "1" for each corresponding output neuron. The better performance was reached with three hidden layers composed by 29 neurons on first layer, 19 on second and 17 on third. Test was implemented with different transfer function and the best results were achieved with sigmoid function in hidden layer and linear function in output layer.

5.2 Evaluation of Neural Network

The neural network was tested with 455 patterns and was calculated for each pattern its error depending the fault and intensity. In the evaluation were extracted signals from real motors with history of failure and signals generated by simulation, although the principal source was simulated motors. Table 1 shows the distribution of real and simulated motors, since flexibility of computational algorithm was possible calculate for each simulated motors its thirteen patterns.

Table 1. Collection of patters for neural network evaluation

Motor type	Rotor broken bar	Stator short circuit	Unbalanced input
Real	9	15	15
Simulated	32	32	32

Table 2 shows evaluation results of neural network divided in three categories of fault diagnostic: fault detection, fault isolation and fault identification. The table shows that error detection was 0%, that is, for any fault pattern no matter which type or intensity, the neural network responses activating a neuron different from normal operation indicating a possible fault. Isolation errors indicate activation of any neuron that does not match its fault type presented at input, from table the biggest isolation error occurs in stator inter-turn short circuit with 3.4%. Finally, identification errors indicate activation of any neuron that does not match its intensity presented at input but does match its type, from table the biggest identification error occurs in stator inter-turn short circuit with 11.5%. This error is not critical because the primordial information is correct: indicate and detect which failure occurs.

Table 2. Evaluation of artificial neural network used in diagnostic

Detection errors	Isolation errors			Identification errors		
	Short circuit	Broken bar	Unbalanced	Short circuit	Broken bar	Unbalanced
0%	3.4%	0%	1.8%	11.5%	9.5%	5.9%

6 Conclusions

The present paper demonstrates that computational model is feasible to recognize motor electric fault patterns, such as stator inter-turn short circuit, rotor broken bar and unbalanced input voltage, verifies that motor patterns chosen effectively characterizing induction motor behaviour under distinct operation conditions in a range between 1kW to 50kW rated output, moreover, these patterns allow generalization of neural network used in diagnostic when is presented

unknown patterns at its input. Mathematical model that use varying parameters as function of motor slip and methodology to calculate constants and parameters from data catalogue are verified too, from these information, modeling of motor failures to simulate motor behaviour under abnormal conditions test successfully the methodology presented to identify early potential faults at induction motors.

References

1. Nandi, S., Toliyat, H.A., Li, X.: Condition monitoring and fault diagnosis of electric motors - A review. *IEEE Trans. Energy Convers.* 20, 719–729 (2005)
2. Chow, M.-Y.: Methodologies of using neural network and fuzzy logic technologies for motor incipient fault detection. World Scientific, Singapore (1997)
3. Ozpineci, B., Tolbert, L.: Simulink implementation of induction machine model - A modular approach. In: *IEEE International Electric Machine and Drives Conference (IEMDC 2003)*, vol. 2, pp. 728–734 (2003)
4. Fitzgerald, A.E., Kingsley, C., Umans, S.: *Electric Machinery*. McGraw-Hill, New York (2006)
5. Lipo, T.A.: *Introduction to AC machine design*. University of Wisconsin (2004)
6. Costa-Montiel, A., Galan, N., Ciumbulea, G., López-Fernández, X.: Parámetros del motor de inducción a partir de datos de catálogo. *Energía y Computación* 12, 1–6 (2004)
7. Joksimović, G.M., Penman, J.: The detection of inter-turn short circuits in the stator windings of operating motors. *IEEE Trans. Ind. Electron.* 47, 1078–1084 (2000)
8. Liang, B., Ball, A.D., Iwnicki, S.: Simulation and fault detection of three-phase induction motors. In: *Conference on Computer, Communications, Control and Power Engineering. TENCON 2002*, vol. 3, pp. 1813–1817 (2002)
9. Krause, P.C., Wasynczuk, O., Sudhoff, S.D.: *Analysis of electric machinery and drive systems*. IEEE Press/Wiley-Interscience (2002)
10. López Cárdenas, R., Sánchez Fernández, L.P., Progrebnyak, O., Costa Montiel, Á.A.: Inter-turn short circuit and unbalanced voltage pattern recognition for three-phase induction motors. In: Ruiz-Shulcloper, J., Kropatsch, W.G. (eds.) *CIARP 2008. LNCS*, vol. 5197, pp. 470–478. Springer, Heidelberg (2008)
11. Arthur, N., Penman, P.: Induction machine condition monitoring with higher order spectra. *IEEE Trans. Ind. Electron.* 47, 1031–1041 (2000)
12. Cruz, S.M.A., Cardoso, J.M.: Stator winding fault diagnosis in three-phase synchronous and asynchronous motors, by the extended park's vector approach. *IEEE Trans. Ind. Appl.* 37, 1227–1233 (2001)

Closed-Loop Identification of a Nonlinear Servomechanism: Theory and Experiments

Rubén Garrido and Roger Miranda

Centro de Investigación y Estudios Avanzados (CINVESTAV), México D.F.
{garrido, rmiranda}@ctrl.cinvestav.mx

Abstract. This paper presents a new methodology for parameter identification of a class of nonlinear servomechanisms. The key element is a closed-loop identification technique where a Proportional Derivative controller stabilizes the servomechanism. Experiments using a laboratory prototype, allows comparing the proposed approach against a standard Least Squares algorithm. It is shown that the disturbances acting on the servomechanism do not significantly affect the parameter estimates obtained using the proposed approach.

Keywords: Nonlinear identification, closed-loop identification, servomechanism, persistent excitation.

1 Introduction

Nonlinear servomechanisms typically appear in applications where a brushed or brushless servomotor drives nonlinear loads such as robot arms. However, the problem of estimating the system parameters may get complicated when the plant to be identified must work in closed-loop or if the system is open loop unstable. Identification of systems operating in closed-loop has been a topic of research in the last years because there exist cases where it is not possible to apply directly open-loop identification methods. For example, in unstable industrial processes operating under feedback control, experimental data can only be collected under closed-loop conditions. This last situation remains valid for most mechanical servo systems such as robotic manipulators and high-precision position control systems where the loop may not be removed for security reasons. Several methods for closed-loop identification of linear plants has been proposed in the literature, [1], [2], [3]. However, the presence of nonlinearities in many real-life plants motivates the development of methods suited for this case. References [4] and [5] deal the problem of closed-loop identification of nonlinear time-varying plants operating under linear or nonlinear feedback. In these references the closed-loop system is transformed into one represented using the Youla-Kucera parameters; subsequently, it is identified using open-loop techniques. The controller used for closing the loop defines a subset of linear plants for which a Youla-Kucera parameter may be obtained. However, the method does not identify the parameters of the nonlinear system, but the parameters of a linearized model. In [6], the closed loop output error method, previously

employed for discrete-time linear plants [3] is applied to nonlinear plants in closed loop with a nonlinear controller. A feature of this approach is the fact that the closed-loop system should be strongly strictly passive. This condition is too restrictive and can not be verified a priori in all cases since knowledge of the plant parameters would be necessary. It is worth mentioning that the aforementioned references do not present numerical simulations or experimental results. Further, these results would not be suitable for identifying certain class of nonlinear plants, for instance, nonlinear servomechanisms or robot manipulators. A depart from these previous methods is proposed in [7] where an inner-outer relay structure is proposed for identifying Coulomb friction in a positioning system, where the inner relay models the Coulomb friction. The outer relay is introduced to close the loop and generates periodic motion. Stability of the periodic solution of the closed-loop system is studied by means of the Poincaré map. Parameter estimation is performed through a recursive Least Squares algorithm. The method is tested experimentally showing good results; unfortunately, it does not provide parameter estimates of the linear part of the positioning system. The method proposed in [8] identifies a second order nonlinear system using a gradient algorithm. A high-gain controller stabilizes the open-loop unstable plant. That approach imposes differentiability conditions on the regressor vector and the paper only shows simulation.

This work outlines an approach for closed-loop identification of nonlinear servomechanism. This method extends previous results [9] to deal with a class of nonlinear servomechanisms. Moreover, compared with previous works [8], no differentiability conditions are imposed on the regressor vector. Experiments permit comparing the proposed approach against a standard Least Squares (LS) algorithm. These experimental results show that disturbances greatly affect the estimates obtained using the LS algorithm. In the case of the proposed approach, disturbances do not significantly influence the parameter estimates. The paper is organized as follows. Section 2 is devoted to the identification algorithm, stability properties and parameter convergence. Experimental results are shown in Section 3 and some concluding remarks are given in Section 4.

2 Closed-Loop Parameter Identification

2.1 Preliminares

Fig. 1 depicts the idea of the closed loop identification algorithm. Two Proportional Derivative (PD) controllers simultaneously close the loop around the nonlinear servomechanism and its estimated model and an update law computes its parameter estimates. It is worth mentioning that both controllers have the same gains; moreover, under some mild conditions, the PD controller is able to stabilize the nonlinear servomechanism without knowledge of its parameters. Consider the nonlinear servomechanism model described by

$$\ddot{x} = -\gamma^T f\left(x, \dot{x}\right) + bu \quad (1)$$

where \dot{x} denotes the time derivative of x , $\mathbf{x} = [x, \dot{x}]^T$ are the states of the nonlinear servo, which are available through sensors, $\gamma \in \mathbb{R}^m$ and $b \in \mathbb{R}^+$ are unknown constant parameters, u is the input control, and $f(x) = f(x, \dot{x}) \in \mathbb{R}^m$ is a nonlinear known function.

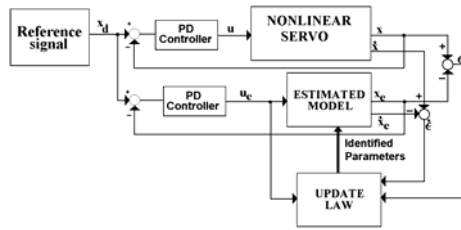


Fig. 1. Schematic for the proposed parameter identification algorithm

Model (1) fulfills the following assumption: **(A1)** The nonlinear function f fulfills $\|f(x)\| \leq k_f, \|f(x_1) - f(x_2)\| \leq L_1 \|x_1 - x_2\|$.

2.2 Stability Analysis of the Closed-Loop System

Consider the PD control law $u(t) = k_p e - k_d dx/dt$, where, $k_p > 0, k_d > 0$ and the error e is defined as $e := x_d - x$. The reference x_d fulfills the following assumption: **(A2)** The reference x_d is bounded, i.e. $|x_d| \leq \beta$. Then, substituting u into (1) yields

$$\ddot{x} = -\gamma^T f(x, \dot{x}) + b k_p e - b k_d \dot{x} \tag{2}$$

Firstly, it is necessary to find conditions under which controller $u(t)$ stabilizes system (1). To this end, consider the following Lyapunov function candidate $V_1(x) = 0.5x^T \begin{bmatrix} b(k_p + \mu k_d) & \mu \\ \mu & 1 \end{bmatrix} x$, where $\mu > 0$. Function V_1 is positive definite if

$\mu < b k_d$. Now, by taking the time derivative of V_1 along the trajectories of (2) and using $\gamma_l = \|\gamma\|$, bounds on $f(x)$, equation of $e(t)$ and bound on x_d leads to $\dot{V}_1 \leq -(b k_d - \mu) \dot{x}^2 - \mu b k_p x^2 + \mu(\gamma_1 k_f + b k_p \beta) |x| + (\gamma_1 k_f + b k_p \beta) \left| \dot{x} \right|$. Let

the PD controller gains to be written as $k_p = \alpha L_p$ and $k_d = \alpha L_d$. Then, assuming that the inequality $\mu < 0.5 b k_d$ holds, and considering $z^T = [x, dx/dt]$ and the following definitions: $M = \text{diag} \{ \mu b L_p, 0.5 b L_d \}, B^T = [\mu(\gamma_1 k_f + b k_p \beta) \quad \gamma_1 k_f + b k_p \beta]$ allows writing the bound dV_1/dt as

$$\dot{V}_1 \leq -\alpha z^T M z + B^T z \leq -\alpha \lambda_{\min}(M) \|z\| \left[\|z\| - \|B\| (\alpha \lambda_{\min}(M))^{-1} \right] \tag{3}$$

Consider the set Ω defined as $\Omega := \{z: \|z\| < \alpha \|B\| (\lambda_{\min}(M))^{-1}\}$. Then, dV_1/dt is negative definite if $z \notin \Omega$. Therefore, the feedback system (2) is Uniformly Ultimate Bounded

(UUB) [10] and the region Ω can be made arbitrarily small if α increases, i.e., with a high gain controller.

2.3 Stability Analysis of the Error Dynamics

Consider the following model of the nonlinear servomechanism (1):

$$\ddot{x}_e = -\hat{\gamma}^T f(x_e, \dot{x}_e) + \hat{b}u_e, \text{ where } \mathbf{x}_e = [x_e, \dot{x}_e]^T \text{ is the estimated state and parameters}$$

$\hat{\gamma}$, \hat{b} are the estimates of γ , b , respectively. Let the PD control law:

$$u_e = k_p e_e - k_d \dot{x}_e, \text{ with } e_e := x_d - x_e. \text{ It is worth noting that the proportional and derivative gains for } u \text{ and } u_e \text{ are the same. Substituting } u_e \text{ into the model of the nonlinear servomechanism leads to}$$

$$\ddot{x}_e = -\hat{\gamma}^T f(x_e, \dot{x}_e) + \hat{b}k_p e_e - \hat{b}k_d \dot{x}_e. \text{ Define the error between the output of the nonlinear system (2) and its model as } \varepsilon := x - x_e. \text{ Therefore, the associated error dynamics corresponds to the following differential equation}$$

$$\ddot{\varepsilon} = -bk_d \dot{\varepsilon} - bk_p \varepsilon - \gamma^T (f(x) - f(x_e)) + \tilde{\theta}^T \phi \tag{4}$$

where $\tilde{\theta}^T := [\hat{\gamma} - \gamma, \hat{b} - b]$, $\phi := [f(x_e), -u_e]$. The stability analysis employs passivity arguments [12]. Define $E := \mathbf{x} - \mathbf{x}_e = [\varepsilon, d\varepsilon/dt]^T$ and consider the storage function

$$H_1(\varepsilon, \dot{\varepsilon}) = 0.5 E^T \begin{bmatrix} bk_p & \mu \\ \mu & 1 \end{bmatrix} E. \text{ This function is positive definite if } \mu < \sqrt{bk_p}.$$

Taking the time derivative of H_1 along the trajectories of (4) leads to

$$\dot{H}_1 \leq \tilde{\theta}^T \phi (\mu \varepsilon + \dot{\varepsilon}) - \frac{1}{2} bk_d (\mu \varepsilon + \dot{\varepsilon})^2 - \frac{1}{2} E^T N E - \gamma^T (f(x) - f(x_e)) (\mu \varepsilon + \dot{\varepsilon}), \text{ where}$$

$N = \text{diag}\{2\mu bk_p - \mu^2 bk_d, bk_d - 2\mu\}$. Matrix N is positive definite if $\mu < \min\{2k_p/k_d, bk_d/2\}$.

Define $\Xi := [|\varepsilon|, |d\varepsilon/dt|]^T$ and consider the following upper bounds

$$\gamma^T (f(x) - f(x_e)) \leq \gamma_1 L_1 \|E\|, \quad \mu \varepsilon + \dot{\varepsilon} \leq \sqrt{\mu^2 + 1} \|\Xi\|. \text{ Since } \|E\| = \|\Xi\|, \text{ it follows that}$$

$$\gamma^T (f(x) - f(x_e)) (\mu \varepsilon + \dot{\varepsilon}) \leq \gamma_1 L_1 \sqrt{\mu^2 + 1} \|E\|^2. \text{ Therefore, } dH_1/dt \text{ is upperbounded as}$$

$$\dot{H}_1 \leq \tilde{\theta}^T \phi (\mu \varepsilon + \dot{\varepsilon}) - \frac{1}{2} bk_d (\mu \varepsilon + \dot{\varepsilon})^2 - \left[\frac{1}{2} \lambda_{\min}(N) - \gamma_1 L_1 \sqrt{\mu^2 + 1} \right] \|E\|^2 \text{ and if}$$

$\lambda_{\min}(N) > 2\gamma_1 L_1 \sqrt{\mu^2 + 1} > 0$ holds, then $\dot{H}_1 \leq \tilde{\theta}^T \phi (\mu \varepsilon + \dot{\varepsilon}) - \frac{1}{2} bk_d (\mu \varepsilon + \dot{\varepsilon})^2$. The

above means that (4) defines an output strictly passive (OSP) mapping described as

$$\tilde{\theta}^T \phi \rightarrow (\mu \varepsilon + \dot{\varepsilon}). \text{ As a consequence, the output defined as } (\mu \varepsilon + \dot{\varepsilon}) \text{ belongs to the space } L_2. \text{ Then, } \varepsilon \text{ is the output of a linear exponentially stable system whose input}$$

belongs to the space L_2 , thus [13]: $\varepsilon \rightarrow 0$ as $t \rightarrow \infty$. The last analysis makes intuitive to consider the next parameter updating law

$$\dot{\tilde{\theta}} = -\Gamma \phi(\mu\varepsilon + \dot{\varepsilon}) \tag{5}$$

where $\Gamma > 0$, $\Gamma = \Gamma^T \in \mathbb{R}^{2 \times 2}$. Let the storage function $H_2(\tilde{\theta}) = 0.5\tilde{\theta}^T \Gamma^{-1}\tilde{\theta}$, whose time derivative along the trajectories of (5) leads to $\dot{H}_2(\tilde{\theta}) = (\mu\varepsilon + \dot{\varepsilon})(-\tilde{\theta}^T \phi)$.

Consequently, (5) defines a passive operator $(\mu\varepsilon + \dot{\varepsilon}) \rightarrow (-\tilde{\theta}^T \phi)$. Therefore, the interconnected system (4)-(5) defines an OSP operator with output $(\mu\varepsilon + \dot{\varepsilon})$ [12]. Let $H = H_1 + H_2$; it is not difficult to conclude that $dH/dt \leq -(1/2)bk_d(\mu\varepsilon + \dot{\varepsilon})^2$ and as result $\{\varepsilon, d\varepsilon/dt, \tilde{\theta}\} \in L_\infty$. Moreover, it was previously shown that (2) is UUB; therefore, $\{x, dx/dt\} \in L_\infty$ and so do $\{x_e, dx_e/dt\}$. Based on the last analysis, it is possible to state the following result.

Proposition 1. Consider the feedback interconnected system (4)-(5), assume that parameter μ satisfies: $\mu < \min\{2k_p/k_d, bk_d/2, \text{sqtr}(bk_p)\}$ and that $\lambda_{\min}(N) > 2\gamma_1 L_1 \sqrt{\mu^2 + 1} > 0$ holds; then, $\{\varepsilon, d\varepsilon/dt, \tilde{\theta}, x_e, dx_e/dt\} \in L_\infty$ and $\varepsilon \rightarrow 0$ as $t \rightarrow \infty$.

2.4 Parameter Convergence

The following persistence of excitation (PE) condition ensures parameter convergence [11].

Definition 1. A vector $\phi : R^+ \rightarrow R^{2n}$ is persistently exciting if there exist $\alpha_1 > 0, \alpha_2 > 0$ and $\delta > 0$ such that $\alpha_1 I \leq \int_{t_0}^{t_0+\delta} \phi(\tau)\phi^T(\tau) d\tau \leq \alpha_2 I$.

3 Experimental Results

The following nonlinear rotating servomechanism model allows testing the proposed methodology

$$J \ddot{x} + f \dot{x} + gml \sin(x) = \tau \tag{6}$$

Angular position, velocity and acceleration correspond to $x, dx/dt, d^2x/dt^2$, respectively; J and f are the servomechanism inertia and viscous damping respectively; $gml \sin(x)$ is a gravitational torque due to a pendulum of length l and mass m ; g is the gravity constant, and τ the torque provided by a DC motor. If the amplifier driving the motor works in current mode, it is reasonable to assume that

$\tau=ku$, i.e., the torque provided by the motor is proportional to the voltage control signal. Fig. 2 depicts the convention for measuring the angular position x . Rewriting model (6) yields $\ddot{x} = -\gamma^T f(x) + bu$, where $f(x) = [\sin(x), dx/dt]^T$, $\gamma = [gml/J, f/J]$, $b=k/J$. The servomechanism employed for the experiments (Fig. 2) is a DC brushed motor controlled by a Copley Controls power amplifier, model 413, configured in current mode. The angular position is measured using a BEI optical encoder with a resolution of 2500 pulses per revolution and it is directly coupled to the motor shaft. Data acquisition is performed by a MultiQ 3 card from Quanser Consulting endowed with inputs for optical encoders. The electronics associated to these inputs multiply by 4 the encoder resolution. Angular position is scaled down by a factor of 10,000 corresponding to one motor shaft turn. The card has 12 bits digital-analog converters with an output voltage of $\pm 5V$. Programming was performed using the Matlab-Simulink software operating with the WINCON software from Quanser Consulting. Sampling period was 1 ms. The experiments are divided into three parts. In the first part, parameters of the nonlinear servomechanism are identified using a standard Least Squares (LS) algorithm and the proposed methodology without adding external disturbances. In the second part, the parameter estimates obtained in the first part allows designing a tracking controller that is applied to the nonlinear servomechanism. Finally, the experiments previously performed in the first part are repeated under a constant disturbance applied to the nonlinear servomechanism.

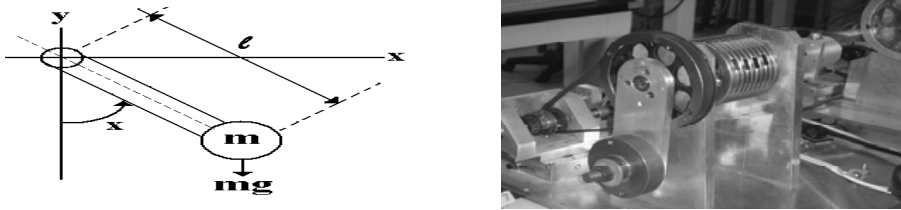


Fig. 2. Convention for measuring x (left hand side); laboratory prototype (right hand side)

3.1 Parameter Identification without Adding a Disturbance

The PD controller gains are set to $k_p=20$ and $k_d=1.3$. Velocity estimates dx_f/dt of dx/dt were obtained through filtering of the position measurements with $dx_f/dt=d\omega/dt$ and $d^2\omega/dt^2+1100 d\omega/dt+30000 \omega = 300000x$. The above filter is also employed in the model even if velocity dx/dt is available. The following relationships allow implementing the LS algorithm (see also Fig. 3)

$$\begin{aligned} \ddot{y}_1 + \lambda_1 \dot{y}_1 + \lambda_2 y_1 &= \lambda_2 \sin(x); \phi_1 = y_1; \ddot{y}_2 + \lambda_1 \dot{y}_2 + \lambda_2 y_2 = \lambda_2 \dot{x}; \phi_2 = \dot{y}_2 \\ \ddot{y}_3 + \lambda_1 \dot{y}_3 + \lambda_2 y_3 &= \lambda_2 u; \phi_3 = y_3; \ddot{y}_4 + \lambda_1 \dot{y}_4 + \lambda_2 y_4 = \lambda_2 x; z = \ddot{y}_4 \end{aligned} \tag{7}$$

with $\lambda_1=40$, $\lambda_2=400$, $z = \theta^T \phi_{LS}$ and $\phi_{LS} = [\phi_1 \ \phi_2 \ \phi_3]$. Note that equation for z holds for all time t ; then it is valid for the instants $h, 2h, 3h, \dots, (k-1)h, kh$, where h is the sampling period. Then, z can be rewritten as $z(kh)=\theta^T \phi(kh)$; omitting the sampling

period yields $z(k)=\theta^T \phi(k)$. The above expression allows using the Least Squares algorithm with forgetting factor [14]. The LS algorithm is implemented using $P(0)=diag\{500,500,500\}$ and a forgetting factor constant value of $\lambda=0.999$. The gains for the proposed algorithm were set to $\Gamma=diag\{500,15000,10000\}$ and $\mu=0.5$. The nonlinear servomechanism is excited using a signal provided by the Band-Limited White Noise block from MatLab Simulink (see Fig. 3) set to: noise power of 5, sample time of 0.1 and seed [12121]. Fig. 4 (a) and (b) show the time evolution of the parameter estimates obtained using the LS algorithm and the proposed approach respectively.

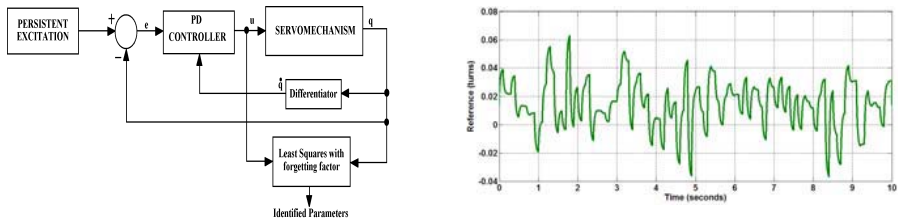


Fig. 3. Block diagram for implementing the Least Squares algorithm (left hand side); reference signal for the identification experiments (right hand side)

For the LS algorithm the parameters estimates approximately settle at $\{\hat{\gamma}^T, \hat{b}\}=\{28,8,2\}$ whereas for the proposed algorithm the corresponding values are $\{\hat{\gamma}^T, \hat{b}\}=\{26,10,3\}$. From the above it is clear that both algorithms give similar parameter estimates.

3.2 Trajectory Tracking Experiments

Let $\{\bar{\gamma}^T, \bar{b}\}=\{\bar{\gamma}_1, \bar{\gamma}_2, \bar{b}\}$ be the estimated parameters obtained with any of the parameter identification algorithms during the first ten seconds. Then, consider the control law
$$\bar{u} = (\bar{b})^{-1} \left[\bar{\gamma}^T f(x, \dot{x}) + \ddot{x}_d + k_1 \dot{e} + k_2 e \right]$$
, with k_1, k_2 being positive constants and e is the tracking error. The above control law applied to (1) leads to the following closed-loop error dynamics
$$\ddot{e}(t) + k_1 \dot{e} + k_2 e = (\gamma - \bar{\gamma})^T f(x, \dot{x}) + (\bar{b} - b)u.$$

Note that if the parameter estimate errors $\gamma - \bar{\gamma}$ and $\bar{b} - b$ are zero, then, the tracking error converges to zero. In practice, good parameter estimate would lead to smaller tracking error e . The reference is a sinusoid with amplitude of 0.25 motor shaft turns and frequency of 4 rad/s. Fig. 5 (a) and (b) show the tracking error when the control law \bar{u} is computed using the LS and the proposed algorithm, respectively. The trajectory errors are similar.

3.3 Parameter Identification under Constant Disturbances

The parameter estimation algorithms are tested under the same conditions used previously, however, a constant disturbance $d=0.5$ v is considered in the model of the nonlinear servomechanism: $\ddot{x} = -\gamma^T f(x) + bu + d$. Fig. 6 (a) and (b) depict the results for the LS and the proposed identification algorithm, respectively.

Note that the parameter estimates obtained using the LS algorithm are far different from the estimates obtained under no disturbance conditions. In particular, note that parameter $\hat{\gamma}_2$ takes negative values. On the other hand, the parameter estimates obtained with the proposed method stay near to the values obtained under no disturbance conditions.

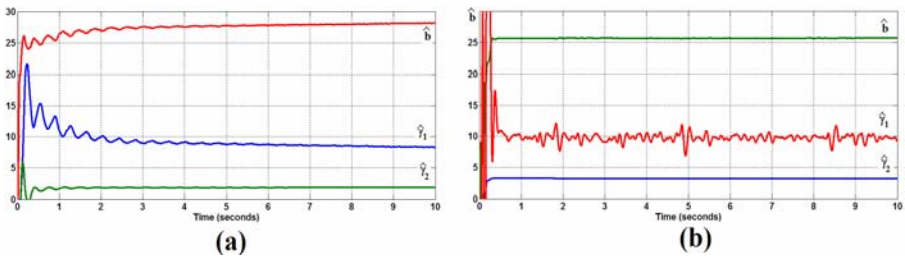


Fig. 4. Parameter estimates (non perturbed case): (a) LS algorithm; (b) proposed method

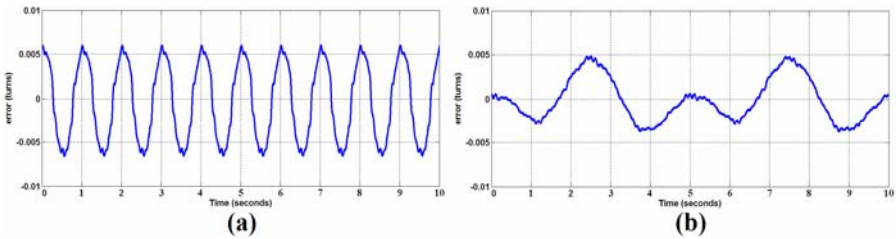


Fig. 5. Tracking error: (a) LS algorithm; (b) proposed algorithm

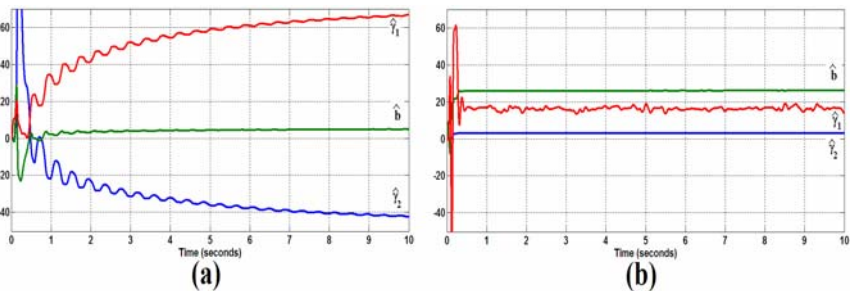


Fig. 6. Parameter estimates (perturbed case): (a) LS algorithm; (b) proposed method

4 Conclusion

This paper presents a new methodology for parameter identification of a class of nonlinear servomechanisms. The key element is a closed-loop identification technique where a Proportional Derivative controller stabilizes the servomechanism without knowledge of its parameters. The proposed identification algorithm and a standard Least Squares method are experimentally compared using a laboratory prototype. The experiments show that both algorithms give similar results when disturbances are not considered and produce low tracking errors when the parameter estimates are used for designing a trajectory-tracking algorithm; however, the proposed algorithm gives reasonable results in face of constant disturbances whereas the Least Squares method produces estimates far from the values obtained in the non-disturbance case.

References

1. Van Donkelaar, E.T., Van den Hof, P.M.J.: Analysis of closed-loop identification with a Taylor made parametrization. *European Journal of Control* 6(1) (2000)
2. Van den Hof, P.M.J.: Closed loop issues in system identification. *Annual Reviews in Control* 22 (1998)
3. Landau, I.D., Karimi, A.: An output error recursive algorithm for unbiased identification in closed loop. *Automatica* 33(5) (1997)
4. Dasgupta, S., Anderson, B.D.O.: Parametrization for the Closed-Loop identification of nonlinear time-varying systems. *Automatica* 32(10) (1996)
5. Linard, N., Anderson, B.D.O., De Bruyne, F.: Identification of a nonlinear plant under nonlinear feedback using left coprime fraction based representations. *Automatica* 35(4) (1999)
6. Landau, I.D., Anderson, B.D.O., De Bruyne, F.: Recursive identification algorithms for continuous-time nonlinear plants operating in closed loop. *Automatica* 37 (2001)
7. Besancon-Voda, A., Besancon, G.: Analysis of a two-relay system configuration with application to Coulomb friction identification. *Automatica* 35(8), 1391–1399 (1999)
8. Huang, J.T.: Parametric Identification for Second-Order Nonlinear Systems in Closed-Loop Operations. *Journal of Dynamic Systems, Measurement, and Control* 128(3) (2006)
9. Garrido, R., Miranda, R.: Autotuning of a DC Servomechanism using Closed Loop Identification. In: *IEEE 32nd Annual Conference on Industrial Electronics, IECON 2006* (October 2006)
10. Khalil, K.H.: *Nonlinear systems*. Prentice-Hall, Englewood Cliffs (1996)
11. Sastry, S., Bodson, M.: *Adaptive Control, Stability, Convergence and Robustness*. Prentice-Hall, Englewood Cliffs (1989)
12. Ortega, R., Loría, A., Nicklasson, P.J., Sira-Ramirez, H.: *Passivity-based control of Euler-Lagrange Systems: Mechanical, Electrical and Electromechanical applications*. Springer, Heidelberg (1998)
13. van der Schaft, A.: *L2 Gain and Passivity Techniques in Nonlinear Control*. Springer, Heidelberg (1999)

Dynamic Classifier Selection with Confidence Intervals

R.M. Valdovinos, M. Sánchez, and Issachar Ruiz

Computo Aplicado Group, Centro Universitario UAEM Valle de Chalco
Universidad Autonoma del Estado de México

Hermenegildo Galeana 3, Col. Ma. Isabel, 56615, Valle de Chalco, México
li_rmvr@hotmail.com, zack79_09@hotmail.com, issa_984@hotmail.com

Abstract. Nowadays, the ensembles are a popular classification method. In order to obtain the final decision the selection and the fusion methods are used. In this paper, the Dynamic Classifier Selection with Confidence Intervals (DCS-CONFI) method is proposed. This method use confidence intervals for identify the true knowledge or the influence of each individual classifier in the final decision, thus, the member with higher confidence interval is chosen for classify the test pattern. The experimental results demonstrated the convenience of to determinate the confidence level when the classifier selection scheme is used.

1 Introduction

Ensemble is a learning paradigm where several classifiers are combined in order to generate a single classification result. It has been found that combining multiple decision can produce more robust, reliable and efficient recognition performance than using a single expert classifiers. Let $\mathcal{D} = \{D_1, \dots, D_L\}$ be a set of L classifiers, and $\Omega = \{\omega_1, \dots, \omega_c\}$ be a set of c classes. Each classifier D_i ($i = 1, \dots, L$) gets as input a feature vector $\mathbf{x} \in \mathbb{R}^d$, and assigns it to one of the c problem classes.

Two strategies are accepted for combining classifier decisions: fusion and selection. In classifier fusion, each member of the ensemble is supposed to have knowledge of the whole feature space and thus, they are either complementary or competitive. On the other hand, in the selection approach only one of the members is who decides the class label for the test sample. Studies pointed out that, the selection between different classifier could be better if the classifiers are "specialized" on the recognition of different data set partitions. Each "specialized" classifier will be thus required to has a high accuracy for the test patterns that belong to one "region" of the feature space [7].

Focusing on the selection strategy, it can be with two methods: static selection and dynamic selection. In the static selection, the classifier expert is assigned during the learning phase and, in the dynamic selection the expert is determined in the classification phase.

In order to identify the confidence of the individual classifier or to analyze the cost-sensitive classification, several strategies are proposed [12], [15], [14], [16]. The goal of determining confidence is to produce a method which takes information about the behavior of each classifier on a specific subject and to obtain the confidence level using the

certain of decision. On other hand, the cost-sensitive classification methods is to build a classifier to minimize the expected misclassification costs rather than minimizing the expected number of misclassification errors.

In the current work, we include two new methods for determining the confidence intervals on the Dynamic Classifier Selection (DCS) [10]. For that, we use two different models of ensembles: based on Genetic Algorithms and based on resampling methods. The effectiveness of these approach is empirically tested over real-problem data sets.

From now on, the rest of the paper is organized as following. Section 2 describe the confidence measures, Section 3 provides related works about classifier selection methods. Section 4 expose the Dynamic Classifier Selection with confidence intervals (DCS-CONFI), method here proposed. In the Section 5 and Section 6 details of the resampling and Genetic algorithms methods are showed. Next, the experimental results are discussed in Section 7. Finally, Section 8 gives the main conclusions and points out possible directions for future research.

2 Confidence Measures

The ensembles can be categorized into three groups to according to the level of classifiers outputs [19]

- Abstract level (class label).
- Rank level (rank order).
- Measurement level (class scores).

The ensembles at measurement level has the advantage to provide richer information of class measures. This information represent the degree of confidence of decision, like the class posterior probability or likelihood. A confidence measure can be defined as a value that describe the trust a classifier has in its own classification results. The higher the confidence in a class label, the higher probability that this class label is indeed the correct label for an unknown test pattern.

The confidence value for a single classifier can be defined as follow. Let D_i that is a single classifier of set D classifiers and a discriminant function $D_i(x) : x \rightarrow \Omega$. For each sample x the classification based on D_i is either correct or incorrect, therefore a correctness function $C(x, \omega)$ can be defined:

$$C(x, \omega) = \begin{cases} 1, & \text{if } D_i(x) = \omega \\ 0, & \text{if } D_i(x) \neq \omega \end{cases} \quad (1)$$

where ω is the true class label of sample x . Thus for a test pattern classified, the confidence is given by

$$q(x) = P(C(x, \omega) = 1|x). \quad (2)$$

that is defined as the probability that the classification is correct. The classification confidence is thereby the posteriori probability for the estimated class.

Many works assumed that the classifiers outputs of neural networks and parametric statistical classifiers already represent the estimate of class probabilities. For many other non-parametric classifiers, however, the outputs have diverse meanings and ranges, and so, they need to be transformed to uniform confidence measurements.

The confidence transformation method consist in the combination of a scaling function and an activation function. The scaling function shifts and re-scales the classifier outputs to a moderate range around to zero such that the outputs of different classifiers are comparable. A simple method to re-scale the output values is the function called *Global Normalization*,

$$f_i(d) = \frac{d_i - \mu_o}{\sigma_o} \quad (3)$$

where μ_o and σ_o are the mean and variance of the pooled classifier outputs respectively, and d denote the set of outputs of classifiers.

The re-scaled outputs is transformed to confidence measure using an activation function corresponding to one of four confidence types: log-likelihood (linear), likelihood (exponential), sigmoid, and evidence.

3 Classifier Selection

The classifier selection methods identify the suitable classifier to assign the class label corresponding to a test pattern (TP). Thus, for each TP only one classifier makes the final decision. For that, the classifier selection methods divide the training set in r interesting regions, where r must be $r > 1$. Each region is denoted by $\mathcal{R} = \{R_1, \dots, R_r\}$ and, the number of classifiers L is not necessarily equal to the number of regions r [6]. In addition, each classifier can be expert in one or several R regions.

3.1 Static Classifier Selection (SCS)

In this method, the regions are established during the training phase, previous to the classification of patterns. In the operation (classification) phase, the region of the pattern y is firstly found (R_j), then, using the D_j classifier (responsible of the region j) the corresponding label is assigned to y . The allocation Region - classifier can become of two forms:

- Begin specifying the region and later assigning a responsible classifier [9], [8].
- Find the region (or regions group) for each classifier, where the classifier has better performance [12], [11].

3.2 Dynamic Classifier Selection (DCS)

Unlike the static selection, in the dynamic selection, the competence regions are obtained during the operation phase. The division commonly is based on the accuracy of each decision, having greater preference the classifier more accurate. This algorithms estimate the local accuracy and are called *Dynamic Classifier Selection with Local Accuracy (DCS-LA)*.

Woods et al [10] propose two methods which consider local accuracies. The first one, called *overall local accuracy* (DCS-OLA), considers the right percentage of TP classified by each classifier L . The second method, *local class accuracy* (DCS-LCA), considers the class assigned by a classifier to the TP, and then, it calculates the percentage of the training patterns correctly classified towards the same class by using the

k -Nearest Neighbor Rule (k -NN). The main difference between both methods is related to the information used to compute the "local accuracy", thus, the former has been called *a priori* selection, and the latter as *a posteriori* selection *GiaTesis*.

The general algorithm of DCS-LA is the next [10]:

1. Design the individual classifier D_1, \dots, D_L . Pick the value of the parameter k .
2. Upon obtaining an input x , label it by D_1, \dots, D_L . If all classifiers agree on the label, then assign this label to x and return.
3. If there is a disagreement, then estimate the local accuracy for each $D_i, i = 1, \dots, L$. To do this, take the class label offered for x by D_i , say $s \in \Omega$, and find the k points closest to x for which D_i has issued the same label. Calculate the proportion of the points whose true labels were s to be an estimate of the local accuracy of D_i with respect to class s .
4. If there is a unique winner of the local accuracy contest, let it label x and return. Otherwise, check if the tied winners have the same labels for x . In this case, accept the label and return. If a unique class label might be selected by plurality among the tied classifier, then assign this label to x and return.
5. Otherwise, there is a class label tied between the most locally competent classifier. The classifier with the next highest local competence is identified to break the tie. If all classifier are tied and the class label is still tied, then pick a random class label among the ties *labels* and return. If there is a unique winner of the (second) local competence contest and it can resolve the tie, then use the winning label for x and return.
6. If none of the clauses in the previous point apply. Then break the class label tie randomly and return a label for x .

Modifications for this method were proposed by Kuncheva [8], that includes the random selection form of the winning classifier when ties appear. Other modification was proposed by [18], that use a k -nearest neighbor weighted by obtain the local accuracies and by Giacinto [6], where the local estimation is made using the class probabilities, unlike Woods, Giacinto only considers the label of class assigned to the TP.

4 Dynamic Classifier Selection with Confidences (DCS-CONF1)

The DCS-CONF1 is based on DCS-LA algorithm, the main difference is the use of two confidence measure in the k -NN rule. For apply the intervals we consider two approaches: using the distances and using the space volume filled by the k nearest neighbor.

The first confidence, take into account the distances of the nearest neighbor as well as their class. The confidence level for the classifier D_i^{cf} is given for the next formula:

$$D_i^{cf} = \operatorname{argmax} \frac{\sum_j^c d_j}{d_k} \quad (4)$$

where d is a Euclidean distance, d_j are the distances of the training samples of the class j including in the k nearest neighbor and, d_k is the sum of the all distances of the k nearest neighbor.

The second confidence measure here used, originally was proposed in [3] as a rejected method of noisy patterns in Handwritten Character Recognition. This function is estimated by looking at the space volume filled by the k nearest neighbor of that class. For estimate the confidence of each class the formula is the next:

$$D_i^{cf} = \operatorname{argmax} \frac{k}{n_i v_i} \quad (5)$$

where n_i is the number of training samples in the class i and v_i is a volume measure filled by the k nearest one.

In the ensembles context, using these confidence value, each classifier take him decision according the high value obtained. When the individual decisions are, all of them are compared and chose that who have higgler confidence value. This procedure is shown following:

1. Design the individual classifier D_1, \dots, D_L . Pick the value of the parameter k .
2. Upon obtaining an input x , label it by $\mathcal{D}_1, \dots, \mathcal{D}_L$. If all classifiers agree on the label, then assign this label to x and return.
3. If there is a disagreement, then estimate the local confidence level for each $\mathcal{D}_i, i = 1, \dots, L$ respect to class s . To do this, use any equation exposed above.
4. The classifier which provide the higgler confidence level let the label x and return.

5 Resampling Methods

Selection with replacement of patterns is the main characteristic of the resampling methods used for classifier ensembles. In this section, we briefly describe two of the methods more widely used. These are here implemented by means of the class-dependent strategy [5], which consists of picking instances in a way that the initial class distributions are preserved in each of the h subsamples generated.

The rationale behind this strategy is to reduce the possibly high computational complexity (computing time and storage loads) of the base classifiers induced by each of the subsamples. In general, the smaller the subsample size, the lower the classifier complexity.

5.1 Bagging

Bagging (Bootstrap Aggregating) [2] is the simplest and earliest resampling method. This algorithm employs bootstrap sampling to generate several subsamples by random sampling with replacement, m examples from the original training set (also of size m). The individual predictions are often by majority voting combined. Note that many of the original instances may be repeated in the resulting subsample while others may be left out.

Briefly, the bagging method generates L bootstrap subsamples $\{S_1, S_2, \dots, S_L\}$ of size m from the original training set T and creates the corresponding L base classifiers. The output produced by the ensemble is the class label with a majority of votes.

5.2 Boosting and Arc-x4

Boosting and its main variant AdaBoost (Adaptive Boosting) [4] sequentially generates a series of individual classifiers, where the training instances wrongly predicted by previous base classifiers are picked more often than examples correctly classified. In general, every variant of boosting attempts to produce new classifiers capable to better predict examples for which the current ensemble fails. This is done for minimizing the expected error. Arc-x4 [1] is a variant of Boosting but, Arc-x4 use a easier method fo adjust de weights.

AdaBoost and Arc-x4 generates L subsamples S_1, S_2, \dots, S_L using a weight for each one of the m instances and thus L individual classifiers D_1, D_2, \dots, D_L are built. At each stage $l(l = 1, \dots, L)$, the weight $W_l(i)$ defines the probability of adding the instance x_i into subsample S_l and represents the "difficulty" in predicting such instance by the previously created base classifier D_{l-1} . Initially, the probability of picking each instance is set to $W_1(i) = 1$, and then the weights are modified at each step $1 < l = L$.

There are two ways in determining the subsamples employed in AdaBoost. The first one is picking a set of examples based on the probabilities of the instances (this probability depends on how often that example was by the previous classifiers misclassified). With this strategy, "difficult" instances are likely to appear more than once in the next subsample.

In the second one, the implementation simply consists of using all the instances and the weights corresponding to its probability. At each step l , the corresponding base classifier D_l is built and then we compute its error E_l using the weights W_l as

$$E_l = \sum_{i=1}^N W_l(i)(l - y_{i,l}) \tag{6}$$

Were $y_{i,l} = 1$ if D_l produces the correct label, $y_{i,l} = 0$ otherwise.

The criterion to stop in this algorithm is $E_l \geq 0.5$. Otherwise it computes a coefficient $\beta_l = E_l / (1 - E_l)$ to be in the weighted voting of the ensemble used and also to update the Boosting weights of the individual instances as follows:

$$W_{l+1}(i) = W_l(i)\beta_l^{l-y_{i,l}} \tag{7}$$

A final classification produced for a new sample x is given by weighted voting among the L base classifiers, where the weight of each base classifier depends on its performance on the subsample used for building classifiers. The final decision of the ensemble D for the sample x corresponds to the class label c with the maximal support according to:

$$D(x) = \underset{c \in \Omega}{\operatorname{argmax}} \sum_{l: D_l(x)=c} \log \frac{1}{\beta_l} \tag{8}$$

On other hand, to update the Arc-x4 weights of the individual instances as follows:

$$W_{l+1}(i) = 1 + \text{misclassified}(i)^4 \tag{9}$$

A final classification produce a new sample x is given by simple majority voting.

6 Genetic Algorithm

The most basic structure of the GA proposed by Holland [13], begins with a set of possible solutions (population) codified as a chain of bits (called chromosome), later with the use of a method to evaluate the behavior (fitness) of each chromosome, the parents of the next population are determined.

In this work we use the GA proposed by [17] where an m -dimensional chromosome represents all the training set of the m samples. This was accomplished by the binary codification, where a specific training sample was either (1) or (0) considered. This codification is randomly accomplished. Thus, a (chromosome) solution for all training samples marked with 1's is formed. Then the training samples marked with 0's are not part of the subsample.

To reduce the processing time of the GA, in addition to the 0's, some chromosomes are reduced in 20%, that is to say, during the evolutive process, several genes marked with a different value of 0 or 1 were ignored. The initial and the subsequent populations (until 30 epochs) were of 15 chromosomes constituted.

Regarding to the fitness method, the leaving-one-out method was employed and, an *elitist* method select the best solutions in each step and uses these chromosomes to apply de genetic operators: crossover and mutation. The former, consists of the uniform crossover and, the latter, randomly change 10% of the genes in each chromosome. An important aspect is that the best solutions are not included in the next epoch.

When the evolutionary process was finished, the best five solutions of the all epochs are used for build the ensemble.

7 Experimental Results

The results here reported correspond to the experiments over eight real-problem data sets taken from the UCI Machine Learning Database Repository (<http://www.ics.uci.edu/~mllearn>). For each data set, the 5-fold cross-validation method was employed to estimate the classification error: 80% of the available patterns were used for training purposes and 20% for test set. On the other hand, it has to be noted that in the present work, all the base classifiers correspond to the 1-NN (Nearest Neighbor) rule. Table 1 shown the databases used in the experimentation.

Table 1. Databases used

Database	Classes	Features	Training samples	Test samples
German	2	24	800	200
Heart	2	13	216	54
Wine	3	13	144	34
Waveform	3	21	4000	1000
Liver	2	6	276	69
Pima	2	8	615	153
Cancer	2	9	546	137

The ensembles have been integrated by using four well-known resampling methods: random selection with no replacement [5], bagging [2], boosting [4], and Arc-x4 [1] and, two Genetic Algorithm methods; with reduction and without reduction scheme [17]. Only the result of the best technique on each database has been presented. Analogously, for each database, related to the number of subsamples to induce the individual classifiers, that is, the number of classifiers in the system, we have experimented with 5 and 7 elements, and the best results have been finally included in Figure 1. Moreover, the 1-NN classification accuracy for each original training set (i.e., with no combination) is also reported as the baseline classifier.

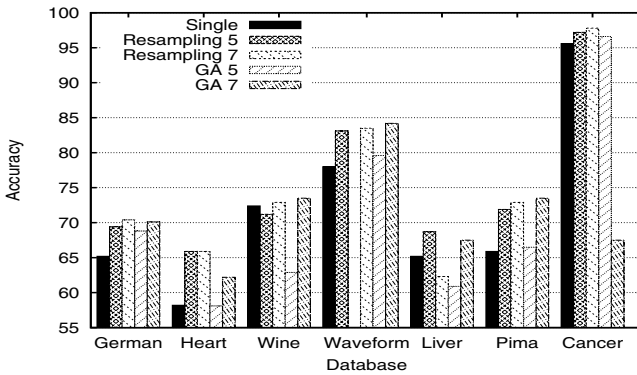


Fig. 1. Overall Accuracy DCS-CONF1

From the results given in the Figures 1, some comments can be drawn. First, it is clear that in all databases the employment of an ensemble leads to better performance than the individual 1-NN classifier. Second, the application of the selection strategy outperforms the combination of classifiers. In fact, we can find the DCS-CONF1 with resampling methods offer higher accuracy than the DCS-CONF1 with GA on all databases using ensembles with seven classifiers. With the Cancer database, differences between the DCS-CONF1 with resampling methods and the rest of the schemes are very significantly.

8 Concluding Remarks

When an ensemble is employed one has to implement some procedure for combining the individual decisions of the base classifiers. Due to some difficulties described above, the majority voting efficiency can become too poor when the performance of the ensemble members is not uniform. Under this practical situation, the selection methods promise to increase the ensemble behavior eliminating the majority voting problems.

In this paper, new methods for dynamic weighting in the framework of DCS methods have been introduced. More specifically, two new methods for determining the confidence intervals has been adapted to be used in a dynamic classifier selection. Experimental results with several real-problem data sets have shown the true benefits of using

the resampling methods over the DCS-CONFI with GA schemes. Results also corroborate that in general, an ensemble clearly outperforms the individual 1-NN classifier.

Future work is primarily addressed to investigate other confidence intervals functions applied to dynamic selection in an ensemble. Within this context, the use of several well-known data complexity measures [5] could be of interest to conveniently adjust the classifier weights. Also, we are interested on develop static selection methods and compare this method with other classic pattern recognition classifiers such as neural networks or Bayesian classifier. On the other hand, the results reported in this paper should be viewed as a first step towards a more complete understanding of the behavior of the confidence intervals procedures on DCS algorithms and consequently, it is still necessary to perform a more exhaustive analysis over a larger number of synthetic and real databases.

Acknowledgments

This work has been partially supported by grants: PROMEP/103.5/08/3016 UAEM-CA-114 and SBI112 from the Mexican SEP, and by the 2703/2008U UAEM project.

References

1. Breiman, L.: Arcing classifiers. *The Annals of Statistics* 26(3), 801–849 (1998)
2. Breiman, L.: Bagging predictors. *Machine Learning* 26, 123–140 (1996)
3. Arlandis, J., Perez, J.C., Cano, J.: Rejection Strategies and Confidence Measures for a K-NN Classifier in a OCR Task. *Universidad Politécnica de Valencia, Valencia, España*, pp. 576–579 (2000)
4. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: *Proc. 13th Intl. Conference on Machine Learning*, pp. 148–156. Morgan Kaufmann, San Francisco (1996)
5. Valdovinos, R.M., Sánchez, J.S.: Class-dependant resampling for medical applications. In: *Proc. 4th Intl. Conf. on Machine Learning and Applications, Los Angeles, CA*, pp. 351–356 (2005)
6. Roli, F., Giacinto, G., Vernazza, G.: Methods for designing multiple classifier systems. In: Kittler, J., Roli, F. (eds.) *MCS 2001. LNCS*, vol. 2096, pp. 78–87. Springer, Heidelberg (2001)
7. Giacinto, G., Roli, F., Fumera, G.: Selection of classifiers based on Multiple Classifier Behaviour. *University of Cagliari Piazza d' Armi, Cagliari Italy* (2000)
8. Kuncheva, L.I.: Switching Between Selection and Fusion in Combining Classifiers: An Experiment. *IEEE Trans. on Systems Man and Cybernetics, Part B-Cybernetics* 32(2), 146–156 (2002)
9. Kuncheva, L.I.: Clustering-and-selection model for classifier combination. In: *Proceedings of the 4th International Conference on Knowledge-Based Intelligent Engineering Systems. Allied technologies (KES 2000), Brighton, UK* (2000)
10. Woods, K., Kegelmeyer Jr., W.P., Bowyer, K.: Combination of Multiple Classifier Using Local Accuracy Estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(4) (1997)
11. Bauckhage, C., Thurau, C.: Towards a Fair'n Square Aimbot Using Mixture of Experts to Learn Context Aware Weapon Handling. In: *Proceedings of (GAME-ON 2004), Ghent, Belgium*, pp. 20–24 (2004)

12. Hartono, P., Hashimoto, S.: Ensemble of Linear Perceptrons with Confidence Level Output. In: Proceedings of the 4th International Conference on Hybrid Intelligent Systems (HIS 2004), pp. 186–191 (2004)
13. Holland, J.: *Adaptation in Natural and Artificial System*. The University of Michigan Press (1975)
14. Turney, P.: Types of cost in inductive concept learning. In: Proceedings of the Cost-Sensitive Learning Workshop at the 17th ICML 2000 Conference, Stanford, CA (2000)
15. Li, J., Li, X., Yao, X.: Cost-sensitive classification with genetic programming. In: Proceedings of IEEE Congress on Evolutionary Computation, vol. 3, pp. 2114–2121 (2005)
16. Krzanowski, W.J., Fieldsend, J.E.: Confidence in Classification: A Bayesian Approach. *Journal of Classification* 23, 199–220 (2006)
17. Diaz, R.I., Valdovinos, R.M., Pacheco, J.H.: Comparative Study of Genetic Algorithms and Resampling Methods for Ensemble Constructing. In: Proceedings of IEEE Congress on Evolutionary Computation (2008)
18. Morales, A.I., Valdovinos, R.M., Sanchez, J.S.: On the Weighted Dynamic Classifier Selection with Local Accuracies. In: Proceedings of the 11th IASTED International Conference Intelligent Systems and Control (ISC 2009), Orlando, USA (2008)
19. Liu, C.L.: Classifier combination based on confidence transformation. *Pattern Recognition* 38, 11–28 (2006)

Optimal Neuron-Controller for Fluid Triple-Tank System via Improved ADDHP Algorithm*

Shaojian Song, Gang Cai, and Xiaofeng Lin

College of Electrical Engineering, Guangxi University,
100, Daxue Road, Nanning, Guangxi 530004, P.R. China
sjsong03@163.com, caigang@mail.gxu.cn, gxulinxf@gmail.com

Abstract. In the present paper, we discussed the implementation of a novel on-line optimal neuron-controller which based on improved Action-Depended Dual Heuristic Dynamic Programming (ADDHP) algorithm, including its schematic diagram, and the training algorithms. This algorithm requires neither an explicit model of controlled system nor the indispensable system performance index ' J '. Compared with standard ADDHP, the proposed method only use the states of present and previous time step to calculate the derivative of the performance function, avoiding predicting the states of next time step, so the model network can be omitted completely. It makes the configuration of the method become simple, more suitable for complex nonlinear systems. The simulation control example is conducted in this paper, and the results show that the proposed ADDHP-based optimal neuron-controller has advantages in fast response, anti-jamming capability and robustness.

Keywords: Improved ADDHP, Optimal neuron-controller, Model-free control, Fluid Triple-tank System.

1 Introduction

Adaptive Dynamic Programming (ADP) was proposed by Werbos [1] and modified by Prokhorov [2]. It is a neural network-based approximation Dynamic Programming (DP) [3]. Because DP is an effective method which can precisely solve such problems: the stochastic, fluctuant and general nonlinear system long-term optimization, but it usually meets "curse of dimensionality" in the practical application. ADP can improve the calculation efficiency and avoid the "curse of dimensionality" by using action-critic structure.

As far as adaptive learning is concerned, ADP links supervised learning with reinforcement learning via critic network. In ADPs, the target data assumed in supervised learning are unknown. Instead, the target data are learned indirectly by

* The work was supported in part by: National Science Foundation of Guangxi Province of China under Grant 0991057; Guangxi natural sciences fund (GSF 0575016); the Science & Research Foundation of Educational Commission of Guangxi Province of China under Grant 2008[27].

optimizing the reinforcement (or critic) in the learning process. We use supervised learning in ADP mainly for training the critic network.

Backpropagation algorithm is a good candidate for supervised learning. Besides, backpropagation is just for passing the derivative of critic between neural networks. Since most of the applications of ADP are control problems, in the following context, we will relate ADP to control of Fluid Triple-tank System.

Suppose that a discrete-time nonlinear time-varying system is described by:

$$x(t + 1) = F[x(t), u(t), t] , t = 0, 1, 2, \dots, l . \tag{1}$$

where the state vector of the system $x \in R^n$ and the control vector $u \in R^m$. Define the performance index or cost-to-go function associated the system in the form:

$$V[x(t)] = \sum_{k=t}^{\infty} \gamma^{k-t} U[x(k), u(k)] . \tag{2}$$

where $U(\cdot)$ is called the utility function, provided a measure of the system performances as a function of the system states or actions. And γ is the discount factor with $0 < \gamma \leq 1$. The objective is to choose a sequence of the control vector $u(k)$ ($k = t, t+1, \dots$), to make the performance index function V to be minimized. Then according to Bellman’s optimal principle [4], the optimal control $u^*(t)$ at time t is the $u(t)$ that achieves to the optimal cost, i.e.

$$u^*(t) = \arg \min_{u(t)} \{ U[x(t), u(t), t] + \gamma V^*[x(t+1), t+1] \} . \tag{3}$$

where $U[x(t), u(t)] + \gamma V^*[x(t+1), t+1]$ is the cost at time t .

The above equation indicates that we should search all the possible states $x(t+1)$, and then find an optimal control sequence $u^*(t+1), u^*(t+2), \dots$, to apply to the system. Unfortunately, the time backward process required for running dynamic programming makes the computation and storage burden become seriously heavy, especially for high order nonlinear systems, i.e., the problem commonly known as “curse of dimensionality”. In the last few years, progress had been made to overcome this problem by building a system called critic to approximate the cost-to-go function V of the dynamic programming in the form of ADPs [5]. The approximate value defined as J , as the output of critic.

The basic structures of ADP [4]-[7] proposed in literature were Heuristic Dynamic Programming (HDP), Dual Heuristic Programming (DHP) and Globalized Dual Heuristic Programming (GDHP). A typical ADP consists of three neural network modules called “action”, “model” and “critic”.

If we omit the model and connect the “action” to the “critic”, this model is called Action-Dependent ADP. The AD version [8] is just use the outputs of the action network as the partial inputs of the critic network, and correspondingly become ADHDP, ADDHP and so on.

In ADPs, the model is not easy to obtain, especially in the complex nonlinear system. Some methods have been introduced to control nonlinear system without the model network, most of which are grounded on ADHDP [6][9]. While different from ADHDP, standard ADDHP still needs a model network in the propagating [5]. The

analysis will be presented in the context. And an improved method based on ADDHP will be presented in this paper. The most advantage of this method is that it only use current and previous states of the system to calculate $J(t)/x(t)$, avoiding predicting the states of next time $t+1$ at time t [10]. In order to illustrate its advanced aspects, a controller example based on this method will be given and applied to Fluid Triple-tank System.

2 Standard ADDHP

Standard ADDHP [11] performs similar operations as DHP, except for the critic network’s output where standard ADDHP approximates the derivative of $J(t)$ with respect to the control actions $u(t)$ and states $x(t)$, denoted by:

$$\begin{cases} \lambda_u = J(t) / u(t) \\ \lambda_x = J(t) / x(t) \end{cases} \tag{4}$$

Both $\lambda_u(t+1)$ and $\lambda_x(t+1)$ are propagated back from next time step $t+1$ and the training of critic in it is to minimize the following error measure:

$$E(t) = \frac{1}{2} \sum_t \sum_i E_i(t)^2 = \frac{1}{2} \sum_t \sum_i \left(\frac{\partial J(t)}{\partial u_i(t)} - \gamma \frac{\partial J(t+1)}{\partial u_i(t)} - \frac{\partial U(t)}{\partial u_i(t)} \right)^2 \tag{5}$$

where $U(t)$ is an utility function, and $i=1, 2, \dots, m$, m is the dimension of $u(t)$.

$$\frac{\partial J(t+1)}{\partial u_i(t)} = \sum_{j=1}^n \lambda_{x_j}(t+1) \frac{\partial X_j(t+1)}{\partial u_i(t)} \tag{6}$$

The value $\lambda_{x_j}(t+1)$ can be obtained by propagating $\lambda_u(t+1)$ back through the action network, which can be stated as follow:

$$\lambda_{x_j}(t+1) = \frac{\partial J(t+1)}{\partial x_j(t+1)} = \sum_{k=1}^m \lambda_{u_k}(t+1) \frac{\partial u_k(t+1)}{\partial x_j(t+1)} \tag{7}$$

The value $\partial J(t)/\partial u_i(t)$ in Equation (5) is obtained by propagating $\lambda_x(t+1)$ back through the model network, so we need a model for updating critic’s weight.

The rule for weight update in action network can propagate critic network’s output λ_u back trough action network, as shown in Fig. 1(b).

From the introduction of standard ADDHP arithmetic, we can find that the method is convenient as the model network is omitted during the training of action network. However, in the training of critic network, the model is still needed to get the derivative information in the feedback calculation.

In the next section, we will introduce an improved ADDHP arithmetic, which only use the states of present and previous time steps to calculate the derivative of the performance function, avoiding predicting the states of next time step, so the model network can be omitted completely.

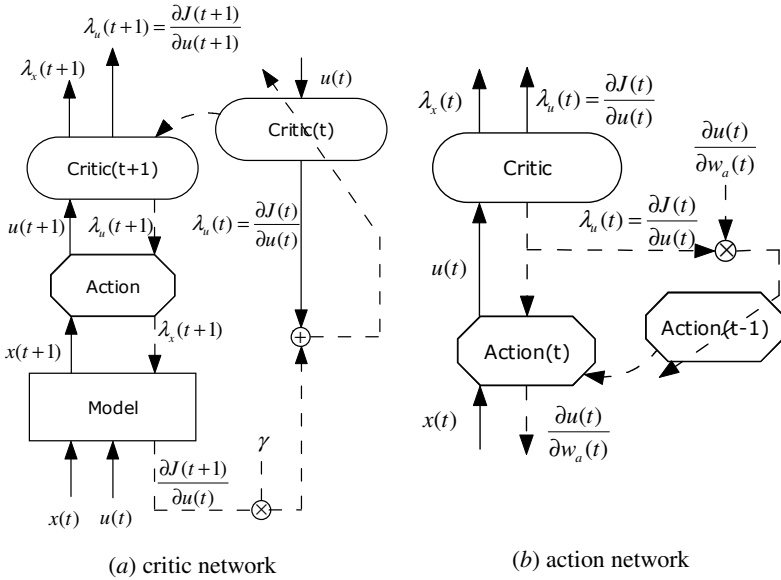


Fig. 1. Training scheme for standard ADDHP

3 The Improved ADDHP Algorithm

An improved ADDHP arithmetic [10] is introduced in this section. Our destination is to build the neural networks which defined the performance index function J to approximate the optimal function V , in Equation (2). If the error between J and V is zero, then we announce that the approximation is accurate. Function J can be formulated as:

$$J[x(t), u(t)] = \sum_{k=t+1}^{\infty} \gamma^{k-t-1} U[x(t), u(t)] . \tag{8}$$

and so, we can obtain the following iterative HJB function:

$$J(t) = \frac{1}{\gamma} (J(t-1) - U[x(t), u(t)]) . \tag{9}$$

The structure of ADDHP expressed in Fig. 2. $\mathbf{x}(t)=[x_1(t), x_2(t), \dots, x_n(t)]$ represents the states vector. $\mathbf{u}(t)=[u_1(t), u_2(t), \dots, u_m(t)]$ is the action vector quantities, the output of action network. The output of critic network is $\mathbf{Q}_i(t)=[Q_1(t), Q_2(t), \dots, Q_n(t)]$, and

$$Q_i(t) = \lambda_i(t) = \partial J(t) / \partial x_i(t) . \tag{10}$$

ω_a and ω_c represent the weight of action network and critic network, respectively. The transfer function of hidden nodes is selected by bi-polar sigmoid function.

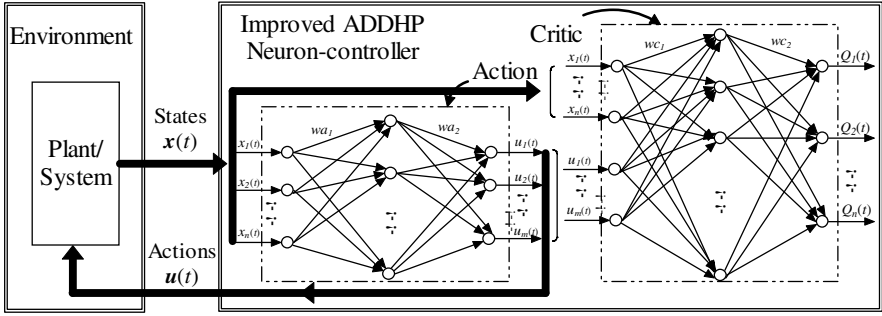


Fig. 2. Schematic diagram of improved ADDHP neuron-controller

3.1 Training for Critic Network

For training critic network, $\lambda^*(t)$ is the value calculated as the “desired output”, and

$$\begin{aligned}
 \lambda^*_i(t) &= \partial J(t) / \partial x_i(t) = \partial \left(\frac{1}{\gamma} J(t-1) - U[x(t), u(t)] \right) / \partial x_i(t) \\
 &= \frac{1}{\gamma} \left\{ \sum_{k=1}^n \lambda_k(t-1) \frac{\partial x_k(t-1)}{\partial x_i(t)} + \sum_{k=1}^n \sum_{j=1}^m \lambda_k(t-1) \frac{\partial x_k(t-1)}{\partial u_j(t)} \frac{\partial u_j(t)}{\partial x_i(t)} \right\} \\
 &= \frac{1}{\gamma} \left\{ \sum_{k=1}^n \lambda_k(t-1) \left[\frac{\partial x_k(t-1)}{\partial x_i(t)} + \sum_{j=1}^m \frac{\partial x_k(t-1)}{\partial u_j(t)} \frac{\partial u_j(t)}{\partial x_i(t)} \right] \right\} \quad (11) \\
 &= \frac{1}{\gamma} \left\{ \frac{\partial J(t-1)}{\partial U[x(t), u(t)]} \left[\frac{\partial U[x(t), u(t)]}{\partial x_i(t)} + \sum_{j=1}^m \frac{\partial U[x(t), u(t)]}{\partial u_j(t)} \frac{\partial u_j(t)}{\partial x_i(t)} \right] \right\} \\
 &= - \frac{\partial U[x(t), u(t)]}{\partial x_i(t)} - \sum_{j=1}^m \frac{\partial U[x(t), u(t)]}{\partial u_j(t)} \frac{\partial u_j(t)}{\partial x_i(t)}
 \end{aligned}$$

The partial derivative team $\partial u_j(t) / \partial x_i(t)$ is computed by back-propagation through the action network and $\lambda_k(t-1)$ can be obtained at time $t-1$. And the error team is calculated in the form:

$$e_c(t) = \lambda^*(t) - \lambda(t) \quad (12)$$

and

$$E(t) = \frac{1}{2} e_c^2(t) \quad (13)$$

The rule for weight update in critic network can be expressed as:

$$\omega_c(t+1) = \omega_c(t) + \Delta\omega_c(t) = \omega_c(t) - l_c \sum_{i=1}^n \frac{\partial E_i(t)}{\partial \omega_c(t)} \tag{14}$$

where ω_c denotes the weight of the critic network, $0 < l_c \leq 1$ is also a given learning rate. The flow chart of critic network for this scheme is shown in Fig. 3(a). From (11), we can see that for obtaining $\lambda^*_i(t)$ at time t , we just need to conserve all the $\lambda_k(t-1)$, $k = 1, 2, \dots, n$, instead of computing or searching all the $\lambda_k(t+1)$ at next time step, and the model network which predicts the state for the next time step is omitted.

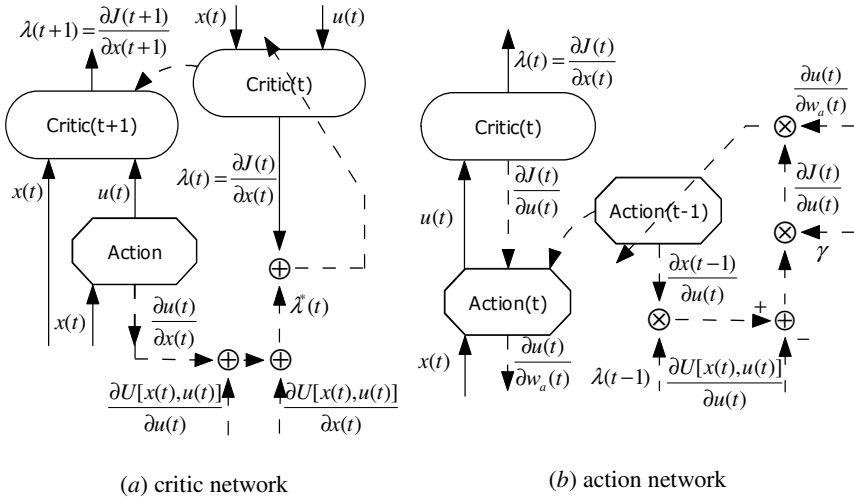


Fig. 3. Training scheme for the improved ADDHP

3.2 Training for Action Network

The main objective of action network is to generate a sequence control signal $u(t)$, $u(t+1)$, $u(t+2)$, ... to make the performance index to be optimal.

For the action network, the weights are updated by the following training rules:

$$\omega_a(t+1) = \omega_a(t) + \Delta\omega_a(t) = \omega_a(t) - l_a \sum_{k=1}^m \frac{\partial J(t)}{\partial u_k(t)} \cdot \frac{\partial u_k(t)}{\partial \omega_a(t)} \tag{15}$$

where ω_a denotes the weight of the action network, $0 < l_a \leq 1$ is the learning rate, and m is the dimension of the control signal $u(t)$. Expanding $\partial J(t)/\partial u_k(t)$ in the form:

$$\frac{\partial J(t)}{\partial u_k(t)} = \partial \left\{ \frac{1}{\gamma} (J(t-1) - U[x(t), u(t)]) \right\} / \partial u_k(t) \tag{16}$$

and

$$\frac{\partial J(t-1)}{\partial u_k(t)} = \sum_{i=1}^n \frac{\partial J(t-1)}{\partial x_i(t-1)} \cdot \frac{\partial x_i(t-1)}{\partial u_k(t)} = \sum_{i=1}^n \lambda_i(t-1) \cdot \frac{\partial x_i(t-1)}{\partial u_k(t)} \quad (17)$$

where n is the dimension of the state vector, the $\lambda_i(t-1)$ is the output of critic network in time $t-1$, and $\partial x_i(t-1)/\partial u_k(t)$ can be obtained from the action network(Fig. 3(b)).

From the introduce and analysis above, the improved ADDHP neuron-controller needn't model during its training progress, not only for the action network but also the critic network, so the controller is a true model-free controller, and the proposed method in this paper will emerge superior performance in next section.

4 Simulation and Results

4.1 The Plant

In order to illustrate the effectiveness of the proposed method for the on-line optimal control of complex nonlinear system, a fluid triple-tank system [12] was selected as plant, which is a strong coupling, nonlinear MIMO system.

The model of fluid triple-tank system [13] for simulation is

$$\begin{cases} dh_1 / dt = (Q_1 - Q_{13}) / A \\ dh_2 / dt = (Q_2 + Q_{32} - Q_{20}) / A \\ dh_3 / dt = (Q_{13} - Q_{32}) / A \end{cases} \quad (18)$$

where h_i is the level of i th tank (Unit: ml); Q_i represents inflow flux generated by i th pump (Unit: m^3/s); A is the cross-sectional area (Unit: m^2); and Q_{ij} represent the interactive flux between i th and j th tank (Unit: m^3/s) and obtained by Torricelli Rule.

$$\begin{cases} Q_{13} = az_1 S_n \operatorname{sgn}(h_1 - h_3)(2g | h_1 - h_3 |)^{1/2} \\ Q_{32} = az_3 S_n \operatorname{sgn}(h_3 - h_2)(2g | h_3 - h_2 |)^{1/2} \\ Q_{20} = az_2 S_n (2gh_2)^{1/2} \end{cases} \quad (19)$$

where g is the gravity acceleration (unit: m/s^2), $\operatorname{sgn}(x)$ is the function to get the sign of variable x , az_k is the efflux coefficient (correction factor, unit: 1) related to k th pot.

Table 1. Fluid triple-tank system's physical parameters

$A=0.0154m^2$	$S_n=5 \times 10^{-5}m^2$	$Q_{1max}=100ml/s$
$h_{max}=62cm$	$g=9.81m/s^2$	$Q_{2max}=100ml/s$
$az_1=0.490471$	$az_2=0.611429$	$az_3=0.450223$

4.2 The Design for the Neuron-Controller

Both the critic network and the action network are simple feedforward networks, each having one hidden layer of bipolar sigmoid nodes. Their architectures are 2-8-1 and

1-8-1, respectively. Outputs of both nets also have linear neurons. The error of the level (practical value subtracted from set point) was selected as the state variable $x(t)$. Both $x(t)$ and control output $u(t)$ are all scaled to $[-1,1]$. The learning rate of the critic network and action network are given as $l_c = 0.4$, $l_a = 0.3$, and the maximum of cycle-index N_c , N_a are assigned as $N_c = 100$ and $N_a = 50$, respectively. The utility function $U(t)$ of the neuron-controller is defined as:

$$U(t) = x(t)x(t)^T + u(t)u(t)^T \quad (20)$$

4.3 Results of Simulation

In this section several experiments is conducted. And the results are shown in Fig. 4. In these figures, all the Y coordinate values are standardized (normalized) value. The fluid position initialization value, to tank T_1 , tank T_2 , tank T_3 , is $h = [0.01 \ 0.01 \ 0.01]$.

In Fig. 4(a), the control object is the level of tank T_1 . At the beginning, the control reference value is 0.2. After 10 time steps the practical level has reach the setpoint with very small overshoot. At time step 300, we change the control reference value to 0.5, the system rapidly reach the new setpoint, similarly. At time step 200 and 500, the disturbance was introduced in to the system.

The disturbance or noise is generated by a random function, scaled to $[-1,1]$, and the seed of it is depended on PC clock.

We can find the action made by action network in response to a stimulus is immediate, and the fluid position come back to the desired level promptly and still keep in balance.

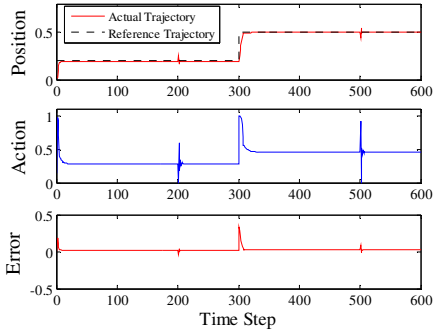
In Fig. 4(b), the control reference value was changed to a function:

$$Obj(k) = \begin{cases} 0.004 \times k & , k \leq 100 \\ 0.4 + 0.2 \times \sin[(k - 100)\pi / 200] & , k > 100 \end{cases} \quad (21)$$

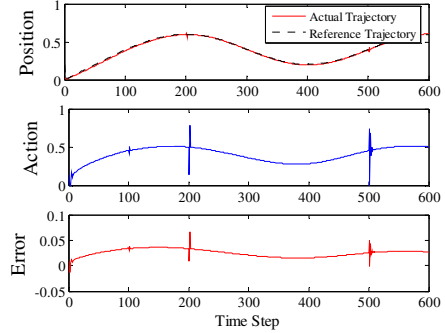
The result shows that this controller has good performance in tracking control, as well, despise of the diminutive burr appeared at the corner of the object function.

In Fig. 4(c) and Fig. 4(d), the triple-tank level control is conducted, i.e., the controlled object was changed to the level of tank T_2 . And the disturbance was introduced into the system at time step 250 and 750 by random function, as well, when the system can still get ideal performance.

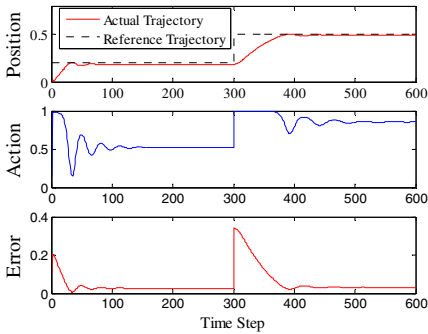
The level of triple-tank level control reach to the object stably with 30-80 steps, which is more than the single tank level control. The overshoot, oscillation and stable error appear in the controlled progresses, which are declining to the small range in few steps. With the artificial introduction disturbance, through several time steps the system could achieve stable again. The results show that the controller can deal with strong coupling and hysteresis problem and have a good performance.



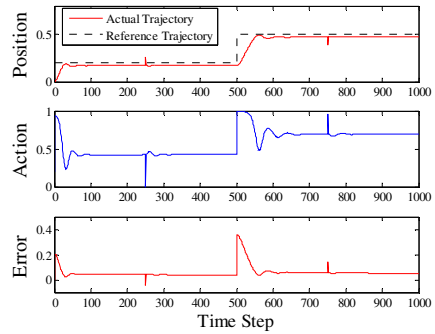
(a) The set point control for single tank with disturbance introduced at step 200 and 500



(b) The tracking control for single tank with disturbance introduced at step 200 and 500



(c) The set point control for triple-tank without disturbance



(d) The set point control for triple-tank with disturbance introduced at step 250 and 750

Fig. 4. The simulation of the improved ADDHP controller

5 Conclusion

In the paper, an improved ADDHP algorithm has been introduced in detail, from architecture to the algorithm, implementing, and finally given experiments of simulation. Compared with the traditional DHP or ADDHP, the improved ADDHP algorithm has following advantages. The configuration is simpler, and it is a true model-free controller, because it only needs the states of present and previous time step to calculate the derivative of the performance function, avoiding predicting the states of next time step, so the model network can be omitted. And, the method is directly built on the derivative of the performance index which can behave more sensitivity. The experiment results indicate that the proposed optimal neuron-controller based on improved ADDHP has the following characteristics: high control precision, fast response, strong robustness and anti-jamming capability.

Therefore, improved ADDHP neuron-controller, as an intelligent model-free approach, has advantages in on-line optimal control for complex, nonlinear, especially

modeling uncertainty systems, which has great prospect for further development and industrial application in the future.

References

1. Werbos, P.J.: Advanced forecasting methods for global crisis warning and models of intelligence. *J. General System Yearbook* 22, 25–38 (1977)
2. Prokhorov, D.V.: Adaptive Critic Designs and their Applications. Thesis. Texas Tech University, TX (1997)
3. Bertsekas, D.P.: Dynamic Programming and Optimal Control. J. Athena Scientific, Belmont, MA (1995)
4. Bellman, R.E.: Dynamic programming. Princeton University Press, NJ (1957)
5. Danil, P., Don, W.: Adaptive Critic Designs. *IEEE Trans. on Neural Networks* 8, 997–1007 (1997)
6. Jennie, S., Wang, Y.T.: On-Line Learning Control by Association and Reinforcement. *IEEE Trans. on Neural Networks* 12, 264–276 (2001)
7. John, J.M., Chadwick, J.C., George, G.L., Richard, S.: Adaptive Dynamic Programming. *IEEE Trans. on Systems, Man, and Cybernetics-Part C: Applications and Reviews* 32, 140–153 (2002)
8. Liu, D.R., Xiong, X.X., Zhang, Y.: Action-Dependent Adaptive Critic Designs. *IEEE Neural Networks Proceedings*, 990–995 (2001)
9. Park, J.W., Harley, R.G., Venayagamoorthy, G.K.: Dual heuristic programming based nonlinear optimal control for a synchronous generator. *Engineering Applications of Artificial Intelligence* 21, 97–105 (2008)
10. Zhang, H., Wei, Q., Liu, D.: On-line learning control for discrete nonlinear systems via an improved ADDHP method. In: Liu, D., Fei, S., Hou, Z.-G., Zhang, H., Sun, C. (eds.) *ISNN 2007. LNCS*, vol. 4491, pp. 387–396. Springer, Heidelberg (2007)
11. Shiah, C.: Abstract of the Dissertation Natural Adaptive Critic Design for Control and its Parallel Implementation. State University of New York, New York (2002)
12. Song, S., Li, J., Lin, X.: Action-dependent heuristic dynamic programming for level control of three tanks. In: *Proc. Chin. Control. Conf.*, Kunming, China, CCC, July 16–18, pp. 468–4739 (2008)
13. Zhou, D.: Introduction of Adaptive Control on Non-linear Systems, pp. 53–55. Tsinghua University Press, Beijing (2002)

Method of Learning for Life Pet Artificial

Rodolfo Romero Herrera, Francisco Gallegos Funes,
and Antonio Gustavo Juárez Gracia

Computing Higher School (ESCOM), School of Engineering and Electrical (ESIME),
National Polytechnic Institute, (IPN), México D.F.
rromeroh@ipn.mx

Abstract. The present article shows the implementation of reinforcement learning, using the equation of the emotional intensity in a virtual pet for an artificial living environment interacting with the computer user. The pet looks like a very natural because of the learning algorithm proposed, which is based on a neural network. Learning depends on motivation given to the pet by its owner. The equation of emotional intensity gives values which allow to feed the neural network and thus generating learning.

1 Introduction

The purpose of this branch of artificial intelligence is to develop techniques that allow computers to learn [1]. From a more concrete way, to create programs capable of generalizing a behavior from non-structured information provided in the form of examples, an induction process of knowledge [1].

The human being performs an automatic learning in a natural way because it is a procedure so simple that no account is given of how the individual is and what it implies, but the machines must indicate how to learn, because if it is not gotten that the machine develops its skills then it won't be possible to get to stick out a learning process, but a repetitive sequence. In machine learning can be three types of knowledge:

- 1 Growth: the one which is acquired from what surrounds us, which saves the information in the memory as if it were footprints.
- 2 Restructuring: the knowledge to interpret the individual thinks and generates new knowledge which is called restructuring.
- 3 Adjustment is obtained by generalizing multiple concepts or generating their own.

Different machine learning algorithms which are grouped into taxonomy based on the output of the same. Some types of algorithms are:

- Supervised learning; in which the algorithm produces a function that establishes a correspondence between inputs and desired outputs of the system.
- Unsupervised learning [2], where the whole modeling process is carried out on a set of examples consisting only of inputs to the system.
- Reinforcement Learning [3] [4] [5] in this case, the algorithm learns watching the world around him. Your information is input or feedback regulation feedback you get from the outside world in response to their actions.

- Transduction that is similar to supervised learning, but does not construct an explicit function. Try to predict the level of future examples based on the input, their categories and examples to new system.
- Learning multitasking: it consists of learning methods that use previously learned knowledge by the system to cope with problems similar to those already seen.

One of the most appropriate methods for learning are neural networks [6] [7], which consist of a set of entities that are related through weights and transfer functions, for learning to change their weight. Many of the algorithms above are performed with neural networks.

1.1 Cathexis

Cathexis is a model for distributed generation of emotions and their influence on the behavior of autonomous agents [8] [9]. Although its main purpose is to model different aspects of the generation of emotions, it also provides motivation for other simple models and an algorithm for selection of activities [10]. In this model, mood and the emotions are modeled as a network composed of proto-called emotional systems specialists. Each proto-specialist represents a family of specific emotions, and constant monitoring of the external and internal stimuli that can generate such emotion [10]. An emotion in the system activates when the input stimuli that cause the proto-specialist exceeds certain threshold. The proto-specialists communicate with one another so that it can generate an emotion or inhibit other [11].

In cathexis, of emotions, each family is represented by a primary emotion, where that is a basic emotion that differ in important respects [10]. However, each of the basic emotions can occur with different levels of intensity and emotion can have several active at one time, which can increase the set of affective states that can be modeled.

To know when to generate an emotion and that emotion should be, it is necessary to evaluate and interpret a set of events known as generators of emotions [10]. Cathexis are seen in both the cognitive emotions as generators of non-cognitive [11]. These generators are divided into four categories:

- Neuronal: they include all the neuro actives agents who can generate excitement, and are caused by hormones, sleep, such environmental conditions, etc. [12].
- Sensory and motor: cover all sensory and motor processes [12]
- Motivational: It was composed by instincts (like hunger or thirst) and other emotions [12].
- Cognitive: Includes all the skills that can generate emotions, such as interpretation of events, comparisons, or memories [12].

As mentioned earlier, each basic emotion can be in different levels of intensity. This level can be affected by many factors, including the level of pre-excitation, the contribution of different generations of emotion, and interaction with other emotions (excited or inhibitors) [11]. In the following equation represents the intensity level for an emotion.

$$I_{et} = \chi \left(\Psi(I_{et-1}) + \sum_k L_{ke} + \sum_l G_{le} \cdot I_{lt} - \sum_m H_{me} \cdot I_{mt} \right) \quad (1)$$

Where I_{et} is the intensity of emotion and at time t ; I_{et-1} is the previous value, $\Psi()$ is the function that represents the decrease of the emotion e [13] [14] [15]. L_{ke} the value of the generator K , where K runs through all of the generators and excitement; G_{ie} is the gain that he applies to the excitement and emotion, I_{it} is the intensity of emotion l in the time t ; H_{me} is loss that emotion m applied to the emotion e ; I_{mt} is the intensity of emotion m in the time t , and $X()$ is the function that requires the emotion e take a value between 0 and its saturation.

1.2 System Behavior

The system behavior is a distributed system composed of a network of behaviors competes for control of the agent [10]. The decision of what should be active based on the value of each, to be re calculated in each cycle. Each behavior is composed of two main components: An expressive and component experience [11].

- Component expressive: it includes facial expression, body posture and vocal expression, which are responsible for showing the emotional state of the agent [11].
- Component of experience. Can be identified as a motivation level that manifests as readiness to act, selection of resources, learning and emotional state [11]. Consider the following aspects:

- ❖ Motivation: It is the leading cause of behavioral activation, but when a behavior is active can also affect different systems affect them and the motivational levels of instincts and emotions [16].
- ❖ Change Action: the changes in this trend are the main aspect of the response to an event with emotional meaning. Cathexis in the trend of action is modeled by the behavior, which usually result from emotions [16]
- ❖ As mentioned before, behaviors compete for control of the agent. The competition is based on the value of each behavior that updates in each cycle and depends on several factors (called liberators) including motivation and external stimuli [10]. The value of a behavior is modeled as the sum of the values of all relevant releases for that specific behavior and the values depend on the liberators of their own nature. In each cycle, the selection of an activity is conducted in accordance with the following algorithm [10]:
 1. Internal and environmental variables are listed.
 2. The values of all motivations (instincts and emotions) are updated according to the equation of intensity.
 3. The values of the behaviors are updated based on external stimuli, and internal motivations

Behavior with the maximum value becomes the active behavior.

2 Learning Module

The objective of this module is that the pet is able to learn sequences of known activities and classified according to the reinforcement generated. Its data requirements are:

- Get the enhancement caused by sequence of activities
- Modify the weights of the network according to the reinforcement obtained.
- Store the modified weights.

2.1 Description of the Classes That Make Up the Learning Modules

The learning unit works by means of a neuronal network, that allows to the modification of its weights using an algorithm of learning by reinforcement. The algorithm used has been designed specifically for this project, because the reinforcement learning applied to neural networks has not been substantial progress to date [17] [18], and no one is applied correctly to your system . The classes used in this module, with its own description, are shown in Figure 1:

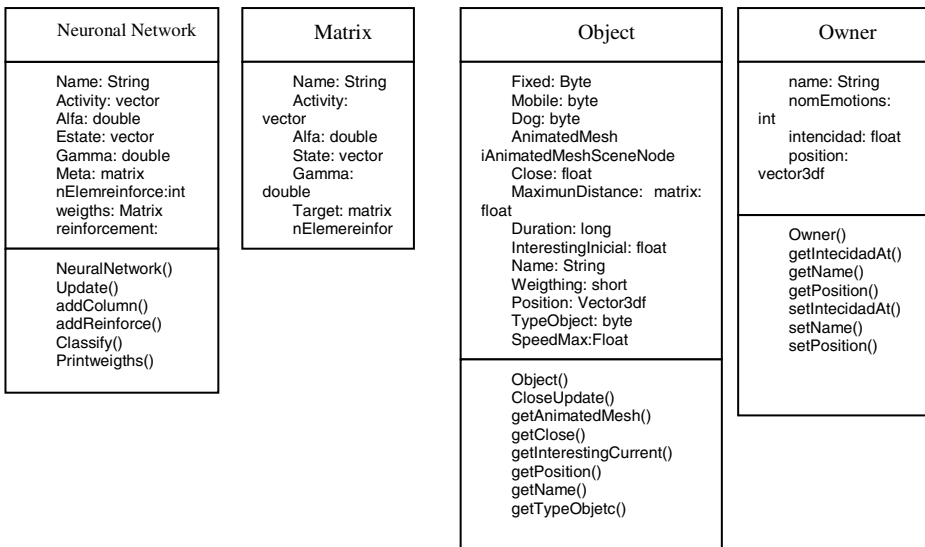


Fig. 1. Learning Classes for its programming

- **Neural Network.** The neural network of each pet is an instance of this class. Through her, the dog can calculate the enhancement obtained in the sequence of activities carried out, and change the matrix of weights for learning. The algorithm that allows updating the array weights are some references I Reinforcement learning in tables [19] [20] [21] [22].

- **Matrix.** The objects of this kind are normal matrices. But for its size is dynamic and able to perform operations such as creating a class with appropriate methods.
- **Object.** Represents all objects that can appear in the system and form part of the user interface. It provides operations to the absolute position and relative size of the pet has influence in their learning.

- Owner. Any pet in the system can have an owner that is associated directly with learning. The emotions on this subject are part of the entry of the neural network, so any change in them represents a reinforcement for the pet.

2.2 Sequence Diagram Learning

Learning is done through a neural network to be trained during the implementation of the system. This training is given by the analysis of the incentives provided to pet and changing the weights of the network from many different stimuli. The learning stage begins when an active one of your pet behavior, and as a result of the user of the system delivers a stimulus to the pet. The stimulus is interpreted by the dog to see if it is positive or negative, and is sent to a neural network as an example to make it a learning iteration. Since this procedure can simulate the learning of the pets, which requires dedicated training by the user.

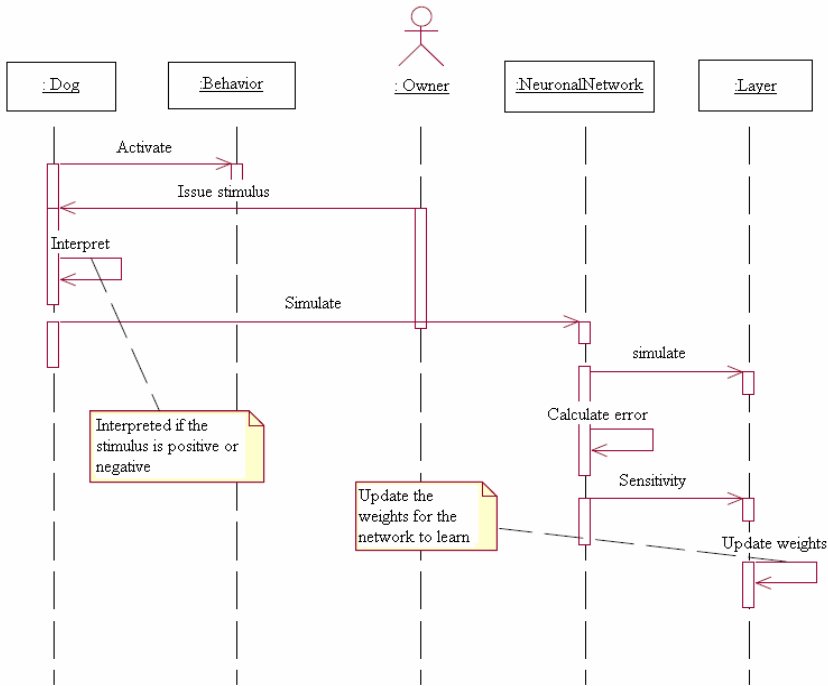


Fig. 2.

3 Learning

Learning to pet via a reinforcement learning algorithm for neural networks adapted to modify the probability that a behavior is selected from a specific state, given by the values of emotions, instincts, user's emotions and closeness to the objects in the environment.

The importance of the neural network is that, since it is obtained the values for each behavior by means of a competitive transfer function, using as weights matrix of a set of values taken from the chromosome of pet items. This weight matrix is modified in each iteration of the system, so that the behavior tends to change towards the direction favorable to the reinforcements obtained.

However, for learning is part of the system, it is necessary to make adjustments Cathexis entering its final stage a neural network enabling it to changing the weights. That is why, in the algorithm described in the generation of emotions and behavior of individual selection, it creates a vector with the intensities of the objects involved in learning and transfer to the neural network, then take the output of the and updating the same values of behavior. What really makes this part is to replace the behavior of Cathexis selected by the network, which maintains the same principle but allows the modification of weights.

The output of the network is obtained from a relatively simple way, it is only necessary to evaluate the input of the network through its transfer function. But now the weight, which makes the learning, is a little more complicated. The following describes the algorithm.

1. Stores the initial state, consisting of entries in the vector of the network.
2. Is obtained out of the network from the matrix of weights and the state are stored, through the competitive transfer function.
3. The result is sent from the network to the Cathexis system, but is also stored.
4. Were obtained following the rule, as amended by the motivations of Cathexis.
5. Strengthening the vector is obtained from the initial state, the state matrix and the resultant goal, which is the set of values that the pet looking for their emotions and instincts and emotions of the user.
6. It is obtained and stored the strengthening the values of vector averaging reinforcement.
7. Repeat the algorithm until the number of stored states for appropriate learning activities in sequence.

The formulas used in the algorithm are:

- a) To get out of the network:

$$a = C(wp) \tag{2}$$

Where p is the vector of inputs or state of the pet, a is a vector of outputs or action to be performed, w is the matrix of weights, $C()$ is a competitive function, which is equal to one for the neuron with an n the larger, and zero for other neurons.

- b) To obtain the vector reinforcement:

$$Vr = |m-pi| - |m-pj| \tag{3}$$

Where Vr is the vector of reinforcements, pi is the initial state, pj is the final state, m is the matrix final.

- c) To calculate the individual reinforcement:

$$r = \frac{1}{n} \sum_{k=1}^n v_{rk} \tag{4}$$

Where n is the number of elements in the vector for reinforcements

d) To calculate the total reinforcement

$$r_i = \sum_j^j r \tag{5}$$

Where j is the number of activities in sequence.

e) To update the weights of the network

$$w = w + 0.01\alpha r_i \sum_{i=1}^k \frac{1}{\gamma^n} a_i p_{i-1}^T \tag{6}$$

Where w is the matrix of weights, 0.01 is the adjustment factor, used to prevent the normalization of the weights of the network, α is the speed of learning, rt is the total reinforcement, γ is the attenuation coefficient for reinforcements no immediate, a_i is an activity that produces state change; p_i is a state stock, k is the number of changes of states involved in the update.

4 Results

The graphs of Figures 3 and 4 show the difference between a pet newborn, which has not been trained by the user and the pet after being trained.

The situation presented is that the pet wishes to defecate near the entrance of the house and move away slowly, the user or master does not want the dog pooped there.

In the graph of Figure 1, the pet is not trained, it is noted that the values of instinct "evacuation 2" and defecation behavior are the same all the time and does not take into account the proximity to that found in the door of the house. Therefore, no matter his dog defecate position. When the user begins to show his wrath (with buttons

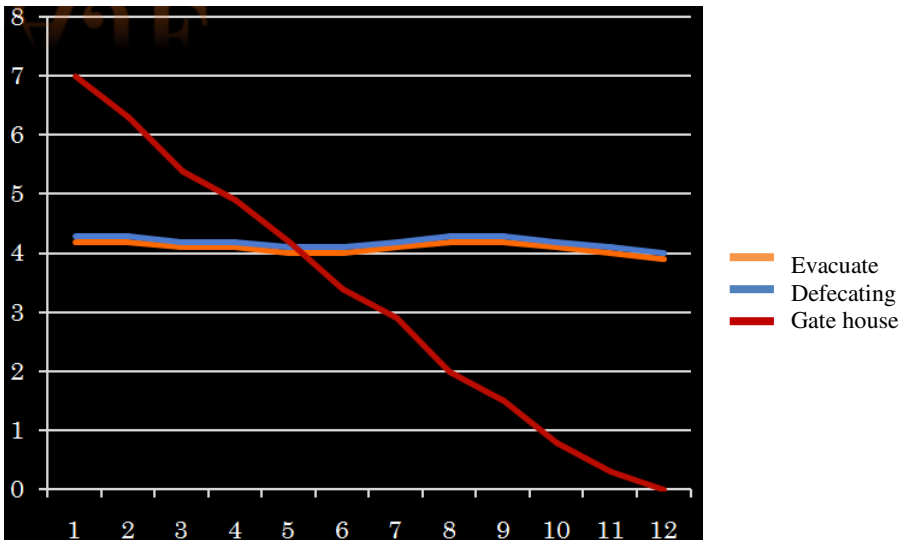


Fig. 3. Graphics before learning

designed for this in the programming), the matrix of weights of the neural network is updated, ie, the pet will learn that what he did was wrong, so that the passage of time likely do so again will decrease until it is 0, that is, has learned not to defecate there.

Graph of the Figure 4, which shows that the values of instinct "evacuation" and defecating behavior will also constitute close to that found on the door. When the pet is away from the door, the values of the instincts and behavior are equal, so she can defecate after that time depending on the level of need you have and if it is greater than the other behaviors.

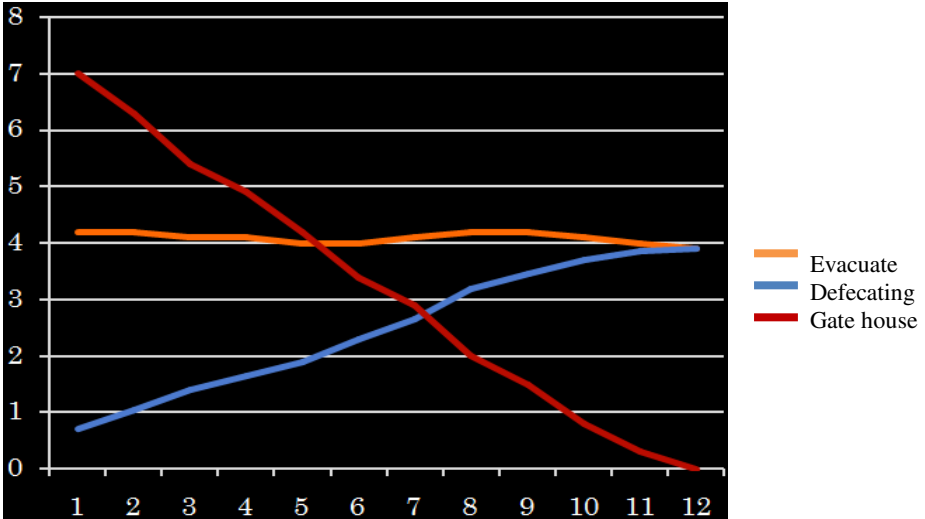


Fig. 4. Graph after learning

The work presented provides a system that allows live with a virtual pet, the pet taking advantage of changes based on learning from reinforcements, other systems consider the emotional part of the mascot, but not learning in the work of Stephen a [22]. Implement other creatures with artificial life, but not an affected system [23]. Due to these considerations, the work presented contributes both the emotional and learning.

5 Conclusions

The selection of cathexis to implement this part of the system was a wise choice in various directions, which provides important advantages over the handling of temperamental mood and style (which differentiates the pets). Another advantage presented by the fact composed objects by weights, which can treat the system as a neural network to which you can learn by modifying their weights.

The learning of the pet is not fully supervised or unsupervised, and that although the master may act as a teacher there are activities that the pet learns only from the stimuli it receives. However, stimulating the learning response is not appropriate

because the problem is not based on classification of patterns. That is why reinforcement learning is selected. The algorithms are based on existing tables, which means you can only use a finite set of states and are not applicable to neural networks, which, because they handle continuous values in each of its entries are an infinite number of possible states. When using tables based learning reduces the chances of behavior and spent more resources. Reinforcement learning based on neural networks is more suitable, which is an algorithm that is designed especially for the learning of the pet, taking only some of the ideas related to learning activities in sequence to maximize the strengthening future.

As advantage, the model fits the animal nature, which must learn some things, known as instinctive activities; the network is only responsible for modifying the weights without affecting the performance of these activities. In addition, through the network to modify the weights to learn, realism does the pet in the sense that it has not learned, seems confused, not know what to do to meet certain needs. This gives the appearance of development in terms of behavior is concerned, what is normal part of growing up. Finally, allow the user to the iteration of learning, as it may influence the reinforcements obtained by the pet.

References

1. Langley, P., Simon, H.A.: Applications of machine learning and rule induction. In: AFIPS Joint Computer Conferences, Communications of the ACM, pp. 54–64. ACM, New York (1959)
2. Aprendizaje Asociativo,
<http://ohm.utp.edu.co/neuronales/Capitulo2/Asociativas/AntecedentesAs.htm>
3. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (2005)
4. Russell, S.J., Peter, N.: Inteligencia Artificial. Un Enfoque Moderno. Pearson, España (2004)
5. Con, K.: Supervised Reinforcement Learning - Application to an Embodied Mobile Robot (Paperback). VDM Verlag DR Muller, USA (2007)
6. Pardern, X.: Introducción a las Redes Neuronales, RED Científica, Ciencia, Tecnología y Pensamiento (2009)
7. Flóres López, R., Fernández, J.M.: Fernández, Las Redes Neuronales Artificiales, NetBiblo S.L., España (2008)
8. Juli, V., Botti, V.: Agentes Inteligentes: el siguiente paso en la Inteligencia Artificial, Novatica (2000)
9. Arafa, Y., Dionisi, G., Fehin, P., Martin, S., Pitt, J., Witkowski, M.: Deliverable D3 Agent-to-Human interaction - principles, Mappa Swedish Institute of Computer Science, Kista, Sweden (2000)
10. Velásquez, J.D.: Modeling emotions and other motivations in synthetic agents. American association for artificial intelligence (1997)
11. Velásquez, J.D.: Cathexis: a computational model fro the generation of emotions and their influence in the behavior of autonomous agents. S.M. Thesis. Departamental of electrical engineering and computer science, Massachusetts institute of technology (1996)

12. Velásquez, J.D.: A Computacional Framework for Emotion-Based Control. MIT Artificial Intelligence Laboratory (1998)
13. DelValle Sierra, J.: Funciones exponencial y logaritmica,
<http://huitoto.udea.edu.co/Matematicas/2.1.html>
14. Juan Antonio Cuellar Carvajal, Matemáticas IV: Relaciones Y Funciones, McGraw-Hill Interamericana, México (2006)
15. Finan, M.B.: Arkansas Tech University,
<http://syssci.atu.edu/math/faculty/finan/2243/business19.pdf>
16. Velásquez, J.D.: When robots weep: a computational approach to affective learning, Massachusetts Institute of Technology Cambridge, MA (2007)
17. Sehad, S., Touzet, C.: Reinforcement Learning and Neural Reinforcement Learning; Network. In: Verleysen, M. (ed.) ESANN 1994. D-Facto publication, Bruxelles (1994)
18. Code for Neural Networks and Reinforcement Learning; Colorado State University, Computer Science Department,
<http://www.cs.colostate.edu/~anderson/code>
19. Haykin, S.: Neural Networks. A Comprehensive foundation. Prentice Hall, USA (1999)
20. Halicin, U.: Reinforcement learning in random neural network for cascaded decisions. Journal of Biosystems, Enero 1997 40(1&2), 83–91 (1994)
21. Williams, R.J.: Simple Stastical Gradiente- Following Algorithms for Connectionist Reinforcement Learning. Maching Learning. Springer Link (2005)
22. Grand, S., Cliff, D., Malhotra, A.: Creatures: Artificial Life Autonomous Software Agents for Home Entertainment (1996)
23. Shibata, T., Yoshida, M., Yamato, J.: Artificial emotional creature for human-machine interaction. System Man and Cybernetics, USA (2002)

An Sliding Mode Control for an Elbow Arm

Jose de Jesus Rubio, Jaime Pacheco, and Gerardo Villegas

Instituto Politecnico Nacional - ESIME Azcapotzalco
Seccion de Estudios de Posgrado e Investigacion,
Av. de las Granjas 682, Col. Sta. Catarina. Azcapotzalco, México D.F.
jrubioa@ipn.mx

Abstract. In this paper it is proposed an sliding mode control for an elbow arm. It is proven that the closed loop system of the sliding model control applied to the model of the elbow arm is asymptotic stable. It is given a simulation.

1 Introduction

Various modified PD control schemes for robots and their successful experimental tests have been published [9] and [10]. But there exist one main weakness in the PD control: Due to the existence of gravity forces, the PD control cannot guarantee that the steady state error becomes zero [2]. Since the friction and gravity of robot influence the steady and dynamics properties of the PD control. Global asymptotic stability PD control was realized by pulsing gravity compensation in [11]. If the gravity is unknown, neural networks can be applied. In [7] the author used neural networks to approximate the whole nonlinearity of robot dynamic. with a neuro feedforward compensator and a PD control, he can guarantee good track performance. The approximation errors of neural identification for the gravitational force and friction can be eliminated by a discontinuous switching control law [14]. When the parameters in gravitational torque vector are unknown, adaptive PD control with gravity compensation was introduced by [12]. SP-ID controller can be used in set-point control without any knowledge of the gravitational force [1]. In addition, the gravitational force is well structured (though uncertain) and hence adaptive SP-D controller can also be used to deal with the uncertainties [13]. PID control does not require any component of robot dynamics in its control law, but it lacks of global asymptotic stability [3]. By adding integral actions or computed feedforward, global asymptotic stability PD control are proposed in [2] and [4]. There is an interesting result given by [15] where a PD controller is proposed and a neural network is used to compensate gravity and a high gain observer is applied to estimate joint velocities, but the neural network is slow to compensate the gravity and sometimes it is necessary to get a fast behavior of the controller. That is way in this paper only is considered a constant value as an estimation of the gravity letting to get asymptotic stability of the controller.

In this paper it is proposed an sliding mode control for an elbow arm. It is proven that the closed loop system of the control and the model of the elbow arm is asymptotic stable. It ais given a simulation result.

2 Preliminaries

The main concern of this section is to understand some concepts of robot dynamics.

The equation of motion for the constrained robotic manipulator with n degrees of freedom, considering the contact force and the constraints, is given in the joint space as follows:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (1)$$

Where $q \in R^{n \times 1}$ denotes the joint angles or link displacements of the manipulator, $M(q) \in R^{n \times n}$ is the robot inertia matrix which is symmetric and positive definite, $C(q, \dot{q}) \in R^{n \times n}$ contains the centripetal and Coriolis terms and $G(q)$ are the gravity terms, τ denotes the control inputs. We define the following two states as follows:

$$\begin{aligned} x_1 &= q \in R^{n \times 1} \\ x_2 &= \dot{q} \in R^{n \times 1} \end{aligned} \quad (2)$$

Then (2) can be rewritten as:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= H(x, \tau) \end{aligned} \quad (3)$$

Where $H(x, \tau) = M^{-1}(x_1) [\tau - G(x_1) - C(x_1, x_2)x_2]$, $x = [x_1, x_2]^T$.

Property 1. The inertia matrix is symmetric and positive definite, i.e. [5], [9]

$$m_1 |x|^2 \leq x^T M(x_1)x \leq m_2 |x|^2$$

Where m_1, m_2 are known positive scalar constant.

Property 2. The centripetal and Coriolis matrix is skew-symmetric, i.e., satisfies the following relationship [5], [9]:

$$x^T \left[\dot{M}(x_1) - 2C(x_1, x_2) \right] x = 0 \quad (4)$$

And the centripetal and Coriolis matrix also satisfy [5], [9]:

$$\|C(x_1, x_2)x_2\| \leq k_c |x_2|^T \quad (5)$$

Where $k_c \in R^n$.

The normal PD controller is:

$$\tau = -K_p \tilde{x}_1 - K_d \tilde{x}_2 \quad (6)$$

Where $\tilde{x}_1 = x_1 - x_1^d$ and $\tilde{x}_2 = x_2 - x_2^d$, K_p and K_d are positive definite, symmetric and constant matrices.

3 Sliding Mode Control with Gravity Compensator

In order to simplify the proof procedure, we only consider regulation case, i.e. the desired velocity is $x_2^d = 0$, so the sliding mode control with gravity approximation is:

$$\tau = -K_p \tilde{x}_1 - K_d x_2 + \hat{G} - K \operatorname{sign}(x_2) \tag{7}$$

Where $\tilde{x}_1 = x_1 - x_1^d \in R^{n \times 1}$, $\tilde{x}_1 \in R^{n \times 1}$ is the tracking error and $x_1^d \in R^{n \times 1}$ is the desired position, $K_p, K_d \in R^{n \times n}$ are positive definite, symmetric and constant matrices, \hat{G} is an approximated value of G . We consider that the approximation error $\tilde{G} = \hat{G} - G$ is bounded as:

$$\|\tilde{G}\| \leq \bar{G} \tag{8}$$

Now we discuss the convergence of the closed-loop system.

Theorem 1. *The close-loop system with sliding mode control (7) for the general robot model (1) is asymptotic stable and the velocity parameter x_2 will converge to:*

$$\limsup_{T \rightarrow \infty} \|x_2\|^2 = 0 \tag{9}$$

Where T is the final time and $\bar{G} < K$.

Proof. The proposed Lyapunov function is:

$$V_1 = \frac{1}{2} x_2^T M x_2 + \frac{1}{2} \tilde{x}_1^T K_p \tilde{x}_1 \tag{10}$$

Substituting (7) into (3) we have the closed-loop system as:

$$M \dot{x}_2 = -K_p \tilde{x}_1 - K_d x_2 + \tilde{G} - K \operatorname{sign}(x_2) - C x_2 \tag{11}$$

The derivative of (10) is:

$$\dot{V}_1 = x_2^T M \dot{x}_2 + \frac{1}{2} x_2^T \dot{M} x_2 + x_2^T K_p \tilde{x}_1 \tag{12}$$

Where $\dot{\tilde{x}}_1 = \dot{x}_1 - \dot{x}_1^d = x_2 - x_2^d = x_2$. Substituting (11) into (12) gives:

$$\dot{V}_1 = -x_2^T K_d x_2 + x_2^T [\tilde{G} - K \operatorname{sign}(x_2)] + \frac{1}{2} x_2^T [\dot{M} - 2C] x_2$$

Using (4), (8) and that $|x_2|^T = x_2^T \operatorname{sign}(x_2)$ we have:

$$\dot{V}_1 \leq -x_2^T K_d x_2 + |x_2|^T \bar{G} - |x_2|^T K$$

$$\dot{V}_1 \leq -x_2^T K_d x_2 \tag{13}$$

Where $\bar{G} < K$. Integrating (13) from 0 to T yields:

$$\int_0^T x_2^T K_d x_2 dt \leq V_{1,0} - V_{1,T} \leq V_{1,0}$$

$$\limsup_{T \rightarrow \infty} \frac{1}{T} \int_0^T x_2^T K_d x_2 dt \leq V_{1,0} \limsup_{T \rightarrow \infty} \frac{1}{T} = 0$$

If $T \rightarrow \infty$ then $x_2 \rightarrow 0$, it is (9).

4 Simulation Results

Consider that the Two-Link Planar Elbow Arm of (5) and (9) is written as (11) and is detailed as:

$$M(q) = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \tag{14}$$

$$C(q, \dot{q}) = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}, G(q) = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}$$

Where $m_{11} = (m_1 + m_2) a_1^2 + m_2 a_2^2 + 2m_2 a_1 a_2 \cos(\theta_2)$, $m_{22} = m_2 a_2^2$, $m_{12} = m_{21} = m_2 a_2^2 + m_2 a_1 a_2 \cos(\theta_2)$, $c_{12} = -m_2 a_1 a_2 \sin(\theta_2) \dot{\theta}_2$, $c_{21} = m_2 a_1 a_2 \sin(\theta_2) \dot{\theta}_1$,

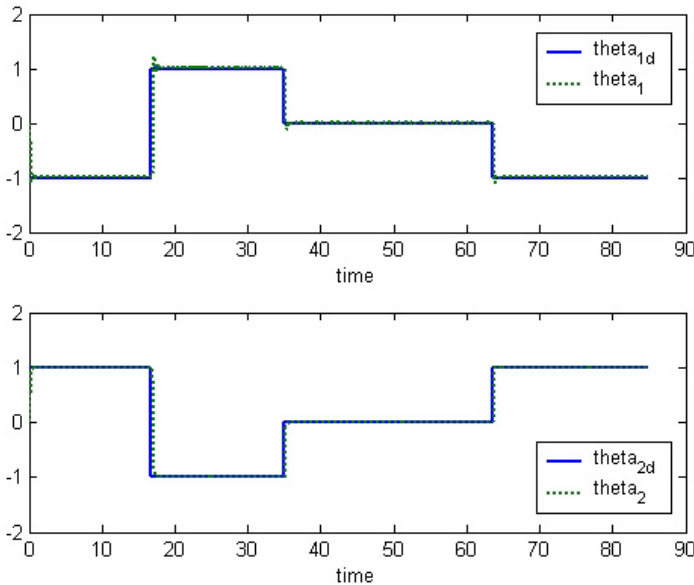


Fig. 1. Simulation behavior of the sliding mode control

$c_{11} = -2m_2a_1a_2 \sin(\theta_2) \dot{\theta}_2$, $c_{22} = 0$, $g_1 = (m_1 + m_2)ga_1 \cos(\theta_1) + m_2ga_2 \cos(\theta_1 + \theta_2)$, $g_2 = m_2ga_2 \cos(\theta_1 + \theta_2)$. m_1 and m_2 are the masses of link one, two, respectively, in kilograms, a_1 and a_2 are the length of link one and two, respectively, in meters, θ_1 and θ_2 are the angles in the joint one and two, respectively, in radians, g is the constant acceleration due to gravity in meter per second. We have the simulation in MATLAB considering $m_1 = m_2 = 1Kg$, $a_2 = a_3 = 1m$ and $g = 9.8m/sec^2$.

We select the parameters of the controller as $K_p = \text{diag}(k_{p1}, k_{p2})$, $K_d = \text{diag}(k_{d1}, k_{d2})$ where $k_{p1} = k_{p2} = 625$ and $k_{d1} = k_{d2} = 50$, $K = [k_1, k_2]^T$ where $k_1 = 5$ and $k_2 = 5$, $\hat{G} = [g_1, g_2]^T$ where $g_1 = 30$ and $g_2 = 10$. The behavior of the controller proposed is shown in Fig. 11.

5 Conclusion

In this paper it was proposed an sliding mode control for an elbow arm. It was proven that the closed loop system of the sliding model control and the model of the elbow arm is asymptotic stable. As a future research, will be proposed an high gain observer for the stimation of the velocity, the tracking control will be considered and another faster compensator than the neural networks will be considered.

Acknowledgment

The authors are thankful with the editor to invite them to be part of the committee.

References

1. Arimoto, S.: Fundamental problems of robot control: Part I, innovations in the realm of robot servo-loops. *Robotica* 13, 19–27 (1995)
2. Kelly, R.: Global Position of Robot Manipulators via PD control plus a Class of Nonlinear Integral Actions. *IEEE Transactions on Automat. Control* 43(7), 934–938 (1998)
3. Kelly, R.: A Tuning Procedure for Stable PID Control of Robot Manipulators. *Robotica* 13, 141–148 (1995)
4. Santibanez, Kelly, R.: Global Asymptotic Stability of the PD Control with Computed Feedforward in Closed Loop with Robot Manipulators. In: *Proc. 14th IFAC World Congress, Beijing*, pp. 197–203 (1999)
5. Lewis, F.L., Abdallah, C.T., Dawson, D.M.: *Control of robot manipulators*, NY 10022 (1993)
6. Kim, Y.H., Lewis, F.L.: Neural Network Output Feedback Control of Robot Manipulator. *IEEE Transactions on Neural Networks* 15, 301–309 (1999)
7. Lewis, F.L., Parisini, T.: Neural Network Feedback Control with Guaranteed Stability. *Int. J. Control* 70(3), 337–339 (1998)
8. Canudas de Wit, C., Slotine, J.E.: Sliding Observers for Robot Manipulator. *Automatica* 27(5), 859–864 (1991)

9. Spong, M.W., Vidyasagar, M.: Robot Dynamics and Control. John Wiley & Sons, Chichester (1989)
10. Ortega, R., Spong, M.W.: Adaptive Motion Control of Rigid Robot: A Tutorial. *Automatica* 25(6), 877–888 (1999)
11. Takegaki, M., Arimoto, S.: A New Feedback Method for Dynamic Control of Manipulator. *ASME J. Dynamic Syst. Measurement* 103, 119–125 (1981)
12. Tomei, P.: Adaptive PD Controller for Robot Manipulator. *IEEE Transactions on Automatic Control* 36, 556–570 (1992)
13. Yazarel, H., Cheah, C.C., Liaw, H.C.: Adaptive SP-D Control of Robotic Manipulators in the Presence of Modeling Error in Gravity Regressor Matrix: Theory and Experiment. *IEEE Transactions on Robotics and Automation* 18(3), 373–379 (2002)
14. Yu, W., Poznyak, A.S., Sanchez, E.N.: Neural Adaptive Control Two-Link Manipulator with Sliding Mode. In: *IEEE International Conference on Robotics and Automation*, Detroit USA, vol. 4, pp. 3122–3127 (1999)
15. Yu, W., Li, X.: PD Control of Robot with Velocity Estimation and Uncertainties Compensation. *International Journal of Robotic and Automation* 21(1), 1–9 (2006)

Stabilization on a Physical Pendulum with Moving Mass

Oscar Octavio Gutiérrez-Frías¹, Juan Carlos Martínez-García²,
and Ruben Garrido-Moctezuma²

¹ Centro de Investigación en Computación del IPN,
México, D.F., México
oscargf@sagitario.cic.ipn.mx

² Departamento de Control Automático, Cinvestav-IPN,
México, D.F., México
{garrido,martinez}@ctrl.cinvestav.mx

Abstract. An asymptotic PD controller with gravity compensation to attenuate the oscillation of a poorly damping physical pendulum system is presented. The active vibration-damping element is a mass that moves along the pendulum arm. The stability analysis was carried out using the traditionally Lyapunov method in conjunction with LaSalle's theorem. The closed-loop asymptotic stability performance was tested with some numerical simulations.

Keywords: Vibration control, PD controller, Lyapunov approach, under-actuated system.

1 Introduction

Vibrating mechanical systems are an important class of dynamical systems including buildings, bridges, car suspensions, pacemakers, wind generators, and hi-fi speakers [6]. In physical terms, all vibrating systems consist of interplay between an energy-storing component and an energy-carrying component. Due to theoretical and technological reasons, the control of vibrating mechanical systems is an important domain of research, which has provided technological solutions to several problems concerning oscillatory behaviors of some important classes of dynamical systems. For instance [1,3,4,5,8,10,14,17] and the references therein.

As far as mathematical tools are concerned, the control of vibrations has mainly been tackled via frequency-domain techniques, which are essentially restricted to linear systems (see for instance [9,21]). If the vibrating systems are nonlinear and if they oscillate too far away from their equilibrium points, frequency-domain techniques are not suitable. Hence, modern approaches employ time-domain nonlinear control strategies.

This paper focuses on active control of under-damped lumped nonlinear under-actuated vibrating mechanical systems following an energy-based approach, i.e. the control of vibrations is tackled via the shaping of the energy flow that characterizes the system. Exist many works related with this problem but a detailed

review of the state of the art of the problem is beyond the scope of this work. However, we refer the interested reader to the following references: [1,2,3,4,5,15] and [11,22].

This paper presents the stabilization of a frictionless under-actuated physical pendulum with a radially moving mass. This nonlinear dynamical system was originally proposed in [18], where several control strategies solved the aforementioned stabilization problem. The proposed stabilizing strategies include a modified nonlinear Proportional Derivative (PD) controller and a neural network approach. Further works also studied this physical system [19,20]. It is interesting to point out that these approaches are mainly heuristic and they do not provided rigorous stability proofs. The proposed controller is composed of two parts, a linear PD controller corresponds to the first part and the second part is a constant term that is equal to the gravity force term associated to the moving mass.

This contribution is organized as follows: Section 2 presents the model of the physical pendulum with moving mass as well as its main physical properties. Section 3 describes the proposed control law and the stability analysis of the closed loop system. Section 4 depicts some computer simulations. The paper ends with some final comments.

2 Physical Pendulum with Moving Mass

2.1 Lagrangian Modeling

Consider a mechanical system consisting of a physical frictionless pendulum of mass M with its pivot at O and an auxiliary mass m able to slide to and from the pivot as depicted in Figure 1.

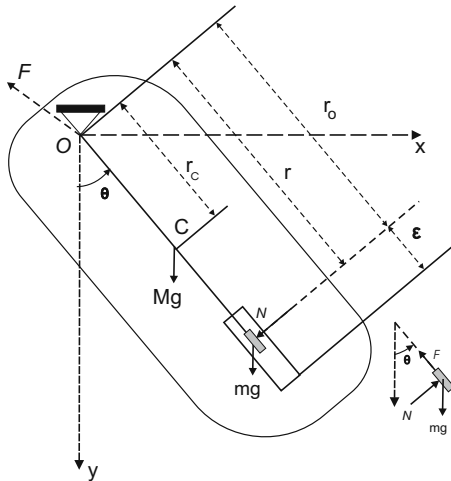


Fig. 1. Physical pendulum with moving mass

The moment of inertia of the pendulum about the pivot is given by I_0 , and its center of mass C is located at distance r_c from the pivot. The forces acting on the mass m are the gravitational force mg and a force F parallel to the guide \overline{OC} and supplied by an actuator, i.e. an electric motor, attached to the auxiliary mass. The pivot O is the origin of the reference frame $x - y$. The x -axis is set in the horizontal direction and the y -axis is set in the vertical direction. The set of generalized coordinates are the angle θ between \overline{OC} and the y -axis, and the radial displacement r of the mass m from the pivot O . It is easy to show that for this system total kinetic energy K_c and the total potential energy K_p are given by

$$K_c = \frac{1}{2}I_0\dot{\theta}^2 + \frac{1}{2}mr^2 + \frac{1}{2}mr^2\dot{\theta}^2 \quad (1)$$

and

$$K_p = Mgr_c(1 - \cos \theta) - mgr \cos \theta, \quad (2)$$

respectively. Note that the total kinetic energy comprises the rotational energy of the pendulum as well as the translational and rotational energy of the sliding mass. The above equations allows writing the Lagrangian function $L(q, \dot{q})$

$$L(q, \dot{q}) = K_c - K_p,$$

where $q = [r, \theta]^T$. From the above, the corresponding Euler-Lagrange equations are given by

$$\begin{cases} m\ddot{r} - mr\dot{\theta}^2 - mg \cos \theta = F, \\ (mr^2 + I_0)\ddot{\theta} + 2mr\dot{r}\dot{\theta} + g(Mr_c + mr) \sin \theta = 0, \end{cases} \quad (3)$$

Note that system (3) and the under-actuated system proposed in [1] and referred as the BBC system have a similar structure. However, in our system the main rotary pendulum is not perfectly balanced around the rotation axes, as it does in the BBC. On the other hand, the system proposed by [1] is locally controllable at the origin, while the system (3) is not, as we can see more latter.

2.2 Model Properties

The mechanical system (3) has several interesting properties, which facilitate the design of the proposed control law. Consider that the following expression describes the force F

$$F = -mg + v. \quad (4)$$

and apply it to (3). The resulting system is equivalent to the following Euler-Lagrange system

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + \nabla_q K_i(q) = Gv, \quad (5)$$

where $G = [1, 0]^T$ and $K_i(q) = Mgr_c(1 - \cos \theta) + mgr(1 - \cos \theta)$ and

$$M(q) = \begin{bmatrix} m & 0 \\ 0 & mr^2 + I_0 \end{bmatrix} \quad \text{and} \quad C(q, \dot{q}) = \begin{bmatrix} 0 & -mr\dot{\theta} \\ mr\dot{\theta} & mr\dot{r} \end{bmatrix},$$

respectively, and v is the new input.

System (5) satisfies the following properties:

P1) $M(q)$ is positive definite.

P2) $H := \dot{M}(q) - 2C(q, \dot{q})$ is skew symmetric with

$$H = \begin{bmatrix} 0 & -mr\dot{\theta} \\ mr\dot{\theta} & 0 \end{bmatrix},$$

P3) The operator $v \rightarrow \dot{r}$ is passive. For this, one verifies, using properties **P1** and **P2** it follows that the time derivative of the total energy function,

$$E(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} + K_i(q), \quad (6)$$

is given by $\dot{E} = v\dot{r}$. That is, the system with v as input and \dot{r} as output is passive (see [12]).

Finally, we finish this section by stating the following important remark.

Remark 1. When $v = 0$ and $-\pi/2 < \theta < \pi/2$, the system (5) has one of equilibrium point defined by $\bar{x} = (r = \bar{r}, \theta = 0, \dot{r} = 0, \dot{\theta} = 0)$, where \bar{r} is a positive constant. These point is stable but not asymptotically stable, that is, the system (5) can oscillate around $\theta = 0$ Moreover, linearization of system (5) around the lower stable equilibrium point \bar{x} , produces

$$\begin{aligned} m\ddot{r} &= 0, \\ (m\bar{r}^2 + I_0)\ddot{\theta} + g(Mr_c + m\bar{r})\theta &= 0, \end{aligned} \quad (7)$$

which is evidently not locally controllable.

3 The Control Law

Before to establish the control objective of this work, we define the admissible set $Q \subset R^2$ as

$$Q = \{q = (r, \theta) : 0 < r - \varepsilon < r < \bar{r} + \varepsilon \wedge |\theta| < \bar{\theta} < \pi\}$$

where ε , $\bar{\theta}$ and \bar{r} are already given. Inside this set is where we want to restrict the movement of the position variable q of the pendulum system (5). Physically, this means that the mass displacement will be restricted to move inside of a perfectly defined region. Additionally, we also excluded all the pendulum trajectory that pass through the unstable equilibrium points defined as $(r = \bar{r}, \theta = n\pi, \dot{r} = 0, \dot{\theta} = 0)$; with $n = \{1, 2, \dots\}$.

Now the main control problem statement is presented:

Problem 1. Consider the physical pendulum with a moving mass described in (5), under the assumption that the position variable q is initialized inside of $Q - \bar{q}$, where $\bar{q} = (\bar{r}, 0)$. Then, the control objective is to asymptotically bring the rotating pendulum with moving mass to the lower equilibrium point $\bar{x} = (\bar{q}, 0)$, keeping the two position variables $q = (r, \theta)$ to be confined inside of the previously defined set Q .

To solve the above mentioned control problem, we introduce the following Lyapunov function candidate

$$E_T(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} + K_m(q), \tag{8}$$

where $K_m(q)$ is the modified potential energy stated as

$$K_m(q) = \frac{k_p}{2} (r - \bar{r})^2 + K_i(q), \tag{9}$$

with $k_p > 0$.

Taking into account the passivity properties of model (5), the first time derivative of E_T along the trajectories of the system is given by

$$\dot{E}_T(q, \dot{q}) = v \dot{r} + k_p (r - \bar{r}) \dot{r}. \tag{10}$$

Since this derivative needs to be definite negative, then the following control law is proposed

$$v = -k_p (r - \bar{r}) - k_d \dot{r}, \tag{11}$$

which ensures that

$$\dot{E}_T(q, \dot{q}) = -k_d \dot{r}^2, \tag{12}$$

with $k_d > 0$.

From the above we have that function \dot{E}_T is negative semi-definite. Thus, only the stability of the equilibrium point \bar{x} , in the Lyapunov sense, can be established. The above means that q and \dot{q} are bounded signals, in a small vicinity of the equilibrium point \bar{x} . To ensure that q and \dot{q} asymptotically converge to zero, it is necessary to apply the LaSalle’s invariance theorem (see for instance [12]).

Now, in order to show that $x = (q, \dot{q})$ converges asymptotically at the origin \bar{x} , with the vector position q belongs to Q ; for all time. We needed to use the convex property of the modified potential energy $K_m(q)$, which is discussed in the following remark.

Remark 2. The selected potential energy $K_m(q)$ has a minimum at $\bar{q} = (\bar{r}, \theta)$ since

$$K_m(\bar{q}) = 0, \nabla_q K_m(q)|_{q=\bar{q}} = 0 \text{ and } \nabla_q^2 K_m(q)|_{q=\bar{q}} > 0.$$

As a matter of fact, the above condition implies that $K_m(q)$ is a convex function around \bar{q} . In geometrical terms, the level curves of $K_m(q)$ are constituted by a set of closed-loop curves around \bar{q} . That is, the set E_c , defined by $E_c = \{q \in R^2 : K_m(q) \leq c\}$ is a strictly convex set, for $c \in (0, \bar{c})$. Where the values of \bar{c} can be always numerically estimated and it depends on the values parameters \bar{r} , ε , $\bar{\theta}$ and k_p (Figure 4 depicts the level curves of the modified potential energy, for two values of k_p).

Before to probe the asymptotic convergence to the origin, keeping q inside of Q , we give some conditions related to the stability domain of the closed-loop system. Since $E_T > 0$ is a non-increasing function (12), we have

$$K_m(q(t)) \leq \frac{1}{2} \dot{q}^T M(q(t)) \dot{q}(t) + K_m(q(t)) = E_T(q(t), \dot{q}(t)) \leq E_T(q(0), \dot{q}(0)) < \bar{C}. \tag{13}$$

So, selecting \bar{C} as

$$\bar{C} = \max\{c > 0 : E_c \subset Q\},$$

we have that for any initial condition $(q(0), \dot{q}(0))$; with $q(0) \in Q$ and $E_T(x(0)) < \bar{C}$ implies that that $q(t) \in Q$; for all $t > 0$.

Remark 3. Notice that the inequality (13) is an invariant set for the closed-loop system defined by (5) and (11). Due to the fact that for all the initial conditions $x(0) = (q(0), \dot{q}(0))$; with $q(0) \in Q$, provided that $E_T(x(0)) < \bar{C}$, it follows that $q(t) \in Q$; for all $t > 0$ with $\dot{q}(t)$ bounded. According to this fact, we can define a compact invariant set Ω , as follows

$$\Omega = \{x = (q, \dot{q}) : E_T(q, \dot{q}) \leq \bar{C}\}. \tag{14}$$

That is, any initial condition starting at Ω remains forever in Ω ; with q belonging to Q . This set allows us to apply the LaSalle’s invariance Theorem (12).

Recapig, from Remark 3, we have the closed-loop solution x of (5) and (11) is bounded for any initial condition such that $x(0) \in \Omega$. To end the stability proof it is necessary to invoke the invariance LaSalle’s Theorem.

Let us define the invariant set S as follows

$$S = \{(q, \dot{q}) \in \Omega : \dot{E}_T(q, \dot{q}) = 0\} = \{(q, \dot{q}) : \dot{r} = 0\}. \tag{15}$$

Clearly, in the set S , we have that $\ddot{r} = \dot{r} = 0$ and $r = \underline{r}$, where \underline{r} is a constant. Thus, substituting these quantities into (5) it follows that

$$\begin{cases} -m\underline{r}\dot{\theta}^2 - mg(\cos \theta - 1) + k_p(\underline{r} - \bar{r}) = 0 \\ (m\underline{r}^2 + I_0)\ddot{\theta} + g(Mr_c + m\underline{r}) \sin \theta = 0 \end{cases} \tag{16}$$

The time derivative of the first equation in (16) yields

$$(2\underline{r}\ddot{\theta} - g \sin \theta) \dot{\theta} = 0 \tag{17}$$

Two cases must then be analyzed

Case (a). If $\dot{\theta} = 0$ in the set S , it also follows that $\ddot{\theta} = 0$. From the second differential equation of (16) it is clear that $\sin \theta = 0$ since $Mr_c + m\underline{r}$ is strictly positive. That is, in the set S , it follows that $\theta = 0$. And from the first equation of (16) $\underline{r} - \bar{r} = 0$. Hence, $r = \bar{r}$ in the set S .

Case (b). If $\dot{\theta} \neq 0$ in the set S , then (17) implies that:

$$\ddot{\theta} = g \sin \theta / 2\underline{r}. \tag{18}$$

Since $q \in \Omega$, then $\underline{r} > 0$. Thus, $\ddot{\theta}$ is well defined.

Taking into account (16) and (18) yields to the following algebraic equation for the variable θ

$$0 = (2Mr_c\underline{r} + 3m\underline{r}^2 + I_0) \sin \theta,$$

which implies that $\sin \theta = 0$ on the set S because $2Mr_c\underline{r} + 3m\underline{r}^2 + I_0 > 0$. This mean that case (b) is not possible since it is assumed that $\dot{\theta} \neq 0$. Therefore $\dot{\theta} = 0$ and case (a) is the only possibility.

The above analysis allows concluding that the largest invariant set contained in S is given by \bar{x} . According to the La Salle theorem, all the trajectories starting in Ω asymptotically converge towards $(\bar{r}, 0, 0, 0)$. The following proposition resumes the stability result previously presented.

Proposition 1. *Consider the closed-loop dynamical model given by (3) in closed-loop with the following control law:*

$$F = -mg - k_p(r - \bar{r}) - k_d\dot{r},$$

with \bar{r} , k_p and k_d positive constants. Then, under the assumption of Remark 2, we have that all the trajectories starting in Ω (14) asymptotically converge towards the lower equilibrium point $\bar{x}=(\bar{r}, 0, 0, 0)$. □

We omit a proof of the main results because we do not have enough space.

4 Numerical Simulations

To illustrate the performance of the proposed control law, a numerical simulation was carried out in the MATLABTM program. The system physical parameters were set as follows

$$m = 1[Kg] \quad M = 2.5[Kg] \quad r_c = 0.7[m] \quad I_0 = 1.22[Kg.m]^2$$

The radial displacement of m was given by $\bar{r} = 2.5[m]$ and $\varepsilon = 0.75[m]$ and the initial conditions were chosen to be $\theta(0) = 0.5[rad]$, $r(0) = 2.5[m]$, $\dot{\theta}(0) = 0[rad/s]$ and $\dot{r}(0) = 0[m/s]$, the restriction on the angular position was taken as $\bar{\theta} = \pi/2$. The control gains, empirically proposed to increase the convergence rate of the closed-loop system, were set as $k_p = 19.6$ and $k_d = 1.9$. For this particular case the largest \bar{C} was equal to 5.5. Notice that the initial condition lies inside the stability domain of attraction, because: $E_T(0) = 5.09$.

Figures 2, 3 and 4 show the closed-loop performance of the nonlinear system with the specified control law. Particularly, Figure 3 depicts the behavior of the force F and the energy function E_T , while Figure 4 shows the level curves associated with the modified potential energy $K_m(q)$, for the two different values of the parameter $k_p = 19.6$ and $k_p = 40$.

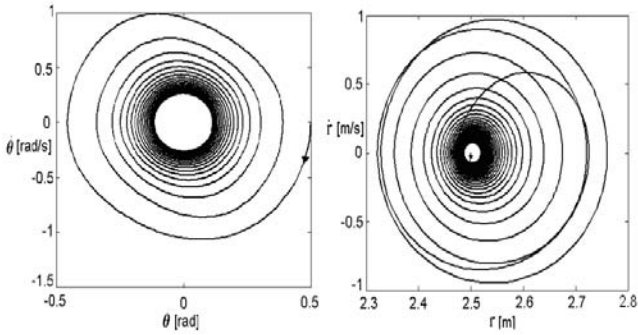


Fig. 2. Feedback controlled performance of the original system to the proposed strategy

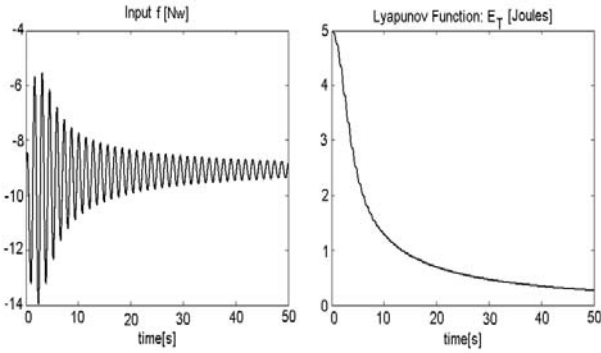


Fig. 3. Closed-loop behavior of the proposed input force and the Lyapunov function, respectively

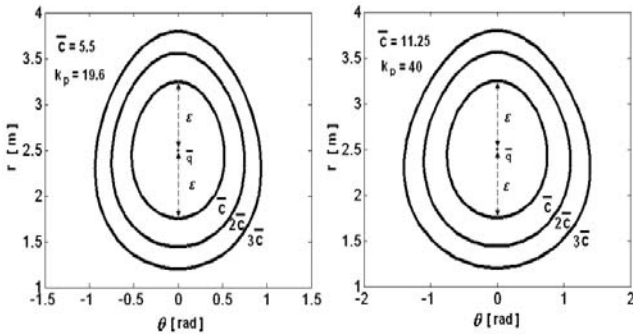


Fig. 4. Level curves of $K_m(q)$ around the origin for $k_p = 19.6$ (left) and $k_p = 40$ (right)

4.1 Simulation Analysis

As can be seen in Figure 2, the radial displacement is always bounded. In fact, the radial moving mass motion is bounded by [2.3,2.8]. This motion is, indeed, more restrictive than the one characterized by the fixed ε . It must be pointed out that the control strategy is effective in stabilizing the system with reasonable control input effort for a large deviation from the equilibrium point. However, the damping injection capabilities of the proposed strategy is somewhat limited. That is, the system is brought to the desired equilibrium very slowly. As can be noted in Figure 4, the region of attraction of the specified equilibrium point can be increased by just augmenting the value of k_p . However, it is not convenient to consider high values for the proportional gain k_p , because it would give rise to high frequencies of the closed-loop oscillations. Thus, it is better to consider small values for parameter k_p in order to guarantee that $|r - \bar{r}| < \varepsilon$.

5 Conclusions

This paper proposes a **PD** control law plus gravity compensation for active vibration damping in a frictionless physical pendulum with moving mass. The control law is able to damp out the oscillations of the pendulum by using as active damper the moving mass. The control law design exploits the underlying physical properties of the physical pendulum with moving mass to shape a convenient Lyapunov function candidate. The asymptotic stability analysis was carried out using the LaSalle's Theorem. Moreover, the control law only needs measurements of the position and velocity of the moving mass. Compared with previous approaches [18,20], the proposed methodology does not need to synchronize the motion of the moving mass with the swings of the pendulum implying simultaneous measurement of the pendulum and moving mass positions. The proposed strategy is a first step towards the reduction of undesirable oscillation in civil structures.

Acknowledgments. This research was supported by the Centro de Investigación en Computación of the Instituto Politécnico Nacional, by the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional (SIP-IPN), under Research Grant 20082694, by the Centro de Investigación y Estudios Avanzados del Instituto Politecnico Nacional (Cinvestav-IPN) by CONACYT-México, under Research Grant 32681-A. Octavio Gutiérrez is a scholarship holder of the Consejo Nacional de Ciencia y Tecnología (CONACYT-México).

References

1. Aguilar-Ibañez, C.F., Guzmán-Aguilar, F., Lozano, R., Chimal-E, J.C.: A control energy approach for the stabilization of the rigid beam balanced by the cart. *Int. Jour. of Robust and Nonlinear Control* (2008), doi:10.1002/rnc.1371
2. Aguilar-Ibañez, C., Sossa-Azuela, H.: Stabilization of the Furuta Pendulum Based on a Lyapunov Function. *Nonlinear Dynamincs* 49(1-2), 1–8 (2007)

3. Aguilar-Ibañez, C., Lozano, R.: The ball and beam acting on the ball, ch. 10. In: Isidori, A., Van Schuppen, J.H., Sontag, E.D., Thoma, M. (eds.) *Non-linear Control for Underactuated Mechanical Systems*, pp. 143–151. Springer, Heidelberg (2002)
4. Aguilar-Ibañez, C., Sira-Ramirez, H.: PD Control for active vibration damping in an underactuated nonlinear system. *Asian Journal of Control* 44(4), 502–508 (2002)
5. Aguilar-Ibañez, C., Sira-Ramirez, H.: A linear differential flatness approach to controlling the Furuta pendulum. *IMA Journal of Mathematical Control and Information Advance* 24(1), 31–45 (2007)
6. Behrens, S.: Potential system efficiencies for MEMS vibration energy harvesting. In: *Proceedings of SPIE-Volume 6414 Smart Structures, Devices, and Systems III*, vol. 6414, pp. 6414D (2007)
7. Butikov, E.I.: Parametric resonance. *Comput. Sci. Eng.* 1(3), 76–83 (1999)
8. Dong, P., Benaroya, H., Wei, T.: Integrating experiments into an energy-based reduced-order model for vortex-induced-vibrations of a cylinder mounted as an inverted pendulum. *Journal of Sound and Vibration* 276(1-2), 45–63 (2004)
9. Fuller, C., Elliot, S.J., Nelson, P.A.: *Active Control of Vibration*. Academic Press, San Diego (1996)
10. Inman, D.: *Engineering Vibration*. Prentice Hall, New York (1994)
11. Isidori, A.: *Nonlinear Control Systems*, 3rd edn. Springer, New York (1995)
12. Khalil, H.K.: *Non-linear Systems*, 2nd edn. Prentice-Hall, NJ (1996)
13. Lefschetz, S., La Salle, J.P.: *Stability By Liapunov's Direct Method with Applications*. Academic Press, New York (1961)
14. Matsuhisa, H., Gu, R., Wang, Y., Nishihara, O., Sato, S.: Vibration control of a ropeway carrier by passive dynamic vibration absorbers. *JSME International Journal (Series C)* 38(4), 657–662 (1995)
15. Ortega, R., Loria, A., Nicklasson, P.J., Sira-Ramirez, H.: *Passivity-based control of Euler–Lagrange systems*. Springer, Berlin (1998)
16. Sira-Ramírez, H., Agrawal, S.K.: *Differentially Flat Systems*. Marcel Dekker, New York (2004)
17. Sira-Ramírez, H., Llanes, O.: Sliding mode control of nonlinear mechanical vibrations. *Journal of Dynamics, Systems, Measurements and control* 122(4), 674–678 (2000)
18. Stilling, D.S.D.: *Vibration attenuation by mass redistribution*. Ph. D. Thesis, University of Saskatchewan, Saskatoon, Saskatchewan, Canada (2000)
19. Stilling, D.S.D., Szyszkowski, W.: Controlling angular oscillations through mass reconfigurations: a variable length pendulum case. *Int. J. Non-linear Mech.* 37(1), 89–99 (2002)
20. Szyszkowski, W., Stilling, D.S.D.: On damping properties of a frictionless physical pendulum with a moving mass. *Int. J. Non-linear Mech.* 40(5), 669–681 (2005)
21. Thomson, W.T.: *Theory of vibrations with Applications*. Allen and Unwin, London (1981)
22. Utkin, V.: *Sliding Modes Control in Electromechanical Systems*. Taylor & Francis, London (1999)

Restricted Growth String for Video-Type Classification

Pedro Luis Sánchez Orellana, José Torres Jiménez,
and Claudio Castellanos Sánchez

Laboratory of Information Technology of Centre for Research and Advanced Studies
LIT Cinvestav - Tamaulipas, Ciudad Victoria, Tamaulipas, México
{psanchez, jtj, castellanos}@tamps.cinvestav.mx

Abstract. The use of videos to extract information from the environment is one of the most challenging task in the computer vision area. The video sequences are of particular relevance for the recognition of humans in environments; in spite of the large databases existing in the literature, there is a lack of the uniformity among them which causes difficulties at the moment of making comparisons. The main reason is the restrictions on the conditions of the sequences (like illumination). To solve this problem we are proposing a set of 5 characteristics that highly influence the human's gait recognition, also we propose the use of the Restricted Grow Strings to cluster the data bases of characteristics that represent the conditions in the videos.

1 Introduction

In the artificial vision area one of the most important aspects is the information extraction from video sequences. This allows to extract and analyze information from different actions like human gait [1,2,3]. However, the main trouble is the way the databases are created and the differences between the existent databases, since most of them were created with the objective of testing a specific architecture. The lack of uniformity on the videos turns into a very difficult task to properly compare the architectures, since it is often necessary to adjust them to fit into the characteristics of the videos. Although, there has been some research work that actually focus on the classification of the videos by analysing its contents [4,5,6] and inspire of the fact that these works can be used to classify the type of scenes present in the videos we are proposing the use of characteristics that must be taking into account at the moment of recording the videos.

This work is the first part of a major work to create a database with the more general characteristics that can be taken to uniform the recording of videos in the area of computer vision. In this first work we focus on the methodology necessary to make an optimum classification of the databases of characteristics present in the videos.

The definition of the characteristics are focused on specific databases for human gait recognition problem. The proposed characteristics are: the illumination

conditions, the amount of objects in motion, whether there is a static or non-static environment and outdoor/indoor environment. This features can be clustered in order to analyze the complexity of a video, since certain configurations on them can increase the difficulty to achieve the recognition.

On this, several algorithms for clustering can be implemented, from the statistical point of view (k-means) to the neural networks approach (SOM); however, we are proposing the use of the Restricted-Growth Strings (RGS), since this algorithm offers advantages for our problem. The first one is that it does not need to know the amount of clusters to separate the main set. Another one is that it allows to know all the possible cauterizations of the training set, which by minimising the similarity measurements ratio between the inter-cluster and intra-cluster of the training set. The other one is that optimum convergence can be ensured by selecting the closest-to-centre element.

Through this paper we describe first the complexity of the video-type classification, then we describe some of the theory behind the RGS, our methodology and finally we will go trough some results, conclusions and future work.

2 Complexity of Video-Type Clustering

The main difficulty relies on the capacity of a given algorithm for classification to separate a set of features (that represents the description of the conditions of a given video). Although knowing the optimum disposition of the videos in a cluster is an exhaustive task that involves the extraction of all the possible configurations for grouping the videos it ensures the best classification possible. This task can be overcome by using clustering algorithms, which are a very important research field on Pattern Recognition area due to its contribution in:

- Data reduction. Many times, the amount of the available data, N , is very large and, as consequence, its processing becomes very demanding.
- Hypothesis generation. In this case we apply cluster analysis to a data set in order to infer some hypotheses concerning the nature of the data.
- Hypothesis testing. In this context, cluster analysis is used for the verification of the validity of a specific hypothesis.
- Prediction based on groups. In this case we apply cluster analysis to the available data set, and the resulting clusters are characterized based on the characteristics of the patterns by which they are formed.

For this purpose many algorithms were created like, Self Organizing Maps [78], K-means [9,10], Learning Vector Quantization [11,12], Hierarchical Clustering [13,14], all of them posses the capability to extract information from very large datasets. However, the main disadvantage is that they do not offer the capability to explore all the possible search space. In this way we are proposing the use of the RG strings, this algorithm offers not only the power to cluster the information but also, by using adequate Branch & Bound strategies we can define an optimization technique that will allows the improvement on the construction time. At the same time, keeping the advantage of full exploration of the space.

In the following section we describe our methodology for using the clustering with RGS of video using a data set of video features.

3 Proposed Methodology

Our methodology consists of the generation of a set of all possible types of cluster configurations of the video sequences set as it can be seen in the figure 1. First of all, we consider video-features data set which contains a feature vector of 5 characteristics that influence at evaluating the gait of a person. The characteristics that we are proposing were established in the outdoor environments and there are, the illumination conditions (day/night), objects in motion through the sequence, background motion, frames per second in the sequence and camera motion. The measurements were done manually.

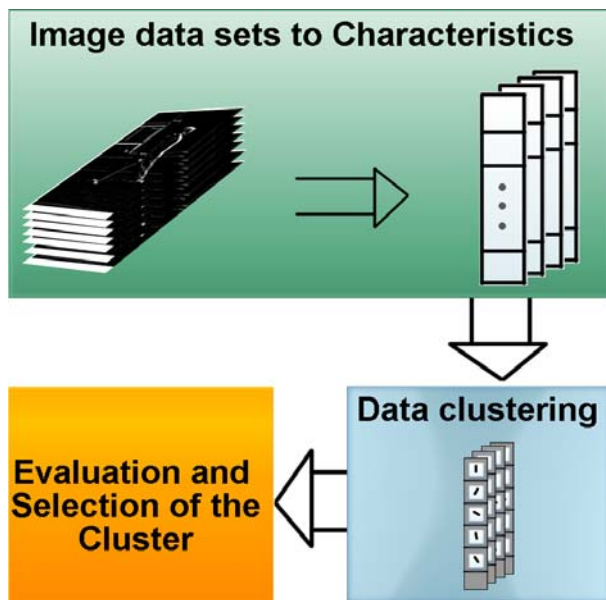


Fig. 1. Proposed architecture. The characteristics of the sequences are extracted manually and then clustered by RGS and measuring the quality of each one formed.

The characteristics were introduced into a RGS algorithm to cluster them, as it is shown in the following subsection.

3.1 RGS

The RGS allows to make a change in the space so the feature vector that represents the conditions in the videos is transformed into a RGS space. This means that we can find the ways to put x items into k groups of n elements, where

the ordering within the groups does not matter nor the ordering of the groups themselves. In other words, a type of set partition. The main advantage compared to other set partition iterators is that they need to iterate over 10 billion partitions just to get the 2.6 million ones in the case of a subset of 4 elements.

The definition of a RGS is the following: for a set partition of n elements, the n character string a_1, a_2, \dots, a_n in which each character gives the set block (B_0, B_1, \dots) in which the corresponding element belongs is called the restricted growth string (or sometimes the restricted growth function). For example, for the set partition 1, 2, 3, 4, the restricted growth string would be 0122. If the set blocks are "sorted" so that $a_1 = 0$, then the restricted growth string satisfies the inequality:

$$a_{(i+1)} \leq 1 + \max(a_1, a_2, \dots, a_i) \tag{1}$$

for $i=1, 2, \dots, n-1$.

Let us consider the following sample: The periods separate the individual sets so that, for example, 1.23.4 is a partition from 1,2,3,4.

Table 1. Possible combinations for 4 elements

Partitions	Blocks produced
1	1234
2	123.4, 124.3, 134.2, 1.234, 12.34, 13.24, 14.23
3	1.2.34, 1.24.3, 1.23.4, 14.2.3, 13.2.4, 12.3.4
4	1.2.3.4

In the following subsection we describe the clustering measures we used to consider the quality of each and every RGS generated.

3.2 Data Clustering

For the clustering formation we used the RGS, this algorithm involves the creation of a tree structure which contains all the possible configurations of the clustering, see figure 2 for an example of a RGS tree of 4 elements. As it can be seen the main disadvantage is the creation and so evaluation of the tree, since its growth is exponential, for a tree of the possible combination grows up to 120.

So we decided to optimize the RGS growing by adding a function to minimize both the distance inside the cluster (inter-cluster) and the distance between the clusters generated (intra-cluster). The function is defined as:

$$\vartheta = \min_{i=1}^n (S_i) \tag{2}$$

where S is the similarity function 6, and n are the total of vectors to evaluate.

The first step of our iterative version of the RGS for video-features clustering consist on determining the centroid of the training data by:

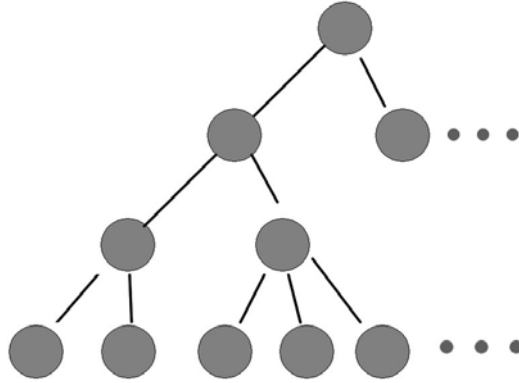


Fig. 2. RGS tree sample formed with a string of 4 elements

$$C = \frac{1}{n} \left(\sum_{i=1}^n I_{S_i} \right) \tag{3}$$

where I_{S_i} is the i -esim vector of features of the i -esim video of the training set.

So we re-sort the training data in order to obtain the closest element of the training set to the C . This causes the reconstruction to become more focused on the centre of the whole set. For the following elements of the training set we have to determine the belongness of the according to the convenience in terms of the closeness to the previous grouped set. So we still have to minimize the ϑ variable. This rule still full fills the requirements of the RGS construction since the options for the following element to group are the maximum of the previous elements already grouped. This process is repeated until we group the total of the videos.

The objective is to create a construction of the RGS optimized not only based on the aptitude function but also tanking into account the closeness of the data with respect to the centroid, and so generating a more compact clusters structure.

However, to measure the closeness of and I_{S_i} element we are proposing to use the quality measures described in the following.

It is natural to ask what kind of standards we should use to determine the similarity between clusters and the quality of them, or how to measure the opposite. Usually, a prototype is used to represent a cluster so that it can be further processed like other objects. Here, we focus on reviewing measure approaches between individuals due to the previous consideration [15][16].

Mainly we look for the dissimilarity intra-cluster (similarity between objects of the same cluster) and the inter-cluster (similarity between different clusters). These measurements are the equivalent to measure the dissimilarity of the grouped objects and sets.

For the intra-cluster measure we use a measure on the dispersion between the elements of a C_i cluster defined as:

$$S_i = \left(\frac{1}{n_i} \sum_{x \in C_i} \|x - w_i\|^r \right)^{\frac{1}{r}} \quad (4)$$

where n_i is the number of the vectors in C_i , w is the weighted mean vector and r is the ratio.

Also to measure the dissimilarity among the clusters we use:

$$d = \left| w_1 - \sum_{k=2}^l w_k \right|^q \quad (5)$$

where w is the mean vectors of existing cluster respectively, the combination of this measure to obtain the dissimilarity value between the two cluster is given by the following formula:

$$M = \frac{\sum_{i=1}^l S_i}{d} \quad (6)$$

where S is the measures of dispersion between the elements of a cluster, d is the intra-cluster dissimilarity measurement and n is the total amount of clusters. These measures can be used to help binding the construction of the RGS tree, by considering the dispersion of each element while it is added to RGS tree. In the following section we show some results obtained and a comparison between two other clustering techniques.

4 Results

In order to compare the clustering capacities of our approach we used a 2 data bases each one of 120 videos with the 5 features extracted from each one. We compared our algorithm against two of the most common clustering techniques, the K-means and the Self Organizing Maps (SOM) trained 1000 epochs. For the test, we ranged the amount of clusters from 3 to 40. Some of these results are shown in the table [2](#). All the clusters produced by the three techniques were measured with the same equations described in our methodology. One important thing to remark is what happens while increasing the amount of clusters, as it can be seen in the case of the k-means, the algorithm decrease its capability to keep the classes as separated as possible. The consequence is that the ratio between distance inter-cluster and intra-cluster. This phenomena does not occur for both the SOM and the RGS. In the case of the K-means, it constructs the clusters considering the a priori designation of the amount of clusters which force the algorithm to group the data into all of them.

Although the percentage of correct classification of the SOM is superior to the RGS, it is important to consider that since the cluster configuration for this algorithm is not always the same. So forth achieving the optimum is not a matter of incrementing the amount of epochs, but it depends on the initial weights. The results we show are the average of several training repetitions, which make the process more expensive than the RGS algorithm.

Table 2. Results of the classification

Partitions generated	Algorithm	M_i
5	K-means	0.483
	SOM	0.383
	RGS	0.402
10	K-means	0.527
	SOM	0.336
	RGS	0.328
25	K-means	0.547
	SOM	0.345
	RGS	0.474
40	K-means	0.872
	SOM	0.456
	RGS	0.540

5 Conclusion and Future Work

In this paper we have proposed an algorithm of complexity $\frac{N(N-1)}{2}$ to solve the problem of video-feature classification, where N is the amount of elements. Also this paper is the first part of a major work for selection the optimum test set for our problem of gait recognition. Since there is no set of sequences that allow testing the different algorithms to solve the recognition task our objective is to determine not only the optimum training set of sequences but the also the configuration of the conditions in the videos. The convenience of using the RGS algorithm due to its shorter training time can also be seen by comparing the computational complexity of the SOM algorithm (which is the one that achieved the best performance); the normal SOM algorithm complexity is $O(N^2)$, and for each the training epoch $O(K)$ [17].

However, there are things that can be improved, one thing is the aptitude function for measuring the ratio distance between inter-cluster and intra-cluster. This can be don by using better distance functions like Mahalanobis. Another thing that can be improved is the the inference mechanism that allow an intelligent growing of the RGS, this can be done by exploring more than just one level beneath the i-esim current level.

Acknowledgment

This research was partially funded by project number 51623 from “Fondo Mixto Conacyt-Gobierno del Estado de Tamaulipas” and the project number 78885 from “Ciencia Básica 2007” of Conacyt.

References

1. Mahmoudi, S., Daoudi, M.: A probabilistic approach for 3d shape retrieval by characteristic views. *Pattern Recogn. Lett.* 28(13), 1705–1718 (2007)
2. Yang, H.D., Lee, S.W.: Reconstruction of 3d human body pose from stereo image sequences based on top-down learning. *Pattern Recogn.* 40(11), 3120–3131 (2007)

3. Chen, H., Bhanu, B.: 3d free-form object recognition in range images using local surface patches. *Pattern Recogn. Lett.* 28(10), 1252–1262 (2007)
4. Dimitrova, N., Golshani, F.: Motion recovery for video content classification. *ACM Trans. Inf. Syst.* 13(4), 408–439 (1995)
5. Dimitrova, N.: Content classification and retrieval of digital video on motion recovery. PhD thesis, Tempe, AZ, USA (1995)
6. Fleischman, M., Decamp, P., Roy, D.: Mining temporal patterns of movement for video content classification. In: *MIR 2006: Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pp. 183–192. ACM, New York (2006)
7. Resta, M.: Self-organizing maps and financial forecasting: an application, pp. 185–216 (2002)
8. Céréghino, R., Park, Y.S.: Review of the self-organizing map (som) approach in water resources: Commentary. *Environ. Model. Softw.* 24(8), 945–947 (2009)
9. Ekanayake, J., Pallickara, S., Fox, G.: Mapreduce for data intensive scientific analyses. In: *ESCIENCE 2008: Proceedings of the 2008 Fourth IEEE International Conference on eScience*, Washington, DC, USA, pp. 277–284. IEEE Computer Society, Los Alamitos (2008)
10. Huang, P., Wang, Y., Shao, M.: A new method for multi-view face clustering in video sequence. In: *ICDMW 2008: Proceedings of the 2008 IEEE International Conference on Data Mining Workshops*, Washington, DC, USA, pp. 869–873. IEEE Computer Society, Los Alamitos (2008)
11. Biehl, M., Ghosh, A., Hammer, B.: Dynamics and generalization ability of lvq algorithms. *J. Mach. Learn. Res.* 8, 323–360 (2007)
12. Ghosh, A., Biehl, M., Hammer, B.: Performance analysis of lvq algorithms: a statistical physics approach. *Neural Netw.* 19(6), 817–829 (2006)
13. Cathey, R.J., Jensen, E.C., Beitzel, S.M., Frieder, O., Grossman, D.: Exploiting parallelism to support scalable hierarchical clustering. *J. Am. Soc. Inf. Sci. Technol.* 58(8), 1207–1221 (2007)
14. Suib, D.S., Deris, M.M.: An efficient hierarchical clustering model for grouping web transactions. *Int. J. Bus. Intell. Data Min.* 3(2), 147–157 (2008)
15. Kalouptsidis, N., Theodoridis, S.: Adaptive system identification and signal processing algorithms. Prentice-Hall, Inc., Upper Saddle River (1993)
16. Duda, R.O.: Elements of pattern recognition. Prentice Hall Press, Upper Saddle River (1994)
17. Kaski, S.: Comparing self-organizing maps. In: Vorbrüggen, J.C., von Seelen, W., Sendhoff, B. (eds.) *ICANN 1996. LNCS*, vol. 1112, pp. 809–814. Springer, Heidelberg (1996)

Author Index

- Adeli, Hojjat 167
Agrawal, Prateek 1
Alanis, Alma Y. 113
- Baig, A. Rauf 339
Banaie, Masood 11, 379
Belmonte-Izquierdo, Rubén 123, 249
Berrospe, Edgar 239
Bhattacharyya, Shankar Prasad 259
- Cai, Gang 483
Cai, Lei 103
Carlos-Hernández, Salvador 123, 249
Castellanos Sánchez, Claudio 443, 519
Chai, Tianyou 155
Chaudhury, Pinaki 259
Chen, Huayi 319
Chen, Tianping 45
Chen, Xiukuan 423
Chen, Xueyou 279
Cleofas, Laura 331
Cong, Qiumei 155
Cruz, Jesús M. de la 179
Cruz-Cortés, Nareli 349
Cuevas, Erik 309
Cui, Guangzhao 359
- del Angel-Guerrero, Edgar 443
Dong, Xiangjun 423
Duan, Haibin 269
- Gallardo-Hernández, Ana Gabriela 219
Gallegos Funes, Francisco 493
Gao, Song 289
Garcia, Enrique 407
Garrido, Rubén 463
Garrido-Moctezuma, Ruben 509
Geethal, B.B. Akila 21
Geng, Runian 423
Gharibzadeh, Shahriar 11
Ghosh-Dastidar, Samanwoy 167
Gomez, Angel D. 369
González, Claudia 189
González-Olvera, Marcos Angel
219, 239
- Guijarro, María 179
Gunasekara, R. Chulaka 21
Guo, Dongwei 433
Gutiérrez-Frías, Oscar Octavio 509
- Han, Zhiwei 387
Hassan, Mafaz 21
Herrera, P. Javier 179
Honggui, Han 73
Huang, Lan 319
- Islas-Andrade, Sergio 219
- Juárez, C. 331
Juàrez Gracia, Antonio Gustavo 493
Junfei, Qiao 73
- Kamran, Muhammad 339
- Lee, Jaewan 63, 83
Lee, Jung-sik 83
Lee, Malrey 63
Lee, Wei-Po 93
León-Javier, Alejandro 349
Li, Cuiling 359
Li, Haobin 359
Li, Jie 415
Li, Xiaoguang 359
Li, Xin 415
Lin, Xiaofeng 483
Liu, Gui-Xia 299
Liu, Huaping 209
Liu, Miao 433
Liu, Zhigang 387
López-Cárdenas, Rodrigo 453
Lu, Bao-Liang 133
Lugo, Raul 369
- Ma, Jie 433
Ma, Yuming 423
Ma, Z.M. 229
Martínez-Arenas, Tomás 443
Martínez-García, Juan Carlos 509
Marwala, Tshilidzi 53, 199
Masood, Sohail 339

- Mateo, Romeo Mark A. 63
 Mathew, C.D. Tharindu 21
 Mikaili, Mohammad 379
 Miranda, Roger 463
 Moreno-Armendáriz, Marco A. 349

 Nandy, Subhajit 259
 Naveed, Nawazish 339

 Osuna-Enciso, Valentín 309

 Pacheco, Jaime 369, 407, 503
 Pajares, Gonzalo 179
 Pantanowitz, Adam 53
 Park, Dong-Chul 145
 Patel, Pretesh B. 199
 Perera, A. Shehan 21
 Pérez-Cisneros, Marco 309
 Pooyan, Mohammad 11, 379

 Revilla-Monsalve, Maria Cristina 219
 Romero Herrera, Rodolfo 493
 Rubio, Jose de Jesus 369, 407, 503
 Ruiz, Issachar 473
 Ruz, José J. 179

 Salazar, Martin 369
 Salvo, Michael Angelo G. 83
 Sanchez, Edgar Nelson 113, 123, 249
 Sánchez, M. 473
 Sánchez-Fernández, Luis Pastor 453
 Sánchez Orellana, Pedro Luis 443, 519
 Sarbaz, Yashar 11
 Shahzad, Farrukh 339
 Shang, Fengjun 397
 Shi, Yulong 289
 Shukla, Anupam 1
 Song, Shaojian 483
 Suárez-Guerra, Sergio 453
 Subasinghe, Harsha 21
 Sun, Fuchun 209

 Tang, Yu 219, 239
 Tineo, Leonid 189
 Tiwari, Ritu 1
 Torres Jiménez, José 519
 Towhidkhah, Farzad 11

 Urrutia, Angélica 189

 Valdovinos, Rosa Maria 331, 473
 Villegas, Gerardo 503

 Wang, Han 299
 Wang, Hong 103
 Wang, Kangping 433
 Welikala, Yohan 21
 Wimalasena, Lasantha 21
 Woo, Dong-Min 145
 Wu, Changying 415
 Wu, Fengge 209
 Wu, Wei 45

 Xiao, Bo 289
 Xing, Zhihui 269
 Xu, Chunfang 269
 Xu, Jiadong 415

 Yan, Guohua 35
 Yang, Bin 319
 Yang, Tsung-Hsien 93
 Yang, Yang 133, 289
 Yu, Wen 155

 Zaldívar, Daniel 309
 Zeng, Sanyou 289
 Zhang, Jianhua 103
 Zhang, Xuncaï 359
 Zhao, Faxin 229
 Zheng, Ming 299
 Zhong, Wei 387
 Zhou, Chun-Guang 299, 319, 433
 Zhu, Yongsheng 35