# Soft Computing and Modeling Approaches for Complex Systems and Signal Processing

# **Research in Computing Science**

## **Series Editorial Board**

**Editors-in-Chief:** 

Grigori Sidorov (Mexico) Gerhard Ritter (USA) Jean Serra (France) Ulises Cortés (Spain) Associate Editors:

Jesús Angulo (France) Jihad El-Sana (Israel) Alexander Gelbukh (Mexico) Ioannis Kakadiaris (USA) Petros Maragos (Greece) Julian Padget (UK) Mateo Valero (Spain)

Editorial Coordination: María Fernanda Rios Zacarias

**Research in Computing Science** es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 103,** Octubre 2015. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. : 04-2005-121611550100-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor responsable: Grigori Sidorov, RFC SIGR651028L69

**Research in Computing Science** is published by the Center for Computing Research of IPN. **Volume 103**, October 2015. Printing 500. The authors are responsible for the contents of their articles. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research. Printed in Mexico City, in the IPN Graphic Workshop – Publication Office.

Volume 103

# Soft Computing and Modeling Approaches for Complex Systems and Signal Processing

Sabu M. Thampi Christian Callegari (eds.)









Instituto Politécnico Nacional, Centro de Investigación en Computación México 2015

# ISSN: 1870-4069

Copyright © Instituto Politécnico Nacional 2015

Instituto Politécnico Nacional (IPN) Centro de Investigación en Computación (CIC) Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal Unidad Profesional "Adolfo López Mateos", Zacatenco 07738, México D.F., México

http://www.rcs.cic.ipn.mx http://www.ipn.mx http://www.cic.ipn.mx

The editors and the publisher of this journal have made their best effort in preparing this special issue, but make no warranty of any kind, expressed or implied, with regard to the information contained in this volume.

All rights reserved. No part of this publication may be reproduced, stored on a retrieval system or transmitted, in any form or by any means, including electronic, mechanical, photocopying, recording, or otherwise, without prior permission of the Instituto Politécnico Nacional, except for personal or classroom use provided that copies bear the full citation notice provided on the first page of each paper.

Indexed in LATINDEX, DBLP, and Periodica

Printing: 500

Printed in Mexico

# Preface

In this issue, we present papers on soft computing and modelling approaches for complex systems and signal processing. This issue includes nine papers selected after peer review process. A summary of papers is as follows.

"UML Behavioral Refactoring for the Specification of Complex Software Systems," by M.T. Chitra and S. Elizabeth, proposes a generic framework for effective code generation from UML models which serves as an interexchange format to combine both structural and behavioral constraints of associated system objects to facilitate consistent source code generation.

"Accuracy of Artificial Neural Network Models of Software Reliability Growth –A Survey," by M. K. Saley and S. Sreedharan, discusses predicting errors of Artificial Neural Networks-based Software Reliability Growth Models (ANN SRGM). This survey concludes that ANN SRGM is better than statistical models for reliability prediction.

"Statistical and alignment based methods for comparison of non-coding DNA sequences," by Kouser and L. Rangarajan, proposes few techniques that capture the information regarding the arrangement of the motifs to assist in the analysis of the promoter sequences.

"On Rate Adjustment Mechanism for Reliable Multicast Transmission in ForCES," by L. Gong, Y. Wang, and C. Li discusses the issues of the reliable multicast rate adjustment mechanisms within Forward and Control Element Separation (ForCES), and proposes a rate adjustment mechanism of control unit. The paper also presents a design of a congestion feedback mechanism based on the feedback representative set.

"Electronic system for orientation control and obstacle avoidance for underwater glider without a rudder," by Y. N. Patil, V. S. Aneesh, M. T. Abhilash, and S. Kadlag, proposes an approach for achieving better propagation of autonomous underwater gliders with the help of orientation sensors.

"A New Approach to Adaptive Power Line Interference Removal," by A. R. Kasetwar and S. M. Gulhane, conducts a study on various methods for power line interference (PLI) removal for biomedical signal processing. Few algorithms are analyzed for their applications in adaptive power line interference removal.

"Development of Dual-Actuated Stage for Positioning Applications," by P. Niranjan, K. V. S. S. S. S. Sairam, and S. Karinka, presents and validates a model for describing the nonlinearity of the piezoelectric actuator.

"Concatenated Tabla Sound Synthesis to Help Musicians," by U. K. Roy proposes a scheme to synthesize pre-recorded tabla sounds to help musicians.

"Chronological Advancement in Compiler Design: A Review", by A. Verma and N. Bakshi carries out a brief survey on key properties of compiler courses in some universities.

Sabu M. Thampi

Associate Professor, School of CS/IT

Indian Institute of Information Technology and Management - Kerala (IIITM-K) Technopark Campus, Trivandrum-695581, Kerala, India E-mail: smthampi@ieee.org

5

Christian Callegari

TLC NETworks research GRoup Dept. of Information Engineering University of Pisa, Italy E-mail: christian.callegari@iet.unipi.it

Guest Editors

# **Table of Contents**

On Rate Adjustment Mechanism for Reliable Multicast Transmission in ForCES Liang Gong, Yining Wang, Chuanhuang Li	.9
Statistical and Alignment Based Methods for Comparison of Non-Coding DNA Sequences	21
Concatenated Tabla Sound Synthesis to Help Musicians	29
A New Approach to Adaptive Power Line Interference Removal	41
UML Behavioral Refactoring for the Specification of Complex Software Systems	51
Electronic System for Orientation Control and Obstacle Avoidance for Underwater Glider without a Rudder	59
Accuracy of Artificial Neural Network Models of Software Reliability Growth – A Survey	33
Development of Dual-Actuated Stage for Positioning Applications	<del>)</del> 3
Chronological Advancement in Compiler Design: A Review	<del>)</del> 9

7

# On Rate Adjustment Mechanism for Reliable Multicast Transmission in ForCES

Liang Gong<sup>1</sup>, Yining Wang<sup>2</sup>, and Chuanhuang Li<sup>1</sup>

<sup>1</sup>College of Information & Electronic Engineering Zhejiang Gongshang University, Hangzhou, Zhejiang Province, China 756695230@qq.com, chuanhuang\_li@mail.zjgsu.edu.cn <sup>2</sup>College of Computer Science and Technology, Zhejiang University, Hangzhou, Zhejiang Province, China 3110102753@zju.edu.cn

**Abstract.** In the specific environment for reliable multicast transmission of ForCES protocol messages, this paper firstly analyzed the congestion control difficulties and problems in the process of the reliable multicast transmission, and then designed and proposed a reliable multicast rate adjustment mechanism which meets the architectural requirements of ForCES routers. Test results also show that the mechanism basically meets the relevant performance requirements.

Keywords: ForCES; Reliable Multicast; Congestion Control; Rate Adjustment.

## **1** Introduction

ForCES (Forward and Control Element Separation) Working Group [1] is one of IETF working groups in the field of routing, which has committed to the research and development of open programmable router architecture and protocol since its inception. ForCES router architecture is composed by the control unit (CE) and the forwarding unit (FE) in the structure, and CE and FE can transmit information through the ForCES protocol.

Within ForCES routers, one CE needs multiple FEs to transmit protocol messages. Compared with multiple TCP unicasts, IP multicast can save bandwidth between ForCES transport mapping layers. Because the protocol message is only copied at the branching node of the multicast tree, it avoids CE producing redundant protocol message packets, and reduces the load of the CE port and improves the transmission efficiency. However, as the multicast transmission is based on UDP protocols, CE transmits protocol messages at a fixed rate, if the network environment is deteriorated, the sending rate of CE cannot be adjusted to the protocol messages flow, which may lead to a further deterioration of the network environment, or even a collapse. There are already a number of multicast congestion control protocols, such as Pragmatic General Multicast Congestion Control (PGMCC) [2], TCP-friendly Multicast Congestion Control (TFMCC) [3], Receiverdriven Layered Multicast (RLM) [4], and other agreements and so on.

9

#### Liang Gong, Yining Wang, Chuanhuang Li

Currently, there are mainly two types of reliable multicast transport protocols which are based on acknowledgment (ACK) or based on negative acknowledgment (NACK). Reliable multicast protocol messages within ForCES adopt the mechanism of error detection and recovery, which is based on the negative acknowledgment (NACK), and it can better ensure the reliable transmission of multicast protocol messages.

In order to ensure the friendly fairness to TCP, Ma *et al.* [5] proposed Fair Active Congestion Control (FACC), which starts with a single rate by the receiver, and takes a transmission rate as congestion control parameters. In order to increase the robustness and adaptivity of the algorithm, the protocol took congestion parameter convergence strategy to redesign the filtering algorithm of the congestion control parameter. To eliminate the phenomenon of "slowest priority", Zhai, Wu and Gu [6] proposed two mechanisms that are Composite Multicast Congestion Control (CMC) and Layered Multicast (LM) Congestion Control by the Particle Swarm Optimization (PSO) (LM-PSO). Both of these mechanisms do not use a single-rate congestion control algorithm, but use an algorithm where each layer of the Layered Multicast adjusts the transmission rate dynamically.

In this paper, in view of the research of the Multicast Congestion Control Mechanisms, we have taken the detailed analysis and research about the issues of the reliable multicast rate adjustment mechanisms within ForCES, and adopted the Wireshark capturing software to test the traffic conditions of the CE port during the process of protocol messages reliable multicast. What is more, by drawing a graph of throughput situations of data packets, we analyzed the problem of reliable multicast about fairness and heterogeneity within ForCES.

# 2 Main Problem of the Reliable Multicast Rate Adjustment Mechanism within ForCES Routers

Similar to the TCP congestion control, currently the ForCES routing channel protocol message is based on multicast on IP layer. The reliable multicast protocol message within ForCES adopts the point-to-point rate adjustment mechanism. Therefore, protocol messages reliable multicast rate adjustment mechanism within ForCES can learn from the TCP congestion control mechanism absolutely, research from the regulator parameters, feedback mechanisms, and other aspects of rate regulation algorithm. However, considering the environment of reliable multicast is much more complex than that of unicast, TCP congestion control mechanisms cannot be applied simply on reliable multicast, and the specific scenarios of reliable multicast within ForCES must be combined to carry out a detailed analysis and design for each point.

Another issue of the reliable multicast rate adjustment mechanism within ForCES that must be solved is fairness, which appears much more important in multicast environment. For reliable multicast, the fairness of congestion control mechanism has two meanings. On the one hand, the fairness is that, the protocol message stream that transmits in the form of TCP shares the network bandwidth with a reliable multicast stream. On the other hand, the fairness is that each FE of multicast receivers is fair. For example, the sending rate of CE cannot be decided by the FE which has the worst network condition, and this is obviously unfair for other FE ports. Therefore, the rate adjustment

mechanism of reliable multicast protocol messages within ForCES must satisfy the heterogeneity between each FE. In order to achieve intra protocol fairness, and to ensure the sufficient and effective use of the network bandwidth of ForCES transport mapping layer, all of these problems should be considered and designed in the research of this paper.

# **3** Analysis of Concrete Problems of Reliable Multicast Rate Adjustment Mechanism within ForCES Routers

#### 3.1 Preferences of Reliable Multicast Rate Regulation

The CE sender of the reliable multicast within ForCES relays on the hardware platform, the transmission rate of which is impacted by the processor hardware itself. If adopting the rate adjustment mechanism based on the sending frequency, the network processor of CE should be made certain modifications on the kernel, and every rate adjustment process should be switched between the kernel space and the user space constantly, which will increase the expense of time, and cannot meet the real-time requirement of the reliable multicast rate adjustment mechanism. Therefore, the reliable multicast within ForCES adopts a rate adjustment based on a window parameter. In order to adjust the rate, and to achieve the rate adjustment mechanism by the software, and to avoid modifying hardware, CE can adjust the size of the sending window according to the feedback messages sent by FE, so that the whole rate adjustment mechanism can have greater flexibility and feasibility.

Referring to TCP congestion control mechanisms, during the process of reliable multicast within ForCES, the size of a sending window maintained by CE is set as  $W_0$ , and then simulating the AIMD mechanism to adjust the size of the sending window. For reliable multicast — a one-to-many model in transmission, CE cannot grasp the network congestion of FE timely and accurately. When the size of sending window increases in an additive way, it will become too large and cause a network congestion. As this result, we cannot introduce TCP congestion control mechanisms to the multicast environment simply.

We use the method that setting a sending threshold denoted as W' at CE, which is used to adjust the size of the sending window. The sending threshold is the maximum size of the sending window at CE during the process of reliable multicast, that is to say, when the size of the sending window has reached the threshold, namely W', if we continue to increase the size of the sending window, the efficiency of multicast will not be improved. So a sending threshold can be set according to the size of the measured values for each multicast test or empirical values. Thus, the adjustment parameter for CE should be set between the sending window and the threshold, the sending window represents the size of each multicast initialization process, namely the size of the minimum window. If the size of the sending window is less than this value, FE may receive multicast messages abnormally, which will waste the multicast network bandwidth, so that the normal performance of ForCES routers cannot be guaranteed.

In conclusion, the parameter selection of the reliable multicast rate adjustment within ForCES bases on the following principles: the initial sending window size of CE, denoted as *W*0. When *W*0 is lower than the available network bandwidth, denoted

#### Liang Gong, Yining Wang, Chuanhuang Li

as W1, increases the size of sending window rapidly; when W0 is between W1 and W', increases the size of sending window slowly to avoid the network congestion; when the network is congested, reduces the size of the sending window, but when W0 is between W1 and W', decreases slowly to avoid the rate oscillation.

#### 3.2 Fairness Problem of Reliable Multicast Transmission

Among reliable multicast protocols within ForCES, the control protocol message sent in the form of multicast competes with the control protocol message within TML in the bandwidth in the rigid way. When the network environment becomes deterioration, the control protocol message flow transmitted in the form of TCP will reduce the sending rate according to the AIMD mechanism, but the multicast protocol message flow will continue being transmitted in the original rate, or even a greater rate. Because of the lack of the corresponding rate adjustment mechanism, it will lead to the collapse of network bandwidth. Therefore, in order to achieve the fairness of TCP, a window mechanism similar to TCP's should be taken, but the AIMD window adjustment mechanism of TCP has a great jitter [7], which is not conducive to transmitting the multicast protocol message. So the AIMD mechanism cannot be applied simply in the multicast environment within ForCES. In order to get an ideal control curve of the sending rate, we should improve the AIMD mechanism, for example, we can add a series of adjustment rules based on the history records to make it smoother.

In the curve of the TCP AIMD mechanism, the increasing factor is set as b, the multiplicative factor is set as a, the size of the sending window is a time function whose feedback time interval is set as t, during the cycle of the sending window, the function is denoted as follow:

$$W(t) = \int_{b^{*}(t-t_{0})}^{W_{0}*a^{t}} \int_{b>0}^{0} dt dt = 0.$$
(1)

Herein,  $W_0$  is the initial value of the sender congestion window,  $t_0$  is the initial time, t is an integer multiple of the feedback time which is also called *RTT*, namely, t = n\*RTT (n = 0,1,...), so the curve is discrete. However, in order to facilitate the analysis, ignoring the effect of timeout and retransmission during the practical transmission, the discrete curve can be simplified as a continuous curve.

From Equation (1), the control curve of AIMD mechanism under the congestion condition obeys the exponential function curve with a as its base, when t is large enough, the size of the window is close to 0, but under the non-congestion conditions it obeys the linear function curve with b as its slope.

According to above analysis, the exponential curve of AIMD mechanism declines much more rapidly at first, when the value is close to the lower limit, it declines slower, the characteristic of which is more appropriate for the protocol message multicast transmission within ForCES. Considering that, during the protocol message multicast transmission process, the initial sending window of CE is denoted as  $W_0$ , the sending window function of CE under the congestion condition is adjusted as:

$$W(t) = W_0 * a + W_0.$$
(2)

At the sending window adjusting algorithm of CE, we adopt the multiplicative decrease method under the congestion condition:

$$W(t) + RTT = a^* (W_t - W0 + W_0) \quad 0 < a < 1.$$
(3)

In order to be friendly to TCP, we adopt the method which uses a maximum value to limit the additive increase under the non-congestion conditions:

$$t + RTT = \min\{W', W, +b\}\ b > 0.$$
 (4)

Under this window adjustment algorithm, the curve of the transmission rate is linear increasing, and will stop at the sending threshold, denoted as W'. As a result, it cannot achieve the initial requirements. Therefore, the rising amplitude of the additive increasing function curve changes from the constant to the variable, the rising amplitude of which is related to the difference between its current value and sending threshold. So we can design as follow:

$$W_t + RTT = W_t + b(W' - W_t) \quad 0 < b < 1.$$
 (5)

We can know from above, the function of Equations (5) and (3) is complementary, and the effect is the opposite, both of which have decided the sending window curve rising at a faster rate at the initial time, and then the rate of the rising becomes much slower due to the value of the sending window is close to the sending threshold. When b is set as an appropriate value, we can get an ideal performance of the curve, as shown in Fig. 1. By the testing, we can see that if the mechanism of the AIMD is improved, and considered strengthening the smoothing of the original curve, it will have smaller amplitude than that of the original curve. However, it is still the additive increase and multiplicative decrease in theory, and friendly to the TCP. So it can satisfy the fairness requirement perfectly.



Fig. 1. Rate curve compared between AIMD and improved AIMD.

#### 3.3 Principles of Feedback and Rate Adjustment Mechanism

The reliable multicast within ForCES takes the way of feedback (which combines with the network congestion trends according to the network bandwidth) to adjust the sending rate.

13

#### Liang Gong, Yining Wang, Chuanhuang Li

The principles which should be adopted during the reliable multicast within ForCES are as follows:

- 1. FE makes the congestion judgment according to the average packet-dropout interval packets during each multicast process, and sends feedback to CE;
- CE makes the judgment about the trend of the network bandwidth by grasping the network congestion of FE and combining the current sending rate with the historical rate during the last multicast, and adjusts the sending rate according to the change of the network bandwidth;
- CE updates the history records of multicast after every sending rate adjustment and keep the history records, in preparation for the next sending rate adjustment.

In order to reflect the congestion situation of the current network correctly, we adopt an exponential weighted moving average method for the historical packet-dropout rate to judge the condition of the network congestion.

In order to assist the judgment of the network bandwidth variation trend, the reliable multicast within ForCES needs keeping a group of history records, denoted as  $H = \{S_c, S_{nc}\}$ , wherein,  $S_c$  stands for the control status of the sending rate under the recent condition of the multicast congestion, and  $S_{nc}$  stands for the control status of the sending rate under the recent condition of the recent condition of the non-congestion. At the beginning of the reliable multicast, the trends of the network bandwidth of CE, denoted as *B*, are documented as follows:

- B = 0: The available network bandwidth is stable;
- B = 1: The available network bandwidth is increasing;
- B = -1: The available network bandwidth is decreasing.

The trend of network bandwidth is judged by the rules which are as follows:

- 1. At beginning, the size of the sending window at CE is denoted as  $W_0$ , if receiving the congestion feedback from FE during the multicast transmission process, and the size of the current sending window, denoted as W, is less than the value of  $S_c$  in the history of H, the network bandwidth will be considered decreasing, so B = -1;
- 2. If not receiving the congestion feedback from FE during the multicast transmission process, and the size of the current sending window, denoted as W, is larger than the value of  $S_{nc}$  in the history of H, the network bandwidth will be considered increasing, so B = 1;
- 3. If the size of the current sending window of CE, denoted as W, is between  $S_c$  and  $S_{nc}$ , or does not meet the two necessary conditions of above two at the same time, the network will be considered stable, so B = 0.

As a result, the reliable multicast rate adjustment algorithm within ForCES, which is based on the sending window, is as follow:

1. If the network bandwidth variation trend is identified as B = -1, updating the size of the sending window according to Equation (5);

- 2. If the network bandwidth variation trend is identified as B = 1, updating the size of the sending window according to Equation (3);
- 3. If the network bandwidth variation trend identified as *B* = 0, updating the size of the sending window according to Equation(6);

$$W = k * S_c + (1-k) * S_{nc} \quad 0 < k < 1.$$
(6)

The history records of CE will be updated after the sending rate being adjusted. If B = 0, this problem will be divided into two parts, if the packet-dropout feedback of FE is not received, and the value of  $S_{nc}$  in the history of H is less than the size of the current sending window, the value of  $S_{nc}$  will be updated to equal the size of the current sending window; if the packet-dropout feedback of FE is received, and the value of  $S_c$  in the history of H is larger than the size of the current sending window, the value of  $S_c$  will be updated to equal the size of the current sending window, the value of  $S_c$  will be updated to equal the size of the current sending window. If B = -1, the value of  $S_c$  will be set as  $W_0$ , and the value of  $S_{nc}$  will not be changed. If B = 1, the value of  $S_{nc}$  will be set as W', and the value of  $S_c$  will not be changed. So the multicast message curve diagram of CE through the simulation analysis can be shown in Fig. 2:



Fig. 2. CE sender multicast message curve.

#### 3.4 Heterogeneity Problem of Reliable Multicast Transmission

Heterogeneity is also called the intra protocol fairness. In the current design of the reliable multicast congestion control, the sender selects its sending rate according to the bandwidth requirement of the worst receiver, namely the sending rate of the receiver is the minimum transmission rate, which is called the Worst-path fairness [7]. The transport model of ForCES protocol messages reliable multicast is an one-to-many communication model, where there are many FE ports at the multiple receiver, and each FE has a different utilization status to the network bandwidth in the different path, and a FE receiver in congestion should not affect the rate of the other FE receivers to receive the multicast packets, and allows each FE receiver to select the received bandwidth according to the congestion status of its own network. So this is the intra protocol fairness, which the reliable multicast rate adjustment mechanism within ForCES should be able to meet with.

15

#### Liang Gong, Yining Wang, Chuanhuang Li

The rate regulation feedback mechanism of the protocol message multicast within ForCES adopts the mechanism which selects a set of worse FE receivers to represent a multicast congestion conditions. During the reliable multicast process of intra protocol messages, initially, CE sends the Session Messages to FE, and FE does not calculate the packet-dropout at the same time, so FE does not send the feedback message. When the second session message arrives, every FE sends the ACK feedback packets which include the packet-dropout during the last multicast process and the RTT timestamps. Among them, because the transmission of session messages between CE and FE is the one-way, the session message is transmitted by TCP. But, each FE forwarding element may serve as an important role in the architecture of ForCES router, as a result, each FE must be ensured receiving protocol messages reliably. However, because the function of the ACK feedback message is to feedback the congestion status to CE, the feedback message is transmitted by UDP. As CE has the overall grasp of the receiving capacity and throughput of each FE to coordinate the rate adjustment and avoid the congestion, losing the ACK feedback message is allowed under the premise of satisfying certain QoS.

We select a group of ACK feedback messages from the session packets dynamically every time, which is required not only to represent the bandwidth utilization and throughput under poor reception conditions of FE, but not to cause an intra protocol to be unfair because of the excessive inhibition of the sending rate of CE. As shown in Fig. 3:



Fig. 3. Feedback set of multicast within ForCES.

#### **Performance Testing** 4

#### Fairness Testing of Reliable Multicast Transmission 4.1

Operating the reliable multicast testing procedures at FE and CE separately, the CE test program joins a multicast group through the IGMP protocol, of which the ID is 0xC0000000, the IP address is 233.4.4.4 and the members include three hosts, whose IP addresses are 10.20.0.59, 10.20.0.190, 10.20.0.229 respectively, and sends 1000 multicast protocol messages to 0xC0000000. On the other hand, setting the parameters

of the SmartBits600 to make it send a UDP message at a rate of 10 *Kb/s* within ForCES, and keep sending for 20 min. Then, in order to reflect that the process of the actual network status is turning good from bad, continue to send the random constructed UDP messages within ForCES at a rate of 50 *Kb/s*. The Wireshark software is used to capture the packets at the CE multicast protocol message port, and the result is shown in Fig. 4:

4		eth1: Capturing - 1	Wireshark	_ 4
ile <u>E</u> dit <u>V</u> iew <u>G</u> o <u>C</u> aptu	ire <u>A</u> nalyze <u>S</u> tatistics Te	elephony <u>T</u> ools <u>H</u> elp		
🛿 🔤 🗟 😂 I		i 🗛 🔶 🤿	- 🛧 🛨 📄 📑 🔍	2, o, o, 🖭   🕁 🔟 🚺 👘
Filter: forces		✓ •]• Exp	ression 🤞 Clear 🛷 Apply	
. Time	Source	Destination	Protocol - Info	
8028 716.537001	10.20.0.229	10.20.0.135	ForCES Message Type:	QueryResponse, Total Length: 60 Byte
8029 716.570730	10.20.0.135	10.20.0.229	ForCES Message Type:	Query, Total Length: 52 Bytes
8030 /10.59/033	10.20.0.229	10.20.0.135	ForCES Message Type:	QueryKesponse, lotal Length: 60 Bytes
8032 716 658009	10.20.0.135	10.20.0.225	ForCES Message Type.	QueryResponse Total Length: 68 Bytes
8033 716.691803	10.20.0.135	10.20.0.229	ForCES Message Type:	Ouery. Total Length: 52 Bytes
8035 716,720619	10.20.0.229	10.20.0.135	ForCES Message Type:	OuervResponse. Total Length: 60 Byte
8036 716.738851	10.20.0.135	10.20.0.229	ForCES Message Type:	Config, Total Length: 68 Bytes
8037 716.760796	10.20.0.229	10.20.0.135	ForCES Message Type:	ConfigRepsonse , Total Length: 68 By
				)
concruce as, ore, measure	NJ_04.30.00 (00.00.40.04.	solcor, osc. ip canki		,
Internet Protocol, Src:	10.20.0.135 (10.20.0.135)	, Dst: 10.20.0.229 (16	0.20.0.229)	
Transmission Control Pro	tocol, Src Port: hbci (36	00), Dst Port: 38092 (	(38092), Seq: 935, Ack: 2535	, Len: 62
Transmission Control Pro Source port: hbci (300	tocol, Src Port: hbci (36 )0)	00), Dst Port: 38092 (	(38092), Seq: 935, Ack: 2535	, Len: 62
Transmission Control Pro Source port: hbci (300 Destination port: 3809	tocol, Src Port: hbci (36 00) 02 (38092)	00), Dst Port: 38092 (	(38092), Seq: 935, Ack: 2535	, Len: 62
Transmission Control Pro Source port: hbci (300 Destination port: 3809 [Stream index: 148]	tocol, Src Port: hbci (36 )0) )2 (38092)	00), Dst Port: 38092 (	(38092), Seq: 935, Ack: 2535	, Len: 62
Transmission Control Pro Source port: hbci (300 Destination port: 3809 [Stream index: 148] Sequence number: 935	tocol, Src Port: hbci (36 00) 02 (38092) (relative sequence num	00), Dst Port: 38092 ( ber)	(38092), Seq: 935, Ack: 2535	, Len: 62
Transmission Control Pro Source port: hbci (300 Destination port: 3809 [Stream index: 148] Sequence number: 935	tocol, Src Port: hbci (36 )0) )2 (38092) (relative sequence num	000), Dst Port: 38092 ( ber)	(38092), Seq: 935, Ack: 2535	, Len: 62
Transmission Control Pro Source port: hbci (300 Destination port: 3805 [Stream index: 148] Sequence number: 935 [Next sequence number: defended of second	tocol, Src Port: hbci (36 00) 22 (38092) (relative sequence num : 997 (relative sequence (relative sequence num	000), Dst Port: 38092 ( ber) ce number)]	38092), Seq: 935, Ack: 2535	, Len: 62
Transmission Control Pro Source port: hbci (300 Destination port: 3809 [Stream index: 148] Sequence number: 935 [Next sequence number: 0.00.21.27.98.cd as 00.0	tocol, Src Port: hbcl (38 00) 22 (38092) (relative sequence num 997 (relative sequence num 997 (relative sequence num 997 (relative sequence num 90 Ac h4 38 c0 08 80 45	00), Dst Port: 38092 ( ber) ce number)]	38092), Seq: 935, Ack: 2535	Len: 62
Transmission Control Pro Source port: hbci (300 Destination port: 3805 [Strean index: 148] Sequence number: 935 [Next sequence number: Admandadament and a sequence 0 00 21 27 98 c6 la 00 0 0 00 72 7 9 3 40 00 40 0	tocol, Src Port: hbci (38 10) 22 (38092) (relative sequence num : 997 (relative sequen : 925 (relative sequen : 525 (relative set) 60 4c b4 38 c0 08 00 45 6 ad 5f 0a 14 00 87 0a	000), Dst Port: 38092 ( ber) ce number)] 00 .'' L.8E. 4rv.8.0.	38092), Seq: 935, Ack: 2535	, Len: 62
Transmission Control Pro Source port: hbci (300 Destination port: 3809 [Stream index: 148] Sequence number: 935 [Next sequence number: Actual Education Source Source 0 00 21 27 98 c 0 1 a 00 d 0 00 21 7 98 c 0 1 a 00 d 0 00 c 25 0 b 8 94 c c 34 d	tacol, Src Port: hbc1 (38 00) 22 (38092) (relative sequence num 2007 (relative sequenc	000), Dst Port: 38092 ( ber) ce number)] 00. '' L.8E. 14 .rw.0.0	38092), Seq: 935, Ack: 2535	Len: 62
Transmission Control Pro           Source port: hbci (300           Destination port: 3809           [Stream index: 148]           Sequence number: 935           [Next sequence number: 05           0         00 21 27 98 c0 la 00 e           0         00 21 27 98 c0 la 00 e           0         00 27 97 93 40 00 40 (0           0         00 c c d1 60 e 00 e0 e	tacol, Src Port: hbc1 (30 00) 12 (38092) (relative sequence num 997 (relative sequence 0525 (relative sequence 053 (relative sequence 05 (relative sequence 05 (relative sequence) 15 (relative sequence) 15 (relative sequence) 15 (relative sequence) 15 (relative sequence) 16 (relative sequence) 16 (relative sequence) 16 (relative sequence) 17 (relative sequence) 18 (relative sequence) 18 (relative sequence) 18 (relative sequence) 18 (relative sequence) 19 (	00), Dst Port: 38092 ( ber) ce number)] 00 .!'L.8E. 14 .rw.@ 184A 6.). 0	38092), Seq: 935, Ack: 2535	Len: 62
Transmission Control Pro:           Source port: hbci (300           Destination port: 3805           [Stream index: 148]           Sequence number: 935           [Next sequence number: 935           0 00 21 79 62 c0 1a 00 0           0 00 27 79 62 c0 1a 00 0           0 00 c2 c7 96 c0 1a 00 0           0 00 c2 c0 bb 89 4 cc 34 0           0 00 bc cd f0 00 0 31 0           0 82 26 00 32 10 30 0	taccl, Src Port: Nbc1 (38 00) 22 (38092) (relative sequence num: 997 (relative sequence 998 (relative sequence) 998 (relative sequence) 999 (relative	000), Dst Port: 38092 ( ber) ce number)] 000 .!' L.8E. 14 .rw.@.@ 184.6.), 08	38092), Seq: 935, Ack: 2535	, Len: 62
Transmission Control Pro           Source port: bbci 300         Destination port: 3805         [Stream index: 148]         Sequence number: 3805         INEX sequence number: 4803         Sequence number: 4803         Sequence number: 4803         Sequence number: 933         Next sequence number: 4803         Sequence number: 933         Sequence number: 934         Sequenc	tecol, Src Port: hbcl (30 00) 12 (38692) (relative sequence num 997 (relative sequence 	000), Dst Port: 38092 ( ber) ce number)] methen 00 .!' L.8Ē 4 .rw.8.0 14 .rw.8.0 01	38092), Seq: 935, Ack: 2535	, Len: 62
Transmission Control Pro           Source port: hbcl 1300         Destination port: bbcl 300         Destination port: 3800         Estimation port: 38000         Estimation port: 3800         Esti	tecol, Src Port: Abcl (30 0) 22 (38092) (relative sequence num: 997 (relative sequence 100 cm (100 cm	000), Dst Port: 32092 ( ber) ce number)] 1000 -!L.SE. 14	38092), Seq: 935, Ack: 2535	, Len: 62
Transmission Control Pro           Source port: hbcl 1300         Destination port: bsci 1300         Jstream index: 1481           Squeence number: 935         [Hext sequence number: 935         Jstream index: 1481         Sequence number: 935           0 00 21 27 98 c0 1 00 00         00 21 27 79 4 40 60 49         00 00 10 27 77 34 40 60 49         00 00 00 00 00         00 00 00 00 00         00 00 00 00 00         00 00 00 00 00         00 00 00 00 00         00 00 00 00 00         00 00 00 00 00         00 00 00 00 00         00 00 00 00 00 00         00 00 00 00 00         00 00 00 00 00         00 00 00 00         00 00 00 00         00 00 00 00         00 00 00 00         00 00 00 00         00 00 00 00         00 00 00 00         00 00 00 00         00 00 00 00         00 00 00 00         00 00 00 00         00 00 00 00         00 00 00 00         00 00 00         00 00 00 00         00 00 00 00         00 00 00 00         00 00 00 00         00 00 00 00 00         00 00 00 00         00 00 00 00 00         00 00 00 00 00         00 00 00 00 00         00 00 00 00 00         00 00 00 00 00         00 00 00 00 00         00 00 00 00         00 00 00 00 00         00 00 00 00 00         00 00 00 00         00 00 00 00 00         00 00 00 00 00         00 00 00 00         00 00 00 00 00         00 00 00 00 00         00 00 00 00 00         00 00 00 00 00         00 00 00 00 00         00 00 00 00 00 <td>tecol, Src Port: Abc. [38 00] 22 (38092) (relative sequence num 997 [relative sequence 1097 [relative sequence 100 [relative seq</td> <td>000), Dst Port: 38092 ( ber) ce number)] methen 100</td> <td>30002), Seq: 035, Ack: 2535</td> <td>, Len: 62</td>	tecol, Src Port: Abc. [38 00] 22 (38092) (relative sequence num 997 [relative sequence 1097 [relative sequence 100 [relative seq	000), Dst Port: 38092 ( ber) ce number)] methen 100	30002), Seq: 035, Ack: 2535	, Len: 62
Transmission Control Pro           Source port: bbcl 306         Destination port: bbcl 306         Destination port: 386         Estimation porti	tecol, Src Port: Abc. [31 00] (calative sequence num 977 (retative sequence num 977 (retative sequence eb 4c b4 38 c0 08 00 45 64 ad 5f 0a 14 00 87 0a 11 36 08 08 29 d2 b9 80 13 68 08 08 29 d2 b9 80 13 68 08 08 14 d1 38 00 13 f8 40 00 00 10 00 00 13 68 40 00 10 10 00 00 13 68 00 08 10 11 00 00 00 7 61 12 00 08 02 00 00	000), Dst Port: 38092 ( ber) cc number)] numbers 14 . nv.@.@. 28	30092), Seq: 035, Ack: 2535	, Len: 62
Transmission Control Pro           Source port: bbcl 300         Destination port: bbcl 300         Destination port: 3800         Estimation port: 3800         Estima	tecol, Src Port: Abc. [36 00] 22 (3602)] (relative sequence num 997 [relative sequence 5526 col 80 00 45 66 ad 57 0m 14 00 70 0m 35 0m 14 00 00 00 00 00 31 00 00 00 10 00 00 31 60 01 00 12 00 10 00 31 00 01 00 05 02 00 00	000), Dst Port: 38092 ( ber) cc number)] 	38092), Seq: 035, Ack: 2535	, Len: 62
Transmission Control Pro           Source port: NoL: (300           Destination port: 308           Estrean index: 1481           Sequence number: 935           Next sequence number: 935           Do 07 27 73 24 00 40 0           0 00 21 37 96 cf lambor           0 00 21 37 96 cf lambor           0 00 22 77 32 40 00 40           0 00 50 cf b 00 00 11           0 00 50 cf b 00 00 11           0 00 20 cf 21 00 00 00           0 00 00 cf 00 00 00           0 00 00 cf 00 00 00 00           0 00 00 cf 00 00 00 00           0 00 00 00 00 00 00           0 00 00 00 00 00 00 00	tecol, src Port: Bec. 138 (2) (38092) (relative sequence num (relative sequence num ed at b4 38 c 00 00 00 00 ad 57 0 at 00 00 00 00 10 08 00 00 00 00 00 10 08 00 00 10 00 13 66 00 00 10 00 00 14 00 10 00 00 00 00 15 00 10 00 00 00 00 15 00 10 00 00 00 00 15 00 00 00 00 00 15 00 00 00 00 00 00 15 00 00 00 00 00 15 00 00 00 00 00 15 00 00 00 00 15 00 00 00 00 15 00 00 00 15 00 00 00 15 0	000), Dst Port: 38092 ( ber) ce number)] 00 14 - rnv.0,	38092), Seq: 035, Ack: 2535	. Len: 62
Transissen Control Fro Source port: NoL (300 Destination port: 300 (Stram Index: 140) Sequence number: 331 Sequence number: 330 0 00 21 27 98 c0 10 00 0 00 72 77 93 40 00 40 0 00 70 70 93 40 00 40 0 00 70 77 93 40 00 40 0 00 72 77 93 40 00 40 0 00 70 70 93 40 00 40 0 00 00 00 00 00 00 00 00 00 00 00 0 00 0	Use of the constraints         Sec Pert: Baci Ist           (2) (30992)         (relative sequence num           997         (relative sequence num           997         (relative sequence num           997         (relative sequence num           998         (relative sequence num           997         (relative sequence num           998         (relative sequence num           997         (relative sequence num           998         (relative sequence num           999         (relative sequence num           991         (relative sequence num           992         (relative sequence num           993         (relative sequence num           994         (relative sequence num           995         (relative sequence num           996         (relative sequence num           997         (relative sequence num           997         (relative sequence num	000), Dat Port: 33092 ( ber) cc nuber)] mokent 00	38092), Seq: 035, Ack: 2535	, Len: 62
Transmission Control For Source port NoL (306 Destination port: 306 Sequence multiple Sequence multiple 0 06 21 37 08 cd 18 06 0 06 21 37 08 cd 18 06 0 06 26 08 08 cc 34 0 08 06 08 08 cc 34 0 08 06 08 08 08 00 08 00 0 08 08 08 08 08 08 00 0 08 08 08 08 08 08 08 0 08 08 08 08 08 0 08 08 08 08 08 0 08 08 08 08 0 08 08 08 08 0 08 0 08 08 0 08 08 0 08	Start         Start <th< td=""><td>000), Dat Port: 38092 ( ber) ce number)] 00 10 10 10 10 10 10 10 10 10 10 10 10</td><td>38092), Seq: 035, Ack: 2535</td><td>, Len: 62</td></th<>	000), Dat Port: 38092 ( ber) ce number)] 00 10 10 10 10 10 10 10 10 10 10 10 10	38092), Seq: 035, Ack: 2535	, Len: 62
Transmission Control Fro           Source port: NoL: (300         Destination port: 308         Destinatio	tack1, src Port: Bac. 134           22 (38992)           (relative sequence nux           997         (relative sequence           987         (relative sequence)           984         100 00 00 00 00           985         100 00 00 00           986         100 00 00 00           987         11 20 00 00 00 00           988         100 00 00 00 00           997         11 20 00 00 00 00           997         12 00 00 02 00 00           998         Packets: 13800	000), Dat Port: 38092 ( ber) cc nuber)] mokent 00	38092), Seq: 935, Ack: 2535	, Len: 62

Fig. 4. Protocol message transmission within ForCES.

From the capturing result, during the first 20 minutes, the number of the multicast protocol message packets did not start from a small value, and soon increased to a more stable level in the initial stage, which is the result of that, CE adopted the sending window based on the history records and adopted the mimic AIMD mechanism which combines the congestion judgment and the trend of the network bandwidth, namely the adjustment parameters take use of the empirical values, which can make the reliable multicast sending window return to the optimal window size more quickly and smoothly. 20 minutes later, the size of the multicast messaging window will soon become a smaller level, and this size will be kept continuously after that. The reason why the sending rate is not excessive inhibition is that the congestion feedback mechanism based on the representative feedback set makes CE check the congestion and reduce the sending rate quickly.

#### 4.2 Heterogeneity Testing of Reliable Multicast Transmission

Operating the reliable multicast testing procedures at FE and CE separately, the CE test program joins a multicast group through IGMP protocol, of which the ID is 0xC0000000 and the IP is 233.4.4.4, and the members include three hosts, of which the IP is 10.20.0.59, 10.20.0.190, 10.20.0.229 respectively, and sends a multicast protocol message with a length of 8000 byte to 0xC0000000. On the other

17

#### Liang Gong, Yining Wang, Chuanhuang Li

hand, the UDP message constructed by SmartBits600 transports at the rate of 200 Kb/s, to simulate the actual network environment, the send status is shown in Fig. 5:

🗲 Applications Places System 🥹 🎕 🔤	<b>8</b> 2	۱	root s	5at Nov	5, 4:1	3 PM 🕼
root@localhost:/home/jkl/netcom_forces_sdk_1.2/app/fe/debug						• ×
File Edit View Terminal Tabs Help						
route add -net 19.1.0.0/32 gw 0.0.0.0						~
SIOCADDRT: Invalid argument						
INFO: TCP_CTRL channel is sending a Control Msg(Length:64) to CE(0x400000000). @S						
ar Nov 5 10:10:24 2011 INFO: TCP CTRL channel recieve a Control Msg(Length:72) from CE(0x400000000), @Sa						
t Nov 5 16:10:24 2011						
In the Function AttriEntrySet PrefixTable, the dwNextHopID is 101						
In the Function AttrientrySet_Prefixiable, the dwBladeLD is 1						
route add -net 19.1.0.1/32 gw 0.0.0.0						
SIOCADDRT: Invalid argument						
INFO: TCP_CTRL channel is sending a Control Msg(Length:64) to CE(0x400000000). @S						
TNFC: TCP CTRL channel recieve a Control Msg(Length:72) from CE(0x40000000), @Sa						
t Nov 5 16:10:24 2011						
In the Function AttriEntrySet PrefixTable, the dwNextHopID is 101						
In the Function Attrientryset_FrefixTable, the dwbtadeiD is i						
route add -net 19.1.0.255/32 gw 0.0.0.0						
SIOCADDRT: Invalid argument						
INFU: ICP_CIRL channel is sending a control Msg(Length:64) to CE(0x40000000). @S						
INFO: UDP DATA channel is sending a Redirect Msg(Length:136) to CE(0x400000000:10						
.20.0.229.3001) with sock(11). @Šat Nov 5 16:10:24 2011						
INFO: UDP_DATA channel is sending a Redirect Msg(Length:136) to CE(0x40000000:10.20.0.229.3001) with sock	11). @Sat	Nov	5 16	:10:25	2011	
INFU: UDP_DATA channel is senaing a Redirect Msg(Length:136) to CE(0x4000000010.20.0.229.3001) with Sock!	11). @Sat	Nov	5 16	-10:20	2011	
INFO: UDP DATA channel is sending a Redirect Msg(Length:136) to CE(0x40000000:10.20.0.229.3001) with sock	11). @Sat	Nov	5 16	:10:28	2011	
INFO: UDP_DATA channel is sending a Redirect Msg(Length:136) to CE(0x40000000:10.20.0.229.3001) with sock	11). @Sat	Nov	5 16	:10:29	2011	
INFO: UDP_DATA channel is sending a Redirect Msg(Length:136) to CE(0x40000000:10.20.0.229.3001) with sock	11). @Sat	Nov	5 16	:10:30	2011	-
INFO: UP DATA channel is sending a Redirect Msg(Length:136) to CE(0x40000000:10.20.0.229.3001) with sock	11). @Sat	Nov	5 16	:10:32	2011	
INFO: UDP_DATA channel is sending a Redirect Msg(Length:136) to CE(0x40000000:10.20.0.229.3001) with sock	11). @Sat	Nov	5 16	:10:33	2011	
INF0: TCP_CTRL channel recieve a Control Msg(Length:24) from CE(0x40000000). @Sat Nov 5 16:10:34 2011						
INFO: ICP_CIRC channel is sending a control $Msg(Length:24)$ to CE( $0xa0000000$ ), $Cat NOV = 5.16:10:34.2011$	11). @Sat	Nov	5 16	10:34	2011	
INFO: UDP_DATA channel is sending a Redirect Msg(Length:136) to CE(0x40000000:10.20.0.229.3001) with sock	11). @Sat	Nov	5 16	:10:35	2011	
INFO: UDP_DATA channel is sending a Redirect Msg(Length:136) to CE(0x40000000:10.20.0.229.3001) with sock	11). @Sat	Nov	5 16	:10:36	2011	
INFO: UDP_DAIA channel is sending a Redirect Msg(Length:136) to CE(UX4000000010.20.0.229.3001) with sock	11). (dSat	Nov	5 16	-10:37	2011	
INFO: UDP DATA channel is sending a Redirect Msg(Length:136) to CE(0x40000000;10.20.0.229.3001) with sock	11). @Sat	Nov	5 16	:10:39	2011	
INFO: UDP_DATA channel is sending a Redirect Msg(Length:136) to CE(0x40000000:10.20.0.229.3001) with sock	11). @Sat	Nov	5 16	:10:40	2011	
INFO: UDP_DATA channel is sending a Redirect Msg(Length:136) to CE(0x40000000:10.20.0.229.3001) with sock	11). @Sat	Nov	5 16	:10:41	2011	
INFO: OF DATA channel is sending a Redirect Hsg(Length:136) to CE(0x40000000.10.20.0.229.3001) with Sock	11). @Sat	Nov	5 16	:10:42	2011	
INFO: TCP_CTRL channel recieve a Control Msg(Length:24) from CE(0x400000000). @Sat Nov 5 16:10:44 2011						
INFO: TCP_CTRL channel is sending a Control Msg(Length:24) to CE(0x400000000). @Sat Nov 5 16:10:44 2011						
THE OUT OF DATA Channet IS Sending a Redirect Psychology (1:15) to CE(0x40000000:10.20.0.229.3001) with sock	.117. @Sat	HOV	3, 10	. 10:44	2011	~
🔝 🔚 root@localhost:~/int 🔣 eth1: Capturing - Wir 🛅 root@localhost:/hom						

Fig. 5. Feedback situation of the feedback represents.

From Fig. 5, during the multicast process, FE sent session messages in the form of UDP to respond to ACK messages periodically; the periodic time was set nearly as long as the cycle timer of the session message, but the feedback ID had a slightly difference. As a conclusion, the feedback mechanism based on the feedback representative set has a more sensitive feedback characteristic and a less feedback oscillation than those based on the worst receiver.

## 5 Conclusion

On the basis of current research status of multicast congestion control, because of the lack of the rate regulatory mechanisms within the original ForCES, this paper puts forward a kind of rate adjustment mechanism of CE based on an improved AIMD mechanism, which meets the demands of fairness within ForCES. Through the research of the heterogeneity of the receiving port, this paper also designs a congestion feedback mechanism based on the feedback representative set and this mechanism can also meet the intra protocol fairness better.

### Acknowledgments

This work was supported in part by a grant from the National Basic Research Program of China (973 Program) (No.2012CB315902), the National Natural Science

Foundation of China (No.61402408,61379120,61170215), Zhejiang Leading Team of Science and Technology Innovation (No.2011R50010-03, 2011R50010-12, 2011R50010-20), Zhejiang Provincial Key Laboratory of New Network Standards and Technologies (NNST) (No.2013E10012).

# References

- Halpern, J., Salim, J.H.: "Forwarding and Control Element Separation (ForCES) Forwarding Element Model", RFC 5812, Mar. 2010, http://www.ietf.org/html.charters/forcescharter.html
- Rizzo, L.: PGMCC: A TCP-friendly Single-rate Multicast Congestion Control Scheme[C], in Proceedings of ACM SIGCOMM, 2000.
- Fairhurst, K.: "TFMCC Protocol Behaviour in Satellite Multicast with Variable Return Path Delays" IEEE 2006.
- McCanne, S., Jacobson, V., Vetterli, M.: Receiver-driven Layered Multicast, in: Conference proceedings on applications, technologies, architectures, and protocols for computer communications, vol. 26, ACM, SIGCOMM, 1996, pp. 117–130.
- Ma, H., Meng, X., Ma, Z., Li, J.: Layered Multicast Congestion Control by Particle Swarm Optimization in Heterogeneous Environment [J]. The Journal of China Universities of Posts and Telecommunications, 2011, 18(3):85–91.
- Zhai, M., Wu, G., Gu, G.: Internet Congestion Control and Reliable Multi-point Delivery Progress [J]. Computer Research and Development, 2000, Vol. 37, No. 1.
- Bhattacharyya, S., Towsley, D., Kurose, J.: The Loss Path Multiplicity Problem in Multicast Congestion Control [C]. Proceedings of IEEE INFOCOM, New York, March 1999, vol. 2, pp. 856–863.
- Xie, F., Feng, G.\*, Siew, C.K.: Study on Nominee Selection for Multicast Congestion Control [J]. Computer Communications 29(2006) 1458–1469.

# Statistical and Alignment Based Methods for Comparison of Non-Coding DNA Sequences

Kouser and Lalitha Rangarajan\*

DoS in Computer Science, University of Mysore, Mysore, Karnataka, India. \*lalithar@compsci.uni-mysore.ac.in

Abstract. Increase in the amount of biological sequence data being generated is due to the advancement in the sequencing technologies. This has led to an upsurge in the need for methods/techniques that analyze these sequences to make biological inferences. Promoter sequences are one such biological sequences that play an important role in the process of gene regulation. These promoter sequences are made up of motifs arranged/configured in a particular order which is very important for its functionality. In this work, we propose a few techniques that capture the information regarding the arrangement of these motifs. Further this captured information help in the analysis of the promoter sequences. The methods use the motif features obtained from the Position Specific Motif Matrices (PSMMs) of the promoter sequences and then give a dissimilarity measure. Alignment-free as well as alignment based methods are proposed. The experimental results show that these methods are quite successful in bringing out the similarity/dissimilarity existing between given promoters/promoter sets. The alignment based method also helps in the process of inferring homology and in the phylogenetic analysis. In general, all the methods might help biologists in applications like drug target identification, biological pathway analysis and gene expression analysis.

**Keywords:** Alignment free, Alignment score, Multiple Sequence Alignment, Promoter sequences, Dissimilarity matrix, Sequence comparison.

# 1 Introduction

With the rapid growth in the generation of large wealth of biological sequence data, making sense of this huge data is a challenging task. There is an immediate need for good analyzing methodologies. These methods/techniques help in the process of extracting meaningful information from the sequences for better understanding of biomedical mechanisms [1] and to attempt to solve some specific biological problems. Some recent efforts are focused to identify functions of these non-coding DNA sequences which have some important function in gene expression analysis, tissue development, phylogenetic analysis which serves as the motivation to this kind of research work [2].

The analysis of these biological sequences consists of two major steps. Initial step is alignment which is later followed by analysis. Most of the existing multiple sequence alignment (MSA) works well on coding regions of the genes or the proteins.

Research in Computing Science 103 (2015)

#### Kouser, Lalitha Rangarajan

The MSA algorithms are built using the principles of dynamic programming (Smith-Waterman and Needleman-Wunsch algorithms) [3], hidden markov model [4] and scoring matrices. There are several tools available that work on these concepts to name a few SW-align, Clustal W [3], MUSCLE [5], K align [6], D align [7], T-Coffee [8], M-Coffee [9]. There is a plenty of work on alignment free sequence analysis methods such as L-words frequency [10], spaced-word frequency methods [11], relative frequencies of dual nucleotides [12] etc. Many methods relating to k-mer/word frequency, substring, information theory and graphical representation are also available. Some softwares available for the methods described above include d2Tools [13], AGP [14], Alfy [15] and WNV typer [16].

In this work, we propose methods to analyze promoter sequences which are made up of repetitive patterns called transcription factor binding sites (TFBS)/motifs. The promoter sequences are a part of the non-coding DNA anywhere before the start codon of a gene and the stop codon of the previous gene, which plays an important role in the process of gene expression. The data set is obtained from the NCBI database and the PSMM are created using the TFSEARCH tool [17].

#### 2 **Materials and Methods**

In this section, we describe the working details of our model and the proposed algorithms. Five frequency based methods proposed and one lacunarity algorithm which are all alignment free sequence comparison methods. Also, we propose one alignment based sequence analysis method which makes use of the motif pair feature. There are two datasets used, dataset 1 consisting of promoters of the Pyruvate kinase gene of different organisms and dataset 2 consisting of all the enzymes of the Glycolysis pathway of the organism human. The overall schema and flow of the method is as described in Figure 1. The detailed description of obtaining the PSMM for a promoter is presented in [18], [19]. Once the PSMM is obtained, they are given as input to the various methods briefly described below.

#### 2.1 **Frequency Based Alignment Free Methods**

The frequency distribution gives a measure considering the occurrences of motifs in promoters in various ways. The detailed description of these frequency based methods is presented [19], [20]. The output of these methods is a dissimilarity matrix.

#### 2.2 Lacunarity Based Method

The lacunarity algorithm is a multi scaled method that uses box counting across different scales to obtain the lacunarity values across different scales and then the Euclidian distance is used to get the dissimilarity between two promoters. This value then goes as an entry into the dissimilarity matrix. The algorithm can be found in detail in [18].

#### **Alignment Based Method** 2.3

This method extracts the motif pair based features from the binarized PSMM [21]. Later, we compare the count of motif pairs between the promoter sequences to find the similarity. The output of alignment is an alignment score.



Statistical and Alignment Based Methods for Comparison of Non-Coding DNA Sequences

Fig. 1. Overall schema and flow of the methods.

# **3** Results and Discussion

All the methods described are quite successful in bringing out the similarity between organisms that are known to belong to the same family. The dissimilarity is very low for similar organisms and high otherwise. The results of the frequency based method and the alignment based methods on dataset 1 are presented in Tables 1 and 2. Also, the result of lacunarity analysis is given in Table 3. Results of the frequency based method and the alignment based methods on dataset 2 are presented in Tables 4 and 5. However some cases where some unusual results were obtained are highlighted in the respective dissimilarity matrices.

Table 1. Dissimilarity matrix of frequency based method on dataset 1.(Organisms: 1. Homo sapien (human), 2. Gorilla gorilla (gorilla), 3.Macaca mulatta(rhesus monkey), 4. Bos taurus (cattle), 5.Felis catus (cat), 6.Pan troglodytes (chimpanzee)7. Canis lupus (dog), 8.Rattus norvegicus (rat), 9.Drosophila melanogaster (fruit fly) and10. Pseudomonas aeruginosa (bacterium)).

	Human	Gorilla	Monkey	Cattle	Cat	Chimpanzee	Dog	Rat	Fruit Fly	Bacteria
Human	0	299	1629	903	1411	1378	1439	451	1531	1346
Gorilla	299	0	890	256	568	581	604	266	796	587
Monkey	1629	890	0	400	500	283	488	1608	306	629
Cattle	903	256	400	0	300	213	256	820	256	369
Cat	1411	568	500	300	0	267	198	1302	480	125
Chimpanzee	1378	581	283	213	267	0	169	1263	101	298
Dog	1439	604	488	256	198	169	0	1310	262	173
Rat	451	266	1608	820	1302	1263	1310	0	1606	1275
Fruit fly	1531	796	306	256	480	101	262	1606	0	497
Bacteria	1346	587	629	369	125	298	173	1275	497	0

#### Kouser, Lalitha Rangarajan

Table 2. Alignment scores of enzyme pyruvate kinase in different organisms. (Organisms: 1. Homo sapien (human), 2. Gorilla gorilla (gorilla), 3.Macaca mulatta (rhesus monkey), 4. Bos taurus (cattle), 5. Felis catus (cat), 6. Drosophila melanogaster (fruit fly), 7. Pseudomonas aeruginosa (bacterium), 8. Pan troglodytes (chimpanzee), 9. Canis lupus (dog) and 10. Rattus norvegicus (rat)).

	Human	Gorilla	Monkey	Cattle	Cat	Fruit fly	Bacteria	Chimpanzee	Dog	Rat
Human	125	117	1	7	0	2	1	32	5	4
Gorilla	117	173	1	11	0	2	1	45	5	4
Monkey	1	1	111	10	1	0	3	0	9	8
Cattle	7	11	10	146	5	2	2	3	6	4
Cat	0	0	1	5	63	2	2	0	6	2
Fruit fly	2	2	0	2	2	201	1	1	1	7
Bacteria	1	1	3	2	2	1	67	1	1	0
Chimpanzee	32	45	0	3	0	1	1	118	3	1
Dog	5	5	9	6	6	1	1	3	142	1
Rat	4	4	8	4	2	7	0	1	1	119

Table 3. Lacunarity scores of enzyme pyruvate kinase in different organisms. (Organisms: 1. Homo sapien (human), 2. Gorilla gorilla (gorilla), 3. Macaca mulatta (rhesus monkey), 4. Bos taurus (cattle), 5.Pan troglodytes (chimpanzee), 6. Canis lupus (dog), 7.Rattus norvegicus (rat), 8.Drosophila melanogaster (fruit fly) and 9. Pseudomonas aeruginosa (bacterium)).

	Human	Gorilla	Monkey	Cattle	Chimpanzee	Dog	Rat	Fruit fly	Bacteria
Human	0	0.02	0.37	0.31	0.12	0.47	0.9	0.09	2.27
Gorilla	0.02	0	0.35	0.3	0.1	0.45	0.88	0.07	2.25
Monkey	0.37	0.35	0	0.05	0.25	0.1	0.54	0.28	1.9
Cattle	0.31	0.3	0.05	0	0.19	0.15	0.59	0.23	1.96
Chimpanzee	0.12	0.1	0.25	0.19	0	0.35	0.78	0.03	2.15
Dog	0.47	0.45	0.1	0.15	0.35	0	0.43	0.38	1.8
Rat	0.9	0.88	0.54	0.59	0.78	0.43	0	0.81	1.37
Fruit fly	0.09	0.07	0.28	0.23	0.03	0.38	0.81	0	2.18
Bacteria	2.27	2.25	1.9	1.96	2.15	1.8	1.37	2.18	0

The features obtained from the proposed alignment free methods are not very useful in the study of phylogeny. However, these methods are faster. The lacunarity features obtained at different scales individually or together seem to be more effective in phylogeny analysis when compared to the frequency based methods. They perform poor when used for gene expression differentiation. The motif pair based alignment features seem to be performing best when one is interested in performing phylogenetic analysis. The proposed alignment does not insert gaps preserving the information regarding the position of motifs, which is important in gene expression studies.

Table 4. Dissimilarity matrix of frequency based method on dataset 2. (Enzymes: 1. HK1 hexokinase 1, 2. GPI glucose-6-phosphate isomerase,
3. PFKM phosphofructokinase, 4. ALDOA aldolase A, fructose-bisphosphate,
5. TPI1 triosephosphate isomerase 1, 6. GAPDH glyceraldehyde-3-phosphate dehydrogenase,
7. PGK1 phosphoglycerate kinase 1, 8. PGAM2 phosphoglycerate mutase 2 (muscle),
9. ENO1 enolase 1, (alpha) and 10.PKM pyruvate kinase (muscle)).

	HK1	GPI	PFKM	ALDOA	TPI1	GAPDH	PGK1	PGAM2	ENO1	PKM
HK1	0	449	256	328	499	363	381	420	217	303
GPI	449	0	470	261	564	495	222	212	315	218
PFKM	256	470	0	344	474	343	348	314	151	307
ALDOA	328	261	344	0	196	223	108	138	266	111
TPI1	499	564	474	196	0	340	394	214	446	290
GAPDH	363	495	343	223	340	0	323	339	414	316
PGK1	381	222	348	108	394	323	0	128	242	86
PGAM2	420	212	314	138	214	339	128	0	196	109
ENO1	217	315	151	266	446	414	242	196	0	174
РКМ	303	218	307	111	290	316	86	109	174	0

**Table 5.** Alignment scores of enzymes in glycolysis pathway of Homo sapien.(Enzymes: 1. HK1 hexokinase 1, 2. GPI glucose-6-phosphate isomerase,

3. PFKM phosphofructokinase, 4. ALDOA aldolase A, fructose-bisphosphate,
 5. TPI1 triosephosphate isomerase 1, 6.GAPDH glyceraldehyde-3-phosphate dehydrogenase,
 7. PGK1 phosphoglycerate kinase 1, 8. PGAM2 phosphoglycerate mutase 2 (muscle),
 9. ENO1 enolase 1, (alpha) and 10. PKM pyruvate kinase (muscle)).

	HK1	GPI	PFKM	ALDOA	TPI1	GAPDH	PGK1	PGAM2	ENO1	PKM
HK1	313	13	11	9	3	10	3	1	4	2
GPI	13	173	10	8	1	9	5	5	9	2
PFKM	11	10	194	5	1	8	3	9	11	3
ALDOA	9	8	5	122	10	9	1	6	4	2
TPI1	3	1	1	10	61	11	4	5	3	0
GAPDH	10	9	8	9	11	78	3	6	4	2
PGK1	3	5	3	1	4	3	92	1	0	0
PGAM2	1	5	9	6	5	6	1	124	8	5
ENO1	4	9	11	4	3	4	0	8	223	0
РКМ	2	2	3	2	0	2	0	5	0	97

# 4 Conclusion

The results obtained from different proposed methods on the two datasets show their success in bringing out the similarity/dissimilarity existing between given promoters. The alignment based method also helps in the process of inferring homology and in the phylogenetic analysis. We notice that results are similar from both the alignment free and alignment based methods showing that both methods are equally useful. In general, all the methods might aid biologists in the process of drug target identification,

#### Kouser, Lalitha Rangarajan

biological pathway analysis and gene expression analysis. As a future work, we are contemplating on selection of subset of important motifs from the promoters. This could result in substantial reduction in PSMM size and will have direct influence on running time of algorithms. Perhaps, dissimilarity measure may also improve since less important motifs are ignored.

### References

- Qin, Y., Yalamanchili, H. K., Qin, J., Yan, B, Wang, J.: The Current Status and Challenges in Computational Analysis of Genomic Big Data. Big Data Research—Special Issue on Computation, Business, and Health Science. Vol.2(1) (2015) 12–18.
- 2. Ghiurcuta, C.G.: Models and Algorithms for Noncoding Genes. Edic Research Proposal. (2009).
- Renxiang, Y., Dong, X., Jianyi, Y., Sara, W., Yang, Z.: A comparative assessment and analysis of 20 representative sequence alignment methods for protein structure prediction. Scientific Reports 3. (2013).
- Siepel, A., Haussler, D.: Combining phylogenetic and hidden Markov models in biosequence analysis. J Comput Biol, Vol.11(2–3) (2004) 413–428.
- 5. blast.ncbi.nlm.nih.gov
- Thompson, D., Higgins, D.G., Gibson, T. J.: CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Res. Vol 22(22) (1994) 4673–4680.
- Lassmann, T., Sonnhammer, E.L.: Kalign—an accurate and fast multiple sequence alignment algorithm. BMC Bioinformatics. (2005).
- Edgar, R.C.: MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res. Vol 32(5) (2004) 1792–1797.
- Notredame, C., Higgins, D.G., Heringa, J.: T-Coffee: A novel method for fast and accurate multiple sequence alignment. J Mol Biol. Vol 302(1) (2000) 205–217.
- Soares, I., Goios, A., Amorim, A.: Sequence Comparison Alignment-Free Approach Based on Suffix Tree and L-Words Frequency. The Scientific World Journal. Vol. 2012 (2012).
- Leimeister, C.A., Boden, M., Horwege, S., Linder, S., Morgenstern, B.: Fast Alignment-Free Sequence Comparison Using Spaced-Word Frequencies. Bioinformatics. Vol.30(14) (2014) 1991–1999.
- Luo, J., Li, R., Zeng,Q.: A Novel Method for Sequence Similarity Analysis Based on the Relative Frequency of Dual Nucleotides. MATCH Commun. Math. Comput. Chem. Vol. 59 (2008) 653–659.
- 13. http://omictools.com/d2-tools-s3305.html
- Cheng, J., Cao, F., Liu, Z.: AGP: A multimethods web server for alignment-free genome phylogeny. Mol Biol Evol. (2013) 1032–1037.
- 15. Domazet, L., Haubold, M.: Alignment-free detection of local similarity among viral and bacterial genomes. Bioinformatics. Vol.27(11) (2011) 1466–1472.
- Pandurang, K., Nilesh, H., Mohan, K., Urmila, K.K.: WNV Typer: A server for genotyping of West Nile viruses using an alignment-free method based on a return time distribution. Journal of Virological Methods. Vol. 198 (2014) 41–55.
- 17. http://www.cbrc.jp/research/db/TFSEARCH.html

Statistical and Alignment Based Methods for Comparison of Non-Coding DNA Sequences

- Kouser, Lalitha, R.: Similarity Analysis of Position Specific Motif Matrices using Lacunarity for Promoter Sequences. ICONIAAC'14.Article. 37. ACM New York. NY. USA (2014).
- Kouser, Lalitha, R., Darshan, S.C., Acharya, K.K., Emin, M.A.: Alignment Free Frequency Based Distance Measures for Promoter Sequence Comparison. IWBBIO 2015. Vol.2 (2015) 183–193.
- 20. Kouser, Lalitha, R., Darshan, S.C., Acharya, K.K.: Frequency Distribution and lacunarity based Promoter Sequence Analysis (FDL-PSA). Communicated.
- Kouser, Lalitha, R.: Promoter sequence analysis through ungapped multiple sequence alignment of motif pairs. ICACCI 2015. (2015).

# **Concatenated Tabla Sound Synthesis to Help Musicians**

Uttam Kumar Roy

Dept. of Information Technology, Jadavpur University, Kolkata, India. u\_roy@it.jusl.ac.in

Abstract. *Tabla* is the prime percussion instrument used in the music of many Asian countries. This cheap instrument can produce variety of pleasant sounds and has high sense of pitch. However, it is difficult to play and consequently there is a huge lack of tabla percussionists. Majority of musicians like to perform with a few known players. Since tuning requires long time and its range is also limited, players are to carry many sets of instruments tuned at different pitch for quick switchover from one scale to another.

This paper proposes a scheme to synthesize pre-recorded *tabla* sounds to help musicians. Since recorded sound is used, the clarity and quality of the sound is unquestionable. The rhythm (taal), tempo or even pitch can be customized depending on musician's requirement. The method can also synthesize some table sounds which otherwise would be impossible to play by table percussionist.

Keywords: Sound, Signal, Tabla, Rhythm, Pitch, Tempo, Synthesis.

# 1 Introduction

To make music more promising, musical instruments often accompany soloists. One of the prime membranophone percussion instruments extensively used in classical music and traditional music of many Asian countries (such as India, Pakistan, Afghanistan, Nepal, Bangladesh, Sri Lanka, Indonesia etc.) is *Tabla* [Fig. 1]. This happens since it can create a wide variety of different pleasant sounds and rhythms, has a 'strong sense of pitch' and is also a very low-cost instrument.

Bad news is that playing *tabla* is not easy at all. It requires extensive and hard use of the fingers and palms in various configurations. This might be one of the primary reasons that there are quite a very few table percussionists in musician community. Novice singers often buy this cheap instrument expecting that some player will accompany him/her. However, they often don't find a player that makes them very uncomfortable and they often give up learning singing.

Tuning *tabla* does not only take moderately long time but also its range is limited. So, accompanists are to carry many sets of table tuned at different pitch to switch-over from one scale to another quickly.

Uttam Kumar Roy



**Fig. 1.** Percussion musical instrument 'Tabla'. Bigger one is 'dagga'(or bayan) and smaller one is 'tabla' (or dayan).

A very few attempts [Sathej & Adhikari] have been made to model this instrument. However, due to eccentric structure of *bayan* and lack of circular symmetry, all attempts have been failed to produce accurate sound.

In this paper, we have proposed a scheme to synthesize pre-recorded *tabla* sounds to accompany soloists. Since we use recorded sound, resultant synthesized sound has no difference from real performed sound. The rhythms (*taals*), pitch and tempo (*lay*) can be customized on-the-fly arbitrarily based on soloist's requirement. Some non-standard sounds and alternative musical expressions can be produced while keeping the performance expression of the traditional Tabla interaction. Using this proposed scheme, a low cost, small, portable *tabla* synthesizer can be produced that is capable of replacing a percussionist and produce table sounds with arbitrary rhythm, scale and speed to accompany musicians.

# 2 Related Work

Although a few electronic *tabla* (Radel's Taalmala Digi-60Dx and Digi-108, Sound Lab's Sangat, Pakrashi's Riyaz,) are available in Indian market, they can only produce limited number (60/100/108) of pre-synthesized rhythms. Some allow composing new but limited (only 2 to 8) rhythms. Although some of them allow limited pitch/tempo changing, they do not allow incorporating new *tabla* sounds.

The perfect model of *tabla* is not yet known. Raman [Raman, 1934] made the first scientific study of this family of drums. He and his coworkers obtained through a series of experiments, the eigen modes and eigen values of the *mridangam*. Ramakrishna and Sondhi [Ramakrishna & Sondhi] subsequently modeled the drum but agree with Raman's experimental values to within 10%. The approximate solutions were provided by [B\_S\_Ramakrishna, Sarojini & Rahman], but the agreement with experimental values is also very poor.

In [Lehana\_ Dubey], a method for the separation of *tabla* sound from a mixer of vocal and *tabla* is presented. Although the separated *tabla* sound didn't contain any residual of vocal sound, the quality of the sound was poor.

In [Kapur\_Ajay], authors describe the design of a simple electronic Tabla controller (*ETabla*). This is too a simple design and cannot even produce moderately realistic sound; hence cannot be used as professional controller.

In this paper, instead of using artificial model, we have used recorded *tabla* syllables to produce arbitrary rhythms with customizable pitch and tempo.

#### **3 Proposed Scheme**

*Tabla* manufacturers use a unique strategy to obtain harmonic overtones by loading the central part of the membrane with greater thickness (Fig. 2) resulting increased clarity of pitch and variety of tonal possibilities unique to this instrument [Sathej & Adhikari]. Further, the elements on the physics of the *tabla* can be found in [Fletcher and Rossing, 1998] or in the early work of Raman [Raman, 1934].

Playing *tabla* requires extensive use of the fingers and palms in various configurations to create a myriad of different sounds and rhythms, reflected in the mnemonic syllables (*bol*). To understand the technical part, we are describing some of the basic stokes:

#### Dagga strokes:

ge: holding wrist down and arching the fingers over the *syahi*, the middle and ring-fingers then strike the *maidan* (resonant)

ghe: similar to 'ge' except the heel of the hand is used to apply pressure or in a sliding motion on the larger drum to change pitch during the sound's decay (resonant)

ke or kath: striking with the flat palm and fingers (non-resonant)

#### Tabla stokes:

na: striking sharply with the index finger at the rim(resonant)tin: striking gently with the index finger between *syahi* and rim(non-resonant)te: striking the center of the *syahi* with the index finger (non-resonant)tun: striking the center of the *syahi* with the index finger (resonant)

### Combined strokes:

Some syllables are produced by striking both tabla and dagga simultaneously and are often called *combined syllables*. For example, 'dha' is a combination of 'na' and 'ge' where as 'dhin' is a combination of 'tin' and 'ghe'.

#### 3.1 Rhythms (Taals)

A rhythm consists of series of syllables (*bols*) of different durations  $(1, \frac{1}{2}, \frac{1}{4}$  note etc.). However, the rhythmic structure (called *taal*) can be quite complex. The basic rhythmic structures can have a large variety of beats (e.g. 6, 7, 8, 10, 12, 16,..) which are grouped in measures (called *Vivhaga*). Table 1 shows some of the popular *taals*.

In this paper, we shall use Keherwa for demonstration.

#### Uttam Kumar Roy

Name	Beats	Division	Vivhaga
Dadra	6	3+3	X 0
Kewarha	8	4+4	X 0
Tintal	16	4+4+4+4	X 2 0 3
Ektal	12	3+3+3+3	X 2 0 3
Jhaptal	10	2+3+2+3	X 2 0 3
Rupak	7	3+2+2	X 2 3

 Table 1. Some frequently used tabla rhythms.

#### 3.2 Processing Rhythms

Indian music is primarily practice-oriented. This means the rules of compositions themselves are taught from teacher to disciple, in person. Accordingly, although oral notation for *tabla* stroke names is very developed, written transcription is not standardized. Fig. 2 shows the names of basic mnemonic syllables with their striking locations.



Fig. 2. Locations for playing basic mnemonic syllables.

In fact, the set of all syllables is fairly large and requires expertise to understand them. For quick understanding, we have used only seven basic syllables; four for *tabla* 'na', 'te', 'tin', 'tun' and three for *dagga* 'ge', 'ghe', 'ke'. However, the proposed scheme works for arbitrary syllables. These syllables were recorded in lossless WAV files. The signals corresponding to these basic syllables are shown in Fig. 3.

It may be noted that signals may not have equal play duration. For example, 'tun' had largest duration (more than 2.5 sec) where as 'ke' had shortest duration (0.045 sec). This means some sound echoes longer period of time than others. To understand how to generate resultant sound that can be played behind a song, let us consider a very simple transcription:

This is the notation for popular 8-beat (*matra*) rhythms called *keherwa*. The 8 beats (written as 4/4) consist of two segments (called *bibhag*) each consisting of 4 beats and



|"ghe - te -" "na ke" "- ke" dhin|"tun - - te" "na ke" "- ke" dha|

Fig. 3. Signals corresponding to (i) tabla (ii) dagga mnemonic syllables.

is delimited by '|' character. A beat may consist of one or more syllables. If a beat has multiple syllables, they are written within ' and ' characters. A silence is represented by '-' character.

Since notation uses some characters (such as '|') other than the syllables, to divide the entire rhythm into parts, we extract only syllables before processing them further. Following shows the same using only syllables.

The timing diagram for the rhythm is shown here.

ghel-ltel-lnalkel-lkel dhin tunl-l-ltelnalkel-lkel dha

This shows that the rhythm consists of 8 beats. The first beat consists of a ghe, a silent, a te and again a silent.

#### 3.3 Separation of strikes

*Tabla* transcription is said to be *monophonic*. This means a single symbol is used even if the corresponding stroke is compound. For example, *dagga* and *tabla* strokes are combined in *dha* (na + ge), and *dhin* (tin + ghe). So, it is necessary to separate *tabla* and *dagga* transcription as follows:

33

Initialize t and d as empty sequences; for each bol in tranacription if it is a tabla bol then append it to t; if it is a dagga bol then append it to d; if it is a combined bol then

Uttam Kumar Roy

```
begin
```

```
[tb, db]=split (bol);
append tb to t;
append db to d;
```

endfor

Here is the separated transcription.

end

Tabla→ '--te-' 'na-''--' 'tin' 'tun--te' 'na -''--' 'na Dagga→'ghe---''-ke''-ke' 'ghe''----' '-ke' '-ke' 'ge'

### 3.4 Serialization

We now calculate the duration of each syllables assuming duration of a beat is unity. For example, since the first beat has four syllables, each of the syllables has duration <sup>1</sup>/<sub>4</sub>. Following shows the syllables with their durations.

Tabla	-	-	te	-	na	-	-	-	tin	tun	-	te	-	na	-	-	-	na
Tabla	1/4	1/4	1/4	1/4	1/2	1/2	1/2	1/2	1	1/4	1/4	1/4	1/4	1/2	1/2	1/2	1/2	1
	ghe	-	-	-	1	ke	1	ke	ghe	1	1	-	1	-	ke	1	ke	ge
Dagga	1/4	1/4	1/4	1/4	1/2	1/2	1/2	1/2	1	1/4	1/4	1/4	1/4	1/2	1/2	1/2	1/2	1

#### 3.5 Normalization

A notation may contain (probably many) silences. During that time, no striking happens but echo of the previous syllable continues. So, if syllable has one or more silences, the duration for the syllable has to be re-calculated. For example, the following sequence of syllables

ghe	-	-	-	-
1/4	1/4	1/4	1/4	1/2

The, duration of each syllable is shown under it. The entire sequence can be replaced by a 'ge' with duration 1.5  $(\frac{1}{4}+\frac{1}{4}+\frac{1}{4}+\frac{1}{4}+\frac{1}{2})$  units. The number of samples required for this duration is 1.5\*FS, where FS is the sampling frequency. If the signal 'ge' has at least those many samples, we take 1.5\*FS samples from the beginning. Otherwise, we append necessary zeros at the end of 'ge' signal to make total number of samples equal to 1.5\*FS. We call this process as *normalization*. Following shows the normalized duration:

	-	te		na		tin	tun	te		na			
tabla	bla 1/2 1/2 2			1	1/2 1/2			2	1	L			
ghe			ke	ke	ghe				ke	ke	ge		
dagga 3/2			1	1/2	5/2 1 1/2					1			
Time→													

Concatenated Tabla Sound Synthesis to Help Musicians

#### 3.6 Combining Components

This is where we get the resultant signal R which can be expressed as:

$$S_{result} = S_{tabla} + S_{dagga}$$

where  $S_{tabla}$  and  $S_{dagga}$  are the signals of *tabla* and *dagga* respectively. The signal  $S_{tabla}$  is obtained by concatenating all signals of *tabla bols*. The signal of a *bol* is obtained as follows:

For sampling rate  $F_s$ , and *bol* duration  $D_b$  sec, number of samples to be used is  $S_b=F_s.D_b$ . Suppose,  $L_b$  is the actual number of samples present in the *bol*. Then there are two possibilities:

### Case 1: $S_b \le L_b$

This means, *bol* has enough samples to be taken. So, we take first  $S_b$  samples from  $L_b$ .

#### Case 2: $S_b > L_b$

This means, *bol* does not have enough samples to be taken. So, first take all  $L_b$  samples and append  $(S_b-L_b)$  number of null samples (having amplitude zero). The signal  $S_{dagga}$  is obtained in a similar way. These two signals are then combined to form the resultant signal. Fig. 4(i) and Fig. 4(ii) shows *tabla* and *dagga* signals respectively whereas Fig. 4(iii) shows the resultant signal.



Fig. 4. Resultant signal (i) Tabla (ii) Dagga (iii) Combined.

#### Uttam Kumar Roy

## 3.7 Tuning Pitch

It refers to increase or decrease of sound frequencies to match with singer's voice or other instruments. Following factors should be considered:

- Two drums are tuned separately.
- The smaller drum is tuned to a specific note, usually the tonic, dominant or subdominant of the soloist's key.
- Changing the pitch must not change tempo.

Tuning must be done at frequency domain. Since input signals are in time domain, we first convert them to frequency domain, perform desired frequency shift and get it in the time domain again.

Since sound samples are uniform, we can use Fast Fourier Transform (FFT) which is an efficient implementation of Discrete Fourier Transform (DFT) to get frequency domain. Mathematically, for a set of N samples (complex numbers)  $x_0$ ,  $x_{1,...}$ ,  $x_{N-1}$ , its DFT is defined as:

$$X_{k} = \sum_{n=0}^{N-1} x_{n} e^{-i2\pi k \frac{n}{N}} \qquad k = 0, 1, 2, \dots N - 1$$
(1)

And the inverse DFT is:

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_n e^{i2\pi n \frac{k}{N}} \quad n = 0, 1, 2, \dots N - 1$$
(2)

Since vector indices start from 1 in Matlab, Equations (1) and (2) may respectively be rewritten as:

$$X_{k} = \sum_{n=1}^{N} x_{n} e^{-i2\pi(k-1)\frac{(n-1)}{N}} \qquad k = 1, 2, \dots N$$
(3)

$$x_n = \frac{1}{N} \sum_{k=1}^{N} X_k e^{i2\pi(n-1)\frac{(k-1)}{N}} \qquad n = 1, 2, \dots N$$
(4)

Let's quickly understand how MATLAB stores frequency domain data. In Equation (3),  $X_1$  is the amplitude of the DC component and  $X_k (1 \le k \le N)$ , is the complex amplitude corresponding to  $k^{th}$  frequency. The non-DC component has two parts; the first half  $X_k (2 \le k \le N/2+1)$  is the set of amplitudes for +ve frequencies and the second half  $X_k (N/2+2 \le k \le N)$  is the set of -ve frequencies. With these points in mind, we can increase the frequency as follows:

- Take the Fourier Transform
- Keep the DC component unchanged
- Shift the +ve part of the spectrum to the right
- Shift the -ve part of the spectrum to the left(or get reversed complex conjugate of the +ve part)
- · Combine DC component, shifted first and second half
- Take inverse Fourier Transform.

To decrease the frequency, the direction of the shift has to be reversed. Fig. 5 (i) shows the FFT of original syllable 'na' and (ii) and (iii) show resultant FFT after a frequency shift of +200 Hz and -200 Hz respectively.

Figure 6 shows the resultant combined signal with a pitch increase of 10 Hz.



Fig. 5. Frequency drift for syllable 'na' (i) Original (ii) +200 Hz (iii) -200 Hz.



Fig. 6. A frequency increase of 10 Hz for tabla syllable sequence.

## 3.8 Changing Speed

For speeding up, we downsample (discard some samples) and for slowing down, we upsample (add/interpolate extra samples) so that the sample rate of the track is brought back to its original rate. We defined a factor called *speed\_factor* that indicates the amount of speeding up/slowing down required. A +ve speed\_factor implies speeding up and –ve speed\_factor implies slowing down. So, the re-sampling rate may be calculated as:

new\_rate = original\_rate/speed\_factor;

37

Uttam Kumar Roy



Fig. 7. Resultant signal with (i) half speed (ii) double speed.

For example, for a sampling rate (original\_rate) 44.1 KHz, a signal having d sec duration has d\*44.1 K samples. To speed up it a factor of 2, we have taken first d\*44.1/2 = d\*22.05 K samples to be played in d sec. More specifically, the signal 'ge' has the duration 2.1 sec and has the total 2.1\*44.1=92.61 K samples. To play it in double speed, we take first 92.61/2= 46.205 K samples and play it in 2.1 seconds.

The resultant combined signal is shown in Fig. 7 (i) and (ii) with half and double speed respectively.

# Conclusion

In this paper, we proposed a concatenated synthesis pre-recorded *tabla* sounds to accompany soloists. Since we use recorded sound, it sounds like real performed one. The rhythm (*taal*), pitch and tempo (lay) can be customized arbitrarily. Nonstandard sounds and alternative musical expressions can be achieved while maintaining the performance expression of the traditional *tabla* interaction.

### References

- Fletcher, N.H., & Rossing, T.D. (1998). *The physics of musical instruments*. Second Edition. New York: Springer Verlag. [Rossing][ [Fletcher and Rossing, 1998]]
- Raman, C. (1934). The indian musical drum. In Science, P. I. A., editor, Reprinted in Musical Acoustics: selected reprints, ed. T.D. Rossing Am. Assn. Phys. Tech., College Park, MD, 1988. [Raman, 1934]
- Sathej, G., and R. Adhikari. "The eigenspectra of Indian musical drums." The Journal of the Acoustical Society of America 125.2 (2009): 831–838. [Sathej & Adhikari]
- B. S. Ramakrishna and M. M. Sondhi, "Vibrations of Indian Musical Drums Regarded as Composite Membranes", The Journal of the Acoustical Society of America 26/4, 523–529 (1954). [Ramakrishna & Sondhi]
- B. S. Ramakrishna, "Modes of Vibration of the Indian Drum Dugga or Left-Hand Thabala", The Journal of the Acoustical Society of America 29/2, 234–238 (1957). [B\_S\_Ramakrishna]
- 6. T. Sarojini and A. Rahman, "Variational Method for the Vibrations of the Indian Drums", J.Acoust.Soc.Am 30/3, 191–1996 (1958). [Sarojini & Rahman]

- Parveen Lehana, Neeraj Dubey & Maitreyee Dutta, "Separation of Tabla from Singing Voice using Percussive Feature Detection", International Journal of Computer Science & Communication, 1/1, 219–222, 2010. [Lehana Dubey]
- Kapur, Ajay, *et al.* "The electronic tabla controller." Journal of New Music Research 32.4 (2003): 351–359. [Kapur\_Ajay]
- 9. Radel Electronics Pvt. Ltd, http://www.radelindia.com/
- 10. Pakrashi and Co., http://www.pakrashi-harmonium.com/
- 11. Sound Labs, http://www.soundlabs.in/
- 12. RafAel Ferer, "Timbral Environments: An Ecological Approach to the Cognition of Timbre", Empirical Musicology Review Vol. 6, No. 2, 2011.
- Mihir Sarkar, Barry Vercoe, "Recognition and Prediction in a Network Music Performance System for Indian Percussion", Proceedings of the 2007 Conference on New Interfaces for Musical Expression (NIME07), New York, NY, USA.

# A New Approach to Adaptive Power Line Interference Removal

Abhay R. Kasetwar<sup>1</sup> and Sanjay M. Gulhane<sup>2</sup>

<sup>1</sup>Department of Electronics and Telecommunication Engineering, Bapurao Deshmukh College of Engineering, Sevagram, Wardha, Maharashtra, India. abhaykasetwar@gmail.com <sup>2</sup>Department of Electronics and Telecommunication Engineering, Jawaharlal Darda Institute of Engineering & Technology, Yavatmal, Maharashtra, India. smgulhane67@rediffmail.com

Abstract. Many studies have been devoted to the adaptive power line interference (PLI) removal design for biomedical signal processing. Most of the existing PLI removal methods are developed for applications in which the presence of PLI strongly affects the system performance. The objective of the work is to investigate the suitable method that is more feasible for its FPGA implementation. The detail study is made on available methods for PLI removal for biomedical signal processing. Different algorithms like LMS, NLMS, SLMS etc. have been analyzed for their applications in adaptive power line interference removal. LMS algorithm with some modifications in weight updation process is proposed for better performance of adaptive power line interference removal. The weights at which system got minimum error in the output are chosen using bubble sorting method to recover clean ECG signal. The implementation of adaptive PLI removal using Minimum Error Weight Adjustment (MEWA) made the system more efficient in terms of signal to noise ratio (SNR) and correlation coefficient with less computational complexity. In FPGA implementation of the system, the focus will be on making the implementation efficient in area and power utilization by using different area and power minimization techniques.

Keywords: ECG; PLI; adaptive filtering; LMS; MEWA; FPGA

# **1** Motivation and Problem Identification

An ECG signal is basically an index of the functionality of the heart. For example, a physician can detect arrhythmia by studying abnormalities in the ECG signal. Since very fine features present in an ECG signal may carry important information, it is necessary to have the signal as clean as possible. But all ECG capturing machines require ac supply due to which power line interference (50/60 Hz) is getting intro-duced in the ECG which distorts the signal badly. The fixed notch filter cannot be the solution because frequency of power line varies about fractions of a Hertz, or even a few Hertz. So we have to use the adaptive filter for the removal of such nonstationary noise. Many

### Abhay R. Kasetwar, Sanjay M. Gulhane

algorithms have been proposed for the implementation of same but could not achieve the best suited algorithm which can give the solution for frequency, amplitude and phase variations simultaneously without distorting the characteristics of original ECG signal.

So it is required to implement novel method for eliminating power line interference using adaptive filter without distorting the original biomedical signal.

# 2 Introduction

One of the application areas of signal processing techniques is biomedical engineering such as electrocardiography. The electrocardiogram (ECG) is graphical representation of cardiac activity of heart. An ECG signal is an index of functionality of heart which plays very important role in the diagnosis of heart diseases. Very fine features in ECG signal carry important information regarding patient's heart condition. So it is very necessary to have a noise free ECG for correct diagnosis. Most of the medical equipments used in hospitals are powered by power supply of working frequency 50/60 Hz. American Heart Association recommends that ECG signal recorder have 3dB frequency range from 0.67Hz to 150Hz. As power supply frequency lies within the frequency range of ECG signal, measured ECG signal is corrupted by power line interference (PLI) [2]. PLI with low frequency and weak amplitude may totally mask the signal of interest and affects the reliability and accuracy of ECG signal. For the quality analysis of cardiac diseases, PLI should be removed from ECG signal, while keeping ECG signal intact [4]. Traditional approach to remove PLI was to use a notch filter with narrow rejection bandwidth. Increase in attenuation level of notch filter, PLI can be removed effectively however increased bandwidth disturbs the nearby spectrum. This is the major drawback of using fixed notch filter for removal of PLI from ECG signal [2]. This approach is not suitable when PLI is non-stationary in nature, i.e. when frequency, phase and amplitude of PLI are varying with respect to time. So adaptive interference canceller is required for the removal of such kind of interference. Adaptive filters have the capability of tracking and detecting dynamic variations and time varying potentials. The purpose of an adaptive interference canceller is to subtract non-stationary type of noise from a received signal in adaptively controlled manner so as to improve the signal to noise ratio (SNR). In adaptive PLI cancellation method, PLI is estimated by minimizing difference using adaptive algorithms. Further estimated PLI is subtracted from noisy ECG signal to cancel the PLI at successive iteration [7]. The adaptive interference canceller shown in Fig. 1 operates on the reference PLI input i(n) to produce estimate of the noise which is then subtracted from the desired signal d(n), i.e. noisy signal to get the error signal e(n). Further the weight updation is done by MEWA LMS algorithm to minimize the error.

Power line interference may occur through two mechanisms: capacitive and inductive coupling. In capacitive coupling, the transfer of energy between two circuits by means of a coupling capacitance pre-sent between two circuits. The value of the coupling capacitance decreases with increasing separation of the circuits while inductive coupling causes because of mutual inductance between two conductors. The geometry of the conductors as well as the separation between them determines the value of the A New Approach to Adaptive Power Line Interference Removal



Fig. 1. Adaptive Interference Canceller.

mutual inductance, and hence the degree of inductive coupling. Actually, capacitive coupling is responsible for high frequency interference while inductive coupling causes low frequency interference. For this reason, inductive coupling is the dominant mechanism of power line interference in electrocardiography. Ensuring the proper application of electrodes, that there are no loose wires, and all components have adequate shielding which can help to minimize occurrence of amount of power line interference.

# **3** Related Work and Preliminaries

Author presents a survey on implementation of an adaptive power line interference canceller for ECG signal processing and recommends to use Least mean-square (LMS) algorithm. Notch filters are ineffective, whenever the power line frequency is not stable or not accurately known. A mismatch between the suppression band and the power line frequency might lead to inadequate reduction of power line interference therefore adaptive interference cancellers should be preferred. LMS algorithm is commonly used for adaptive filtering since it is computationally simple and efficient [1]. The least mean square (LMS) is the most used algorithm to iteratively minimizing the mean square error (MSE) of the system output. In some practical applications, the LMS algorithm can be implemented only with delayed coefficient adaptation [11]. The use of delayed coefficient adaptation in the LMS algorithm has enabled the design of modular systolic architectures [12]. The convergence behavior of this delayed least mean squares (DLMS) algorithm, when compared with that of the standard LMS algorithm, its performance is degraded with the increase in the adaptation delay. Modular design systolic architecture for transversal adaptive filtering that maintains the convergence behavior of the LMS algorithm by minimizing the adaptation delay, and also supports high input sampling rates with minimal input/output latency [15]. The first used adaptive method to remove 50/60 Hz interference from a corrupted ECG signal using adaptive filter was proposed by Widrow. This method is very robust and has the capability to remove 50/60 Hz interference [8]. Ziarani et al. proposed nonlinear adaptive filter for removal of PLI from ECG signal. Structure of filter is highly simple and required little arithmetic [5].

Adaptive noise canceller (ANC) method with internal reference signal is introduced by Ziarani which is having more complex structure. Martenes *et al.* proposed simple adaptive noise canceller as improvement in Widrow method and neglecting the presence of PLI harmonics [6]. Ziarani proposed very practical method for PLI removal but

43

### Abhay R. Kasetwar, Sanjay M. Gulhane

adaptation constant may not lead to a successful equation phase. The im-proved adaptive canceller (IAC) is proposed by Martens *et al.* with important improvement such as error filtering and adaption blocking based on QRS complex detection [7].

Further the work is extended by Bharath with the use of appropriate window length. Yue Der Lin and Yu Hen Hu proposed LDA based adaptive filter overcome drawback of unrealistic assumptions for existing methods with less computational complexity. Structure is capable to eliminate PLI with variable frequency and other sinusoidal interference [3]. Some researchers have presented the real time imple-mentation of an adaptive canceller using FPGA. Larger the value of step size parameter (u) increases the convergence speed but affects the stability and vice versa [14],[17].

# 4 Proposed Method

A block diagram of the proposed adaptive PLI removal using Minimum Error Weight Adjustment (MEWA) is depicted in Figure 1. This proposed adaptive canceller is capable to suppress the nonstationary PLI in ECG recordings. Weights of filter are automatically updated using MEWA algorithm. This algorithm is highly efficient because of its robustness and low computational complexity. We have used the sliding window approach for filtering. As the order of filter is 2, the window length is also taken as 2. So the computational complexity is getting reduced which will be a very important parameter for real time implementation. The weights at which the system got minimum error in the output are chosen to recover the clean ECG signal. The first order digital FIR filter is used for filtering purpose. Both the filter weights are getting updated at every iteration. The performance of system is analyzed by observing the value of error for different values of step size parameter and number of filter taps. The bubble sorting method has been used to find out the location at which minimum value of error is present. That location identifies the most appropriate weights of filter to reconstruct the required ECG signal. Those weights are maintained till the next minimum value of error gets appear. So the method is called as Minimum Error Weight Minimization (MEWA) Method. Sign Least Mean Square (SLMS) algorithm is used for updating the weights. Different PLI signals are artificially generated for testing purpose using inductive coupling mechanism. Mathematically, MEWA can be demonstrated as follows:

$$e(n) = d(n) - y(n) \tag{1}$$

where d(n)=x(n)+i(n)

x(n) is a clean ECG signal and i(n) is added interference

 $W = lam^*W + mu^*sign(e(n))^*X$ ; where e(n) is an error signal (2)

$$Location(l) = n\{min(e(n))\}$$
(3)

$$w = w(l);$$
 where w is the current weight vector (4)

A New Approach to Adaptive Power Line Interference Removal



Fig. 2. Clean ECG signal from MITBIH database.



Fig. 3(a). Noisy ECG Signal with 5 db SNRin.



Fig. 3(b). Noisy ECG Signal with 0 db SNRin.



Fig. 3(c). Noisy ECG Signal with -5 db SNRin.

### 4.1 Evaluation Method

In order to validate the performance of MEWA LMS algorithm, simulation is carried out using ECG signals from MITBIH database as shown in Fig. 2. The ECG signals are of 1 s duration with a sampling frequency of 250 Hz. First the signal is passed

45

### Abhay R. Kasetwar, Sanjay M. Gulhane

through a channel to introduce channel noise. Further the received ECG is added with the additive noise to form a noisy signal as shown in Fig. 3(a)(b)(c). This signal is used as an input signal to evaluate the performance. To quantify the performance of the interference canceller, signal to noise ratio (SNR) and the correlation coefficient (CC) between the original ECG signal and the filtered ECG signal are calculated. Signal to noise ratio is measured at the input, as SNRin and at the output as SNRout of the adaptive canceller. SNR improvement is one of the most important parameter to evaluate the system performance. SNRin is defined as the ratio of the power of the ECG signal to the power of added noise and SNRout is defined as the ratio of the power of the ECG signal to the residual noise power. And Correlation Coefficient gives the degree of similarity between clean ECG and filtered ECG.

# 4.2 Results and Discussion

Clean ECG signal recorded at Beth Israel Hospital (BIH) in Boston and made available by Massachusetts Institute of Technology (MIT) is taken for simulation. The clean ECG signal is having 2560 discrete values as shown in Fig. 2. This clean ECG signal is passed through noisy channel and further polluted by synthetically generated nonstationary interference as shown in Fig. 3 (a)(b)&(c). The algorithm is analyzed by using MATLAB simulation. In order to compare the performance of proposed adaptive algorithm, first the performance of normal LMS algorithm is taken into the consideration. Thereafter proposed MEWA LMS algorithm is analyzed for different values of step size parameter. Following Table 1 shows the comparative analysis of normal LMS algorithm with MEWA LMS algorithm with respect to SNRout and Corr. Coeff values.

Fig. 4 (a)(b)(c) shows the simulation results of MEWA LMS algorithm for improved values of SNRout and Correlation Coefficient (CC) when u=0.01 and No. of taps = 2. In each Fig. plot 1 shows the overlapping of noisy and recovered noisy signal where plot 2 shows the overlapping of clean ECG and and recovered ECG signal.

No.of taps (N)	SNRin (db)	SNRout(db)		Corre. Coeff.(CC)	
		Normal LMS (u=0.000001)	MEWA LMS (u=0.01)	Normal LMS	MEWA LMS
2	-10	1.34	32.83	0.6535	0.9833
2	-5	7.82	37.2	0.6457	0.9816
2	0	13.39	41.82	0.6695	0.9832
2	5	18.62	44.04	0.8044	0.9815

Table 1. SNRout and Corr. Coeff. Values for Normal LMS and MEWA LMS.

A New Approach to Adaptive Power Line Interference Removal



Fig. 4(a). Output of MEWA LMS for SNRin = -5 db.



**Fig. 4(b).** Output of MEWA LMS for SNRin = 0 db.



Fig. 4(c). Output of MEWA LMS for SNRin = 5 db.

47

Abhay R. Kasetwar, Sanjay M. Gulhane

### 5 Conclusion

Cancellation of Power Line Interference from ECG signal is a challenging problem as the frequency of Power Line Interference may change with time. This work suggested the minimum error weight adjustment (MEWA) mechanism for LMS algorithm. An adaptive interference canceller using MEWA LMS efficiently filters the ECG signal without distorting the characteristics of ECG signal with fewer computations. The choice of step size parameter is again a critical issue. Result shows the improved values of SNRout and Correlation coefficient. As the filter requires only two taps for designing, the computational complexity is getting reduced. This will help the designer for real time implementation. The proposed algorithm gives SNR improvement of about 40db. The comparative analysis proves the advantage of MEWA LMS over Normal LMS.

# 6 Work in Progress

Currently the work is going on for finding the best suitable algorithm for adaptive power line interference removal. The feasibility of algorithm for real time implementation will be checked first. Once the algorithm is getting finalized, real time implementation of the algorithm will be done. Adaptive interference cancellers can be implemented using general-purpose microcontrollers or digital signal processors. These solutions are not optimum since they have low processing speed and sequential execution due to the fact that adaptive cancellers are implemented with a software program. Another solution is the implementation of the adaptive controller with field-programmable gate array (FPGA) [10][19].

The adaptive filters can be implemented by sequential, parallel and semi-parallel architectures. The type of architecture chosen is based on sample rate and number of coefficients. The parallel architecture is well suited for a high sampling rate requirements and a small number of coefficients. However, the sequential architecture is more suited for the low sampling rate requirements and a large number of coefficients. The semi-parallel architecture is a good compromise that permits to implement filters having a large number of coefficients and requiring a high sampling rate [9]. So the adaptive power line interference removal will be implemented using semi-parallel architecture on FPGA.

After real time implementation on FPGA, the next target is to calculate the requirement of power and area for chip level implementation probably on 90nm technology. Quartus and Cadence tool will be used for detail power and area analysis.

## References

- Abhay R. Kasetwar and Sanjay M. Gulhane : "Adaptive Power Line Interference Can-celler A Survey" in *International Journal of Advances in Engineering & Technology*, Jan. 2013 ISSN: 2231–1963.
- 2. Maryam Butt, Nauman Razzaq, Ismail Sadiq.: "Power Line Interference removal from ECG Signal using SSRLS Algorithm" 9<sup>th</sup> International Colloquium on Signal Processing and Applications, 2013.

A New Approach to Adaptive Power Line Interference Removal

- 3. Yue-Der Lin, Yu Hen Hu: "Power Line Interference Detection and Suppression in ECG Signal Processing," *IEEE Transactions on Biomedical Engineering, Vol.55, No.1 January2008.*
- 4. H. N. Bharat, K. M. M. Prabhu: "A New LMS based Adaptive Interference Canceller for ECG Power Line Removal," *International Conference on Biomedical Engineering February*, 2012.
- Alireza K. Ziarani, Adalbert Konrad: "A Nonlinear Adaptive Method of Elimination of Power Line Interference in ECG Signals," *IEEE Transactions on Biomedical Engineering*, Vol. 49, No. 6 June 2002.
- Martens S.M.M., Bergmans J.W.M. and Oei S.G.: "A Simple adaptive interference cancellation method for power line reduction in electrocardiography", proceeding of 25<sup>th</sup> Symposium on Information Theory in the Benelux, pp. 49–56 (2004).
- Martens S.M.M., Mischi M., Oei S.G, and Bergmans J.W.M.: "An improved adaptive power line interference canceller for electrocardiography", *IEEE Transactions on Biomedical Engineering*, Vol.53, pp. 2220–2231 (2006).
- 8. B. Widrow et al :"Adaptive noise cancelling : principles and application." *Proceeding of the IEEE*, Vol. 63, no. 12, pp.1692–1716. Dec. 1975.
- Abhay R. Kasetwar and Dr. Sanjay M. Gulhane: "A Survey of FPGA Based Interference Cancellation Architectures for Biomedical Signals" in *International Conference on Computer Communication and Informatics (ICCCI-2013)*, Jan. 04–06, 2013, Coimbatore, INDIA 978-1-4673-2907-1/13/\$31.00©2013IEEE.
- Mohammed Bahoura, et al.: "FPGA-Implementation of a Sequential Adaptive Noise Canceller using Xilinx System Generator," in *International Conference on Microelectronics*, 2009.
- G. Long, F. Ling, and J. Proakis: "LMS algorithm with delayed coefficient adaptation," *IEEE Transaction on Acoustic, Speech & Signal Processing*, vol. 37, no. 9, pp. 1397–1405, 1989.
- 12. Dai Huhe, *et al.*: "A Novel Suppression Algorithm of Power Line Interference in ECG Signal," in *First International Conference on Pervasive Computing, Signal Processing and Applications*, 2010.
- 13. B. S. Lin *et al.*: "Removing residual power-line interference using WHT adaptive filter," in *Proceedings of Second Joint EMBS / BMES Conference*, USA, Oct 23 26, 2002.
- LD Avendano-Valencia, *et al.*: "Improvement of an Extended Kalman Filter Power Line Interference Suppressor for ECG Signals," in *Journal of Computer in Cardiology*, 2007, vol. 34, pp. 553 - 556G.
- Ramanathan S, *et al.*: "A systolic architecture for LMS adaptive filtering with minimal adaption delay," in *Proceedings of 9<sup>th</sup> International Conference on VLSI Design*, Jan. 1996, pp 286 – 289.
- Zhiguo Zhou, et al.: "FPGA-Implementation of LMS Adaptive Noise Canceller for ECG Signal Using Model Based Design," in *IEEE International Symposium on Bioelectronics* and Bioinformatics, Nov 3–5, 2011.
- 17. N. Jindapetch, S. Chewae, P. Phukpattaranont: "FPGA implementation of an ADALINE adaptive filter for power line noise cancellation in surface electromyography signals" in *ELSEVIER measurement*, 45 (2012) 405–414.
- 18. G. Chen : "The FPGA Implementation of Kalman Filter," in 5th WSEAS nternational Conference on Signal Processing, Computational Geometry & Artificial Vision, 2005.
- R. Ramos *et al.*, "Application by means of FPGA of an adaptive canceller 50 Hz interference in electrocardiography," in *IEEE Instrumentation and Measurement Technology Conference*, Hungary, May 21–23, 2001.

# UML Behavioral Refactoring for the Specification of Complex Software Systems

M.T. Chitra<sup>1,2</sup> and Sherly Elizabeth<sup>2</sup>

<sup>1</sup>University of Kerala <sup>2</sup>Indian Institute of Information Technology and Management - Kerala {chitra,sherly}@iiitmk.ac.in

**Abstract.** Behavioral models play a prominent role in specifying software systems by facilitating the abstract behavior views and analyzing the elementary aspects of a system. The sequence diagram, one of the key behavioral diagrams in UML, provides intuitive ways to capture requirements and scenarios as a sequence of events. The paper proposes a generic framework for effective code generation from UML models. The proposed framework acts as an interexchange format that helps to combine the structural and behavioral constraints of the system objects associated thereby facilitating consistent source code generation. Model Refactoring contributes to improve the software quality and productivity thus mitigating flaws in the system during the design phase itself. The case study presents the refactoring of online simulation of a mill optimization problem in a thermal power plant system.

Keywords: Behavioral diagrams, Model Refactoring.

### 1 Introduction

The increasing complexity of the software systems demands the advent of novel effective development approaches that can overcome the major drawbacks of the existing development technologies. Model Driven Engineering (MDE), fostered by Object Management Group (OMG) helps reducing the development costs of complex software systems through the use of technologies like Model Driven Architecture (MDA) that supports rigorous analysis of software models [2, 3]. In MDA, models are the primary entities for entitling the system at different levels of abstraction. Models can help to detect inconsistencies or incompleteness in requirement model by evaluating the simulation of the scenarios.

The Unified Modeling Language (UML) is one of the prevalent languages used for modeling complex safety-intensive software systems. It provides a collection of modeling notations for design specifications to build models that describe the different views on a system during various stages such as requirements analysis and design artifacts of software systems. UML Sequence diagrams are now

51

becoming familiar to represent the behavioral specifications of the complex systems. Sequence diagrams emphasize on modeling the interactions between the collaborating objects participating in the interactions as a time-ordered set of messages [1, 3].

However, UML diagrams fail to specify the semantics in representing the entire information to determine the complete system behavior. The lack of formal semantic specification for UML makes it difficult to analyze the consistency notion in these diagrams. Hence, several transformation approaches are being adopted for analyzing and specifying the consistency of behavioral models which transform the UML models into some semantic domain where the consistency constraints can be represented and validated. The formal specification languages like OCL, TOCL, Z, etc. helps to add additional information to the diagram thereby ensuring the completeness of the model.

OCL is a declarative, side-effect free, formal specification language used along with UML diagrams for specifying the object constraints and queries on the UML models. It precisely defines the well-formedness rules for UML as well as the OMG-related metamodels. OCL is mainly used to specify the invariants of objects as well as the pre and post conditions of the operations [6].

Representing the dynamic behavior of complex time-safety critical systems is a rigorous task as well as a tough research problem to be taken care of. It involves the careful analysis of the semantic aspects with respect to the system constraints. Moreover, there are no model-based techniques or tools available so far for analyzing such temporal properties in UML behavioral diagrams, especially in sequence diagrams. Existing approaches use model transformation techniques that transform the UML models to some other language that supports automated analysis, which are also complex and erroneous.

The Temporal OCL (TOCL) finds its significance in this context. The time and safety related constraints as well as the behavioral specifications, which are hard to express using OCL can be specified using TOCL, a temporal logic extension of OCL. In this paper, we evaluate the impact of OCL and TOCL along with UML behavioral diagrams to completely represent the system behavior especially in safety-critical software environments.

Towards this goal, we provide a refactoring approach to embed the static as well as the temporal constraints involved in the system behavior through a development chain. It constitutes the generation of the behavioral design model using UML sequence diagrams, specification of the constraints using OCL and TOCL, generation of code from the behavioral models by applying the refactoring approach and the execution and analysis of generated code. This approach helps in predicting the system behavior at modeling level itself by comparing the simulation results with real time results. This consequently allows the scientists or the researchers to work exclusively at modeling level in order to obtain the optimized system models. They can analyze the scenario specifications directly from the execution model and can add changes accordingly.

The work focuses on interaction modeling, showcasing the dynamic aspects of the interaction between the participating objects. The dynamic modeling

### UML Behavioral Refactoring for the Specification of Complex Software Systems

includes visual specification of the system functionalities in detail, where the functionalities are realized through the message passing between the objects participating in the interaction. The paper discusses about the development of a framework which shows the automated refactoring of UML interaction diagrams, especially the sequence diagram designs, to improve the understandability and maintainability of the design for the efficient source code generation process.

This paper is an extended work and its main new contribution is that the framework has been enhanced with the potential of including a set of safety and temporal constraints in addition to the static constraints of the system [16]. The work proposes a novel approach that yields an interexchange framework to include the complex system behaviors into UML Sequence diagram design as constraints, thereby enriching the model elements with the necessary details of the system without affecting its external behavior.

Incompleteness in the generated source code may often arise due to the absence or failure in representing all possible information regarding the objects that participate in that system. The constraint specification languages play an intelligent role here, by removing this inconsistency in the model design which in turn reflects in the completeness of the generated source code. The proposed framework paves the way for improving the software code quality and productivity fulfilling all the specified system requirements during the code generation process.

The main contributions of this work can be summarized as follows:

- The behavioral refactoring approach in sequence diagram designs is proposed, which paves a way to incorporate static, temporal and safety related constraints into the design and derive a refactored interexchange model from the existing SD specification.
- To provide a generic behavioral pattern for implementing the proposed refactoring methodology in UML models that provides a space for consistent code generation of software systems.

The remainder of this paper is structured as follows: Section 2 gives the related works and identifies and explains the different refactoring activities. Section 3 describes the proposed refactoring approach. Section 4 presents the implementation of the approach through a case study discussion. Finally, Section 5 shows the results and discussions part of the case study and Section 6 concludes.

### 1.1 Background

The software refactoring is an emerging area where a lot of researches are being carried out on exploring the ways to address refactoring in a consistent manner. Refactoring revolutionizes the design by applying some effective process for improving code quality. The term refactoring was instigated in 1992 by William F. Opdyke in his research work in the context of object-oriented software to support software evolution and reuse. He defined refactoring as "behavior preserving program restructurings or transformations containing particular preconditions that

must be verified before the transformation can be applied in order to make the design of a program clearer and to make it easier to add new features" [7, 8].

Refactorings can also be applied to reduce or eliminate redundant parts of program codes [7]. Martin Fowler defined the process of refactoring as "a change made to the internal structure of software to make it easier to understand and cheaper to modify without changing its observable behavior". He proposed the refactoring catalog which focuses on manual refactoring, demonstrated with examples regarding the principles of refactoring, the useful ways to identify and find the associated low-level refactoring(s) that helps fixing a code problem in a controlled and efficient manner [9].

Mens *et al.*, in their review on model-driven refactoring discussed on the latest approaches in model refactoring and also the various challenges encountered while applying refactoring on model level [13]. Mohamed *et al.* discussed on their the existing model refactoring approaces based on feature based modeldriven taxonomy [14]. Most of the investigations focus on UML class diagrams, for applying model refactorings [10]. There are only a few approaches that focus refactoring behavioral diagrams and prove their behavior preservation properties in a standard way. G. Sunye *et al.* were the first to present a set of refactoring(s), particularly on UML class models and state machine models, and explained how they can be modeled so as to preserve the behavior of a UML model. They showed that refactorings can be defined for UML in such a way that their behaviorpreserving properties are protected, based on OCL constraints defined at the metamodeling level [11].

Alessandro Folli and Tom Mens used Algebraic Graph Grammar tool to support refactoring on the UML designs. They focus on class models and state machine models and define a metamodel similar to the UML meta model as a type graph. The limitation of this work is that this type graph can represent only the simplest version of the UML metamodel [17]. France *et al.* described a meta modeling approach to pattern-based model refactoring in which refactoring(s) are used to introduce a new design pattern instance to the model [18].

Several formalisms have been suggested to examine model refactoring(s). Most of these propose exhibiting model refactoring in a declarative way. Graph transformation theory was used in several works for describing model refactoring and formal properties have been used to review these refactoring(s) [15].

Most of the refactoring approaches focused on code-level refactoring and only a limited works address the model refactoring approaches and especially with class diagrams [13]. But while taking care of the behavior preservation of design models, we need to rely on behavioral models and the constraints associated with them. Only a few works are available in the literature with respect to refactoring of behavioral models. None of the existing approaches can be used to verify operation-based model refactoring that involves changes to operation specification. The core idea presented in this paper focuses on facilitating the integration of system behavioral properties throughout the time point at different abstraction levels one wants to guarantee for the safe or normal execution of the system. UML Behavioral Refactoring for the Specification of Complex Software Systems

# 2 The Conceptual Approach

### 2.1 Refactoring the Behavioral Specifications

Model transformation is the process of modifying the source model to produce the target model. Refactoring is a model transformation approach which restructures the system by altering its internal behavior without affecting the external behavior [1, 2]. Modifying the UML model by enriching with the necessary constraints will significantly increase the quality of the design as well as the generated source code from it. The process of refactoring enables the model to adapt to future extensions.

By definition, refactoring should ensure behavior-preserving transformations of an application. It means that the external behavior of the model before and after the refactoring must remain the same. The major problem faced by designers is to measure the actual impact of modifications on the various design views, as well as on the implementation code which serves as a valid proof to prove the correctness of system and hence showing the behavior preservation property of refactoring approach. Another crucial task here is identifying or determining the exact element to which the refactoring has to be applied.

We are applying the refactoring approach in UML sequence diagrams, which primarily shows the behavioral interactions between the objects participating in the system. Such behavioral transformations need extra care as their change raises some difficulties.

In the proposed work, the model refactoring is applied to the lifeline model element of the sequence diagram design for ensuring the correctness of the system. Since the primary focus of the sequence diagram is to specify the interaction between the collaborating objects, represented as lifelines, it has been chosen as the focal point for refactoring.

We also need to justify why the behavior preservation condition holds for these model transformations:

- Adding the constraint schema into the lifeline model element does not make any modification to the behavior of that element. Rather, we are only giving space to group all the data related to that model element into a common point in order to improve the understandability.

The overall architecture of the refactoring process proposed in this work is shown in Figure 1.

The XML Metadata Interchange (XMI) is OMG standard interchange format used in UML models for exchanging metadata information via XML. It includes information about elements in a model and their relationships. The transformation of the models into XMI is indeed a breakthrough in facilitating interoperability across various tools and platforms [5]. The XMI generated would then be fed into an XMI parser to retrieve the meta model information which inturn is used for the generation of source code. Besides, OCL and TemporalOCL also supplements more precise information specification in the UML behavioral model. The work is an attempt to allow the specification of inter-object scenarios for object-oriented system models in a succinct expressive manner.

M.T. Chitra, Sherly Elizabeth



Fig. 1. Architecture of the Refactoring process

#### $\mathbf{2.2}$ Generation of the OCL-TOCL Framework

The work focuses on developing X-CodeFrame, an XML code framework for representing OCL and TOCL constraints into UML behavioral models, especially the Sequence diagram model thereby facilitating automatic behavioral code generation. This approach helps in simplifying the transformation of the static, temporal and safety-related constraints of the system, thereby smoothening the code generation process.

The constraints that are typically hard to express in OCL, which entitles the temporal and safety related issues of the system are specified using the temporal OCL (TOCL). The static as well as the temporal constraints specified for a system using OCL and TOCL are parsed for framing the constrained elements in order to generate the framework as per the defined schema [schema No-Fig.]. The extracted constraints are then set appropriately into the OCL-TOCL frameworks by matching the context  $\langle classname \rangle$  attribute.

An XML Schema Definition (XSD) typically called an XML schema, is the de facto standard for describing XML documents, controlled by the World Wide Web Consortium (W3C). The XML schema formally defines the XML document structure accompanying the rules for data content and semantics. The OCL schema proposed here defines an XSD file that provides the design pattern of how the constraints will set within the XMI file obtained after the refactoring process.

The OCL-TOCL schema is the backbone of the proposed UML refactoring process. Based on the defined schema rules, the constraints are embedded into the XMI file of the UML model thereby refactoring it for quality source code generation. The basic types that we are considering for the schema framework generation are the invariants, methods, pre-condition and post-condition operations.

The schema defines the specification rules for framing the constraints appropriately associated with each of the objects participating in the system interaction. For each class defined in the constraint files (OCL/TOCL) there will be an < extendedConstraints > tag generated which is the root element for the constraint construction framework. All the constraints that belong to a particular class will come to an individual < extendedConstraints > tag created for that class. It includes the class name details using the attribute context and also stores the file type within the type attribute.

### Fig. 2. Extended constraints

The schema template structure for adding invariants into the OCL-TOCL framework is as follows. The < MethodDetails > tag is the main tag element inside the < extendedConstraints > tag which represents the whole constraint block within a particular class based on the nature of the constraints. The < body > tag element sets the constraints within the < MethodDetails > appropriately. The < constraints > tag is the complex type tag set that helps in identifying all the constraint expressions, operations etc in the < MethodDetails > and include these entities as individual elements under the < constraint > tag within it. The schema template structure for generation of an operation or method is as shown in Fig. 3.

For the execution of a method in any complex system generally includes a precondition which must be considered for a safe system functioning, a body part which includes the actual functionality to be performed and a postcondition operation in which the status of the generated output is checked for further processing of the system. Hence the syntax for generation of the method schema also include these as the key factors while framing data with respect to a method. The < precondition > tag element helps in representing all those conditions for checking the pre-conditions that should satisfy or perform if any before performing the actual method body. The < body > tag include the ex-

57

```
<extendedConstraints context="class_name1" type="ocl/tocl">
  <MethodDetails name="inv_name1 " returntype=" " type="inv">
    <body>
       <constraints>
         <constraint>
           <condition exprn="expression1"/>
           <condition exprn="expression2"/>
             .....
         </constraint>
       </constraints>
    </body>
  </MethodDetails>
  <MethodDetails name="inv_name2 " returntype=" " type="inv">
    <body>
       <constraints>
         <constraint>
          <condition exprn="expression1"/>
          <condition exprn="expression2"/>
         </constraint>
      </constraints>
    </body>
  </MethodDetails>
      .
</extendedConstraints>
```

Fig. 3. Schema template structure for generation of an operation or method

pression and operations that have to be taken place within that method and the < postcondition > tag represents all those constraints that must have to be executed after the method execution.

The generation of java class files corresponding to the OCL schema is done using JAXB (Java Architecture for Xml Binding), which directly binds the XSD/XML element to particular fields of java classes and vice versa using the properties marshalling and unmarshalling.

The OCL files containing the static constraints and the TOCL files which contain the temporal and safety related constraints for the system are parsed. The constraints are parsed and extracted in a way that helps in fixing the context to which the respective specification refers to in the generated OCL-TOCL frameworks. The constraints extracted are set with the associated java objects of the java classes created using JAXB.

The generated OCL TOCL frameworks are then injected into the XMI of the Sequence diagram model based on the lifelines associated with it.

The OCL -TOCL framework generation process is shown as Algorithm 1.

Algorithm 2 explains the process of filling the constraint objects with appropriate constraint body tag elements.

Algorithm 1. X-CodeFrame Generation

<b>Require:</b> OCL file for sd's (OCLsd) and also TOCL file for sd's (TOCLsd), if any.					
Ensure: XML framework of the constraints w.r.t. different contexts referred.					
1: begin					
2: Initialize the ocl-tocl parser with the input					
3: Read OCL / TOCL file					
4: for each line in file do do					
segregate the contents against the context					
: for each new content in context do do					
7: create an object of ExtendedContraints with type OCL/TOCL					
8: store the object against context					
9: parse the context string					
10: extract the details context data elements like class name, constraint					
type, constraint name, return type if any					
11: end for					
12: for each new constraint within a context do					
13: get the ExtendedConstraints object against a context					
14: add an object of MethodDetails under ExtendedConstraints					
15: invoke populateConstraintBody()					
16: end for					
17: iterate the list of ExtendedConstraints of all context					
18: marshall the objects to the XML content					
19: store it against the context					
20: end for					

### 2.3 Applying Refactoring to the Model: An Algorithmic approach

Identification and extraction of the exact information that represent functionalities or behavior from the UML models are crucial. This work is proposing a automated refactoring approach for the scenario based UML designs, especially, the UML sequence diagrams, to model the complex system behaviors, henceforth building the behavioral scenarios for individual subcomponents of the system. Refactoring supports a highly dynamic software lifecycle by improving the internal structure of a piece of code block without altering its external behavior. The operation specifications which cannot be directly included in the UML designs are expressed using the static as well as the temporal constraint languages.

Given a model M which consists of model elements which are associated with it to perform the behavioral aspects as per the system requirement specification based on the metamodelling standards (IM). Applying model refactoring in the model MR = (pre, TR), where pre is the precondition or set of rules that model must preserve and satisfy, and TR is the model transformation that is applied to the model.

The Algorithm 3 describes the whole process taken place during the UML refactoring approach.

We illustrate how the refactoring process will happen through the simulation of the mathematical modeling process for optimizing the time factor in a pulverizing process in a thermal power plant system.

59

Algorithm 2. The populateConstraintBody()

**Require:** ExtendedConstraints object for a *context* Ensure: Modified ExtendedConstraints object with constraint body 1: begin 2: if Precondition then create an object of ExtendedConstraints.MethodDetails.Preconditions 3: create Constraints object within Preconditions 4: set the expression string to the Precondition constraints 5: 6: else if Postcondition then 7: create an object of ExtendedConstraints.MethodDetails.Preconditions 8: create Constraints object within Preconditions 9: set the expression string to the Precondition constraints 10: else if Body or inv then create an object of ExtendedConstraints.MethodDetails.Preconditions 11. 12:create Constraints object within Preconditions 13:set the expression string to the Precondition constraints 14: end if 15: return the modified object of ExtendedConstraints 16: end

Algorithm 3. The Automated UML Refactoring Process

**Require:** ExtendedConstraints object for a *context* 

Ensure: Modified ExtendedConstraints object with constraint body

1: begin

2: if Precondition then

- 3: create an object of ExtendedConstraints.MethodDetails.Preconditions
- 4: create Constraints object within Preconditions
- 5: set the expression string to the Precondition constraints
- 6: else if Postcondition then
- 7: create an object of ExtendedConstraints.MethodDetails.Preconditions
- 8: create Constraints object within Preconditions
- 9: set the expression string to the Precondition constraints
- 10: else if Body or inv then
- 11: create an object of ExtendedConstraints.MethodDetails.Preconditions
- 12: create Constraints object within Preconditions
- 13: set the expression string to the Precondition constraints
- 14: **end if**
- 15: return the modified object of ExtendedConstraints
- 16: end

# 3 Case Study : The Coal Pulveriser Optimization Problem

We demonstrate the capability of the proposed refactoring approach though the online coal pulverizing mill optimization problem [16]. In this system, we have considered both the static as well as the behavioral properties involved in the pulverizing process to demonstrate the applicability of the approach. The

### UML Behavioral Refactoring for the Specification of Complex Software Systems

main function of the pulverizing mill is to grind and dry the moisturized raw coal supplied to it from the coal storages. The two main classes involved in the pulverisation process in a coal mill are the CoalStorage and the Pulveriser.

The proposed system behavior is modeled using the UML sequence diagram. The safety and time related constraints as well as the static constraints are expressed as TOCL and OCL files respectively. The model is exported as an XMI file and is parsed appropriately for extracting the relevant tag data elements for the code generation process. The OCL as well as the TOCL files are simultaneously parsed to extract the relevant details into the corresponding tag elements in the XMI file using the proposed X-CodeFrame framework.

The UML together with the OCL and TOCL helps in representing the facts that belong to the behavioral level completeness of the system. The constraints of the pulveriser which cannot be typified visually are represented using OCL and TOCL files. The pre and post conditions to be satisfied and the invariants of the system model specified using the OCL file are embedded to the design model by the applying the refactoring approach proposed in the work in order to accomplish the model consistency. The pulveriser optimization process simulation is performed by transforming the mathematical system model using the Object Constraint Language. Along with the sequence diagram model information the temporal as well as the static constraints related with the CoalStorage and the Pulveriser sub systems are also supplied to the model in order to enrich the design data for the code generation process.

### 3.1 Simulating the Online Coal Pulverisation Process Optimization

The following mathematical model explains the coal pulverisation process [19, 20]. The model is converted to discrete time form for the purpose of online implementation. Figure 2 illustrates the overall online pulverising mill optimization process. The mill model variables are monitored dynamically in real time. The unknown parameters in the equations are estimated using evolutionary computation technique (Genetic Algorithms) and system simulation techniques based on the on-site measurement data. The normal pulverisation process is described mathematically using the following equations:

$$W_{air}(t) = 10 \sqrt{\triangle P_{pa}(t) \frac{273}{273 + T_{in}(t)} \frac{28.2}{22.4}}$$
(1)

$$W_c(t) = K_{fs} * F_s(t) \tag{2}$$

$$W_{pf}(t) = K_{16} * \triangle P_{pa}(t) * M_{pf}(t)$$
 (3)

$$M_c(t) = W_c(t) - K_{15} * M_c(t)$$
(4)

$$M_{nf}(t) = K_{15} * M_c(t) - W_{nf}(t)$$
(5)

$$P(t) = K_6 * M_{pf}(t) + K_7(t) * M_c(t) + K_8$$
(6)

M.T. Chitra, Sherly Elizabeth



Fig. 4. Online mill performance simulation

$$\Delta P_{mill}(t) = K_9 * \Delta P_{pa}(t) + \Delta P_{mpd}(t) \tag{7}$$

$$\Delta P_{mpd}(t) = K_{11} * M_{pf}(t) + K_{12} * M_c(t) -K_{13} * \Delta P_{mpd}(t)$$
(8)

$$T_{out}(t) = [K_1 * T_{in}(t) + K_2] * W_{air}(t) - K_3 * W_c(t) - [K_4 * T_{out}(t) + K_5][W_{air}(t) + W_c(t)]$$
(9)

The main input variables supplied to the pulveriser system include raw coal flow into the pulveriser, primary air differential pressure and primary air inlet temperature. The output variables include pulveriser differential pressure, outlet temperature and mil current. The online coal mill optimization process is specified as a constraint file as shown below. During the refactoring process these equations are affixed under the < body > tag elements section of the Pulverising() method call.

The values of the constant co-efficients are obtained using the Genetic Algorithm. The code generation helps the researchers and experienced engineers in analyzing and comparing the simulation results of the online mill model with the real plant data set values and thereby improving the mill performance.

### 3.2 OCL Constraints in the Coal Pulverising Process

This section presents the approach by specifying the temporal properties associated with the pulverisation process in a coal-fired thermal power plant system. The structural constraints that must necessarily hold true or checked during the pulverization process are represented as invariants using the OCL file.

```
Mc = ((Wc - (k15 * self.Mc)) * T )
Mpf = (((k15 * self.Mc) - (self.Wpf)) * T)
DPmpd = (((k11 * self.Mpf + (k12 * self.Mc) - (k13 * self.DPmpd))) * T)
```

UML Behavioral Refactoring for the Specification of Complex Software Systems

The following are a few constraints associated with the two classes CoalStorage and PowerPlant in the thermal power plant system:

```
context CoalStorage inv cosize:CoalSize=20
context CoalStorage inv hval: HGI=55
```

The invariants that must hold true for the CoalStorage class are:

```
context CoalStorage inv cosize:CoalSize=20
context CoalStorage inv hval: HGI=55
```

The typical invariants that must hold for the Pulveriser class are as shown below.

```
context Pulveriser inv: Wc $<=$ 45</pre>
context Pulveriser inv: Wair $<=$ 75</pre>
context Pulveriser inv: Tin $<=$ 300</pre>
context Pulveriser inv: DPpa $<=$ 180</pre>
context Pulveriser inv: outlet temperature $<=$100
context Pulveriser inv: DPmill $<=$ 500
context Pulveriser inv: P $<=$ 60</pre>
context Pulveriser inv: if a.Stage = 3 then
 Speed=54 else Speed=52 endif
context Pulveriser inv:if a.Stage = 3
then self.Type = 'HP803PXBowl'
 else self.Type = 'XRP763BowlRoller'endif
context Pulveriser inv:if CoalHGI = 55 and CoalMoisture = 0.1
  and CoalFineness = 0.7 then
  if a.Stage = 3 then self.Capacity2 = 39.9 else
       self.Capacity2 =33.8
 endif
  else self.Capacity2 $ <>$ 0
endif
```

### 3.3 Safety and Temporal Constraints in the Coal Pulverizing Process

The temporal constraints involved in the coal milling process include the safety and time related aspects of all the classes or objects participating in that process. For the **CoalStorage** class the main temporal constraints involved are:

1. Failure in the level of the minimum storage level or maximum storage level occurred; that means the coal level exceeds the limit of the maximum storage level or has reached the minimum level.

For the **Pulverizer** class the main temporal constraints to be considered are:

- 1. The system is in initialization mode until all the physical sub components or units inside the pulveriser are in ready mode and all the external attributes supplied to the system satisfie the normal range values for proper functioning of it or a failure in the level of coal quantity supplied to the pulverizing mill has been identified.
- 2. The system is in the normal mode, which is the standard operating mode when the program tries to maintain the raw coal level in the pulveriser between the level of values with which all physical units are operating correctly.

The below listed are a few safety and temporal constraints associated with the coal pulverisation process.

When the Pulveriser system is in the initialization mode, it remains in this mode until all physical units are ready or a failure in the pulverising mill has occurred.

context Pulveriser inv: self.mode = # Initialization implies always self.mode = # Initialization until (Pulveriser Unit.allInstances -> forAll(u: Pulveriser Unit | u.ready))

Instantly the program recognizes a failure in the Coal Mill system until it goes into the rescue mode.

context Pulveriser inv: self.coalmillFailure implies next self.mode = # Rescue

Failure of any of the physical or measured units except the coal mill puts the program in to degrade mode.

context Pulveriser inv: (if Wc > 45 or HGI! = 55 or Wair > 75 or Tin<=300 or DPpa > 180 or outlet\_temp > 100 or DPmill > 500 or P > 60) implies next self.mode = # Degraded

When the Coal Mill system is in the initialization mode and a failure of the coal mill is detected, it puts the program into emergency stop.

-- Wc<=45 context Pulveriser inv: (self.mode = # initialization and self.WcFailure) implies next self.mode = # EmergencyStop

context Pulveriser::openMillValve()
pre: self.valve.mode = # Off
post: self.valve.mode = # On

--Coal Storage constraints context CoalStorage::getCoalLevel(): Double pre: self.program.mode = # Normal post: self.coalLevel = result

UML Behavioral Refactoring for the Specification of Complex Software Systems

## 4 **Results and Discussions**

The UML SD design model for the coal pulverization process is shown in Figure 5:



Fig. 5. UML Sequence diagrams

The OCL framework generated for the OCL constraints given below is as shown in Figure 6. Here the CoalStorage class has two constraints associated with it, which are converted into two < MethodDetails > elements under the < ExtendedConstraints > object tag.

The OCL-TOCL framework generated for the mill optimization process is as follows.

The framework generated for the preconditions and postconditions specified as OCL constraints that must satisfy for the Pulveriser object is:

The code framework generated for invariants of the Pulveriser subsystem is:

# 5 Conclusion

This paper concentrates on providing a generic framework for refactoring the specification of complex systems modeled using UML2.0 sequence diagrams. It focuses on combining the structural and the behavioral constraints, thereby offering a path for consistent and quality source code generation. The system has been formally modeled using the OCL/TOCL language to provide explicit and precise system information to the design. A generic template framework has been built based on the constraints as well as the UML sequence metadata of

```
<extendedConstraints context="CoalStorage" type="ocl" >
  <MethodDetails name="hval" returntype="boolean" type="inv">
    <body>
       <constraints>
         <constraint>
           <condition exprn="HGI=55"/>
         </constraint>
       </constraints>
    </body>
  </MethodDetails>
  <MethodDetails name="cosize" returntype="boolean" type="inv">
     <body>
       <constraints>
         <constraint>
           <condition exprn="CoalSize=20"/>
         </constraint>
       </constraints>
     </body>
  </MethodDetails>
</extendedConstraints>
```

Fig. 6. The OCL framework generated for the OCL constraints

```
<body>
    constraints>
<constraint>
     self.Tout">
<condition exprn="P = (k6 * self.Mpf) + (k7 * self.Mc) + k8"/>
condition exprn="DPmill = k9 * self.DPpa"/>
<condition exprn="Wpf = k16 * self.DPpa + self.Mpf"/>
  </constraint>
/constraints>
iody>
```



the system by using refactoring approach. The proposed method facilitates the mathematical verification of pulveriser system in a thermal power plant. The representation of extra information as static as well as temporal constraints attached to certain locations of the objects lifelines in the sequence diagram allows the identification of gaps and contradictory specifications during the source code generation process.

#### Acknowledgments 6

The authors would like to thank the Control and Instrumentation Group, CDAC, Thiruvanathapuram for providing the raw plant data of pulverising mill which is used for modeling and validating the system. This work is supported by SPEED-IT programme of Kerala State IT-Mission under Govt.of Kerala.

UML Behavioral Refactoring for the Specification of Complex Software Systems

Fig. 8. The preconditions and postconditions specified as OCL constraints that must satisfy for the Pulveriser object

```
<MethodDetails name="inv_21" type="inv" returntype="boolean">
<body>
 <constraints> <constraint>
  <condition exprn="DPpa&lt;=180"/>
 </constraint> </constraints>
</body>
</MethodDetails>
<MethodDetails name="inv_22" type="inv"
returntype="boolean">
<body>
 <constraints> <constraint>
  <condition exprn="outlet_temp&lt;=100"/>
  </constraint> </constraints>
 </body>
</MethodDetails>
<MethodDetails name="inv_23" type="inv" returntype="boolean">
<body>
 <constraints> <constraint>
  <condition exprn="DPmill&lt;=500"/>
 </constraint> </constraints>
</body>
</MethodDetails>
<MethodDetails name="inv_24" type="inv" returntype="boolean">
<body>
 <constraints>
  <constraint>
   <condition exprn="P&lt;=60"/>
  </constraint>
 </constraints>
 </body>
</MethodDetails>
```

Fig. 9. The code framework generated for invariants of the Pulveriser subsystem

### References

- 1. Unified Modelling Language, http://www.uml.org/.
- Object Management OMG. Unified Modeling Language Specification 2.0: Infrastructure. Technical Report ptc/03-09-15, OMG (2003).
- G. Booch, J. Rambaugh, and I. Jacobson, The Unified Modeling Language User Guide, Addison-Wesley (1999).
- Model Driven Architecture, White Paper, Object Management Group OMG, May (2014).
- 5. XML Metadata Interchange, www.omg.org/spec/XMI/
- OMG, Object Constraint Language (OCL) Specification, version 2.0,2006 http://www.omg.org/spec/OCL/2.0/
- Opdyke, W.F.: Refactoring: A Program Restructuring Aid in Designing Object-Oriented Application Frameworks, PhD thesis. Univ. of Illinois (1992).
- Roberts, D.: Practical Analysis for Refactoring. PhD thesis. Univ. of Illinois (1999).
- 9. Fowler, Martin: Refactoring: Improving The Design of Existing Code. Pearson Education India (1999).
- Astels, Dave: Refactoring with UML. In: 3rd International Conference on eXtreme Programming and Flexible Processes in Software Engineering, 67– 70 (2002).
- Suny, G., Pollet, D., Le Traon, Y., Jzquel, J. M.: Refactoring UML models. In International Conference on Unified Modeling Language UML -2001. LNCS 2185. Springer Berlin Heidelberg, 134–148 (2001).
- Maddeh, M., Romdhani, M., Ghdira, K.: Classification of Model Refactoring Approaches. Journal of Object Technology, 8.6, 143–13958 (2009)
- Mens, Tom, Tourw, Tom: A survey of software refactoring. Software Engineering, IEEE Transactions on 30.2, 126–139 (2004).
- Mens, Tom, Taentzer, Gabriele, Mller, D.: Model-driven Software Refactoring. Model-Driven Software Development: Integrating Quality Assurance, 170–203 (2008).
- Mens, Tom, Taentzer, Gabriele, Mller, Dirk.: Challenges in model refactoring. In: 1st Workshop on Refactoring Tools, University of Berlin. Vol. 98 (2007).
- Chitra M. T., Elizabeth Sherly.: Refactoring sequence diagrams for code generation in UML models. In IEEE Int. Conf. on Advances in Computing, Communications and Informatics ICACCI, 2014, 208–212 (2014).
- 17. Folli, Alessandro, Mens, Tom.: Refactoring of UML models using AGG. Electronic Communications of the EASST (2008).
- France, R., Chosh, S., Song, E., Kim, D. K.: A metamodeling approach to pattern-based model refactoring. Software, IEEE, 20(5), 52–58 (2003).
- Y. G. Zhang, Q. H. Wu, J. Wang, G. Oluwande D. Matts, and X.Zhou.: Coal Mill Modeling by Machine Learning Based on Onsite Measurements, Energy Convers. IEEE Trans., 17(4), 549–555 (2002).
- J. Wei, J. Wang and Q. H. Wu.: Development of a Multisegment Coal Mill Model Using an Evolutionary Computation Technique, IEEE Trans. Energy Convers., 22(3), 718–727 (2007).

# **Electronic System for Orientation Control and Obstacle Avoidance for Underwater Glider without a Rudder**

Yuvraj N. Patil<sup>1</sup>, V. S. Aneesh<sup>2</sup>, M. T. Abhilash<sup>3</sup>, and Swati Kadlag<sup>4</sup>

<sup>1</sup>Student, Dept. of Electronics and Telecommunication Engineering, Symbiosis Institute of Technology (a constituent of SIU), Pune – 412115, India. yuvraj.n.patil22@gmail.com
<sup>2</sup>Technical officer, Defence Avionics Research Establishment (DARE), DRDO-Bangalore, India. aneeessh@rediffmail.com
<sup>3</sup> Prof., Dept. of Electronics, DRDO-DIAT(DU), Pune, India. chipmaker2005@gmail.com
<sup>4</sup>Prof., Dept. of Electronics, Symbiosis Institute of Technology (a constituent of SIU), Pune – 412115, India. swatik@sitpune.edu.in

Abstract. Autonomous underwater gliders are classified in buoyancy operated long endurance underwater vehicles. The glide control algorithm has been developing for many designs. A system is proposed as an improvement to the existing designs to achieve glide control with 3 chamber internal hydraulic system and to avoid obstacles in its saw tooth motion profile and the orientation control that can be achieved with injection of fluids in internal tanks instead of relying on rudder or movable internal mass. Two algorithms for obstacle avoidance and correcting its approach vectors i.e. a self correcting mechanism with sensors and servo motors is represented. The onboard embedded system developed with payload sensors, navigation, communication and actuator control of the prototype design is also discussed in general.

**Keywords:** Gyroscope; Ultrasonic; compass; Obstacle; Servo; Prototype; Compensation; Orientation.

# 1 Introduction

An autonomous underwater glider goes in a saw tooth motion with some changes in its buoyancy. Such vehicles are small, low power and long endurance and are used for many applications [1]. The hydrodynamics of existing underwater gliders have been deployed and tested in various environments [2]. However there can be more techniques than those used for controlling the motion of a glider. There exists a scope for smart techniques like obstacle avoidance, change in use of rudders and movable internal mass to control the steering of the glider etc.

69

### Yuvraj N. Patil, V. S. Aneesh, M. T. Abhilash, Swati Kadlag

In this paper, we are proposing a new approach to achieve a better propagation of the glider with help of orientation sensors. The results obtained by the existing gliders show a random motion of the glider into the ocean and collection of the data at those points. The ocean can pose various unwanted disturbances and obstacles. The gliders are likely to be deployed in coastal regions or low depth regions where obstacles like coral reefs, underwater mountains and marine life can cause threat to the glider [3]. To avoid this, glider must be able to detect the obstacle and avoid it by modifying its path of propagation. For this purpose, sensors like ultrasonic sensors, gyroscopes and accelerometers are integrated with system. The steering of the glider would become more difficult and complex if the rudder encounters malfunctions or if there is no rudder at all. For such situation, there's an alternative system for steering with movable internal mass mechanism. This system can be altered by utilizing the water around the glider to change its orientation. By letting in a controlled amount of water in chambers can provide necessary tilt and the vehicle can glide in that direction as a result of upward buoyant force and the resultant angle of the wings.

A system on chip approach is selected as the main control unit of the platform because of its flexibility of design, higher performance in terms of speed, computing power, better resource management than conventional microcontrollers, low cost and low power implementations. The current design uses PSoC5LP platform for the development of the control system. Obstacle avoidance and orientation control algorithms have been developed to adapt to the changes in environmental conditions of the glider. Some of the orientation sensors and navigation system have been interfaced according to the needs of the project. The glider by NASLab of Michigan Technical university [14] is based on single buoyancy tank. We are proposing a 3 cylinder approach. A detailed description of the interface and algorithm has been specified for the prototype design.



Fig. 1. Prototype of Underwater Glider designed by: Naval Constructors' Wing (NCW), IIT Delhi.

# 2 Previous Work

Initial models of some of the underwater gliders were Slocum battery and thermal, spray, sea glider etc. All these gliders use different methods to vary the buoyancy for its gliding motion in a saw tooth pattern. Each dive cycle consists of diving to a certain depth and rising to the surface. Almost all existing gliders depend on hydraulic type buoyancy engines powered by batteries. The Slocum thermal is an exception to these gliders and relies on

thermalocline of ocean to convert solid medium to liquid and again from liquid to solid state [4]. It results in its displacement and helps to propagate in desired direction. However, this method restricts its functionality to areas where thermocline is well developed.

The Seaglider has a TT8 microcontroller as its electronic platform which uses Motorola MC68332 M.C. with 12-bit ADC and power management circuits [5]. It basically has conductivity temperature depth (CTC) sensing components on board. The Seaglider relies on a satellite communication link to transmit onboard data to the mission computer [6]. Whereas the Spray is a TT8 controlled glider that utilizes subsystems to control attitude and buoyancy variations, data acquisition, storage and transmission over satellite channel [7]. Along with CTDs, the spray also has flurometers, altimeter, Iridium satellite communication system, turbidity and dissolved oxygen measurement system.

The author [8] described selection of piston operated tanks for the prototype. The basic algorithm of the buoyancy engine was clockwise and anti-clockwise motion of a DC motor achieved that simulated to and forth motion of the piston in tanks that would result in a saw tooth profile. The onboard sensors included a temperature and a pressure sensor. The data from these sensors was stored onto a SD card and transmitted using XBee s2b Pro wireless Zigbee module after each dive cycle. However, the system was neither mounted on glider body nor was tested in actual or virtual environments.

# **3** Present Prototype Design

The microcontroller platform for the onboard control system remains to be PSoC5LP from Cypress. The data storage is in SD card as done previously and the trans-receiver module is still XBee s2b Pro. However some modifications to the existing design have been made as follows.

# **3.1 Buoyancy Engine**

In previous design, a DC motor simulated the action of a piston operated buoyancy engine. The design approach was for a single cylinder mount for the periodic charging and discharging to result into a saw tooth motion. However the motor did not drive the piston mechanism and just simulated the change in rotation periodically. Also the design did not take into account the steering or the orientation control system of the glider. This gave rise to following objectives for a modified design.

- Build a piston operated cylinder structure for testing the algorithms.
- Couple the DC motor to a rack and pinion mechanism to drive a piston of a syringe cylinder.
- Modify the engine to a 3 cylinder design to provide orientation control
- Interface servo motors to operate pistons precisely

Unlike the spray that had a movable internal mass mechanism for steering [9], the design would make use of water from outside the glider body for similar operation. This system is also suitable for scenarios where rudder is not present or has encountered a malfunction.

Yuvraj N. Patil, V. S. Aneesh, M. T. Abhilash, Swati Kadlag



Fig. 2 & 3. Top view of the design of demonstration set up, Front view of the design showing position of servo motors and a DC motor with respect to cylinders.



Fig. 4 & 5. Side view of the servo controlled cylinder. Servo motor is connected to a rack and pinion assembly, Side view of the DC motor controlled cylinder responsible for saw tooth profile of the glider motion.



Fig. 6 & 7. Buoyancy engine for the prototype design showing location of the cylinders inside the body of the glider, Glider turning right as one side tank becomes heavier by letting in more water than the other tank.

Figures 2, 3, 4 and 5 show the schematics of the proposed three tanks of the glider. However this placement is not tested on actual prototype glider body for the functionality neither is the position of those cylinders fixed as shown in above figures. The control system block consists of motors and rack and pinions that drive the pistons in those cylinders.

The position of the tanks inside the body is shown in Figure 6 and orientation change with respect to initial position is shown in Figure 7. The modeling parameters for the mechanical system are skipped in this paper and more emphasis is given on developing an algorithm for their operation.
Electronic System for Orientation Control and Obstacle Avoidance for Underwater Glider ...

#### 3.2 Onboard Sensors and Drivers

The board has now been interfaced with orientation sensors which include a 3 axis analog accelerometer (ADXL335), a digital gyroscope (MPU6050) and a digital compass (HMC5883L). The payload includes a BMP085 pressure and temperature sensor and a GPS module. Two HCSR04 ultrasonic sensors are interfaced for obstacle avoidance. The PSoC5 at the receiver end has been replaced by a XBee programmer module connected directly to PC via USB cable.

The gyroscope, BMP085, and digital compass have i2c interface with PSoC5LP where as the ultrasonic sensors are PWM operated. These sensors are basic robotics sensors and might not be suitable for underwater applications. These sensors have similar interfaces and are used to develop the algorithms. The paper includes making use of the data acquired from these sensors for developing the control algorithm. The DC motor is interfaced via LM369 motor driver IC as shown.

#### 3.3 Navigation System and Link with Base Station

As the glider surfaces during its saw tooth evolution [1], the base station needs to acquire a fix on the current location of the glider. Based on important parameters such as high sensitivity of 165dBm, 22mA 3.3V low power consumption is made. UART interface, NMEA format Output, the GPS module SKG13C from SkyLab is selected for the application. Though the module is accurate up to 3 meters, it is cheap and user friendly. Also the module is low power and gives complete information data with position, velocity and time information in serial NMEA protocol. The module is suitable for most of the portable devices.

The module works on input voltage between 3.0V to 4.2 V and on minimum current of 100mA. The receiver supports active antenna or passive antenna with maximum gain of 25dB. The module has on full duplex serial channel UART. This port is connected to the UART of PSoC5LP to receive data frames continuously that are being acquired form available satellites. However the acquisition is not continued. The data reception only takes place when the glider is on surface and the reception gets cut off when the glider goes inside the water. The GPS module requires 36 sec. for cold start and 1 sec. for hot start. This time depends upon the time for which the glider stays inside the water. The frames are in NMEA format and consist of GGA-Global Positioning System Fixed Data, GSA-GNSS DOP and Active Satellites, GSVGNSS Satellites in View and RMC-Recommended Minimum Specific GNSS Data required information which can be extracted and recorded.

For the communication with base station, XBee Pro s2b trans-receiver is selected. The module operates in ISM band (2.4GHz) and has maximum data rate of 250kbps. The Pro series modules have a longer range compared to other modules which is 2 miles of outdoor coverage in line of sight. And 200m in indoor scenarios with transmission power of 63mW. The module is low power and operates in voltage range between 2.6V to 3.3V.

#### 3.4 Embedded System

The embedded system required for this application is supposed to be low power, low cost and versatile in nature. The low power operation of the microcontroller gives the

73



Yuvraj N. Patil, V. S. Aneesh, M. T. Abhilash, Swati Kadlag

Fig. 8. Onboard interfaces with PSoC5LP development board.

glider its long endurance while its versatile nature supports various sensor and actuator interfaces which can vary in future as well [3]. Hence for this application a system on chip (SoC) approach is selected. SoCs are low power, Re-programmable, resource saving powerful computing devices [6]. PSoC5LP has 256KB flash memory, 64 KB of SRAM, and SD card interface for data storage accompanied by a 12-bit Delta sigma A/D converter and a 20-bit Successive Approximation Rate (SAR) A/D converter.

It has over 64 GPIOs (General Purpose Input Output) which can be configured as analog or digital input/outputs. It also supports I2C and SPI protocols, which are required by most digital output sensors. Figure 8 shows interfacing on transmitter side and receiver side.

On the receiver side of the system, the XBee module is mounted on a programmer board from Sparkfun electronics and is directly interfaced with mission computer via USB cable. The data reception on the PC is done with help of Realterm software [15] that captures the incoming data on serial ports.

#### **Obstacle Avoidance Algorithm** 4.

#### 4.1 Ultrasonic Sensors

Since sound has proved to travel farther in water without much fading than radio waves, sonar has been an effective technology in detecting objects at farther distances without an actual contact with main body. For interface purpose and developing an algorithm two HC-SR04 ultrasonic sensors are interfaced with PSoC5LP. The sensors can detect any solid obstacle up to 400 centimeters. The reason behind using two such sensors is that the glider should not consider a small fish or some junk sinking in the water as an obstacle. By comparing both sensors the PSoC5LP can determine whether or not the obstacle is large enough to change motion of the glider, these sensors con detect obstacle from 0 degrees to 15 degrees. So they can be placed on both wings or in the nose of the glider to cover required area.

#### 4.2 Interfacig

Two HC-SR04 ultrasonic sensors are interfaced with PSoC5LP and placed at a distance of 3 cm from each other for demonstration purpose.



Fig. 9. Two ultrasonic sensors interfaced with PSoC5LP.

On the actual glider, they can be placed on both wings to cover more area and better obstacle detection. The sensors are PWM type and needs a trigger of 10 micro seconds after which they send out 8 pulses of 40kHz and then weight for an echo. A counter runs for that amount of time till the echo pulse is received. Both the sensors give almost same distance with offset of 1 cm. The distance is calculated based on following equations.

$$Range = \frac{echotime \times velocity of sor}{2}$$
(1)

There will be some noise present in the received echo but can be easily dealt by applying certain threshold. Complex filters are not required since there purpose of these sensors is just to detect an obstacle and not to map an entire terrain.

The algorithm is for obstacle avoidance applied for the central buoyancy tank since this is the tank that would be responsible for the saw tooth motion of the glider. As the water is being drawn into the tank, the glider is going down. If the obstacle is detected, the piston motion is reversed so as to push the water out as a result the glider starts upward motion before reaching its predefined depth to avoid the obstacle. The AUG central piston and tank is imitated by a 100ml surgical syringe operated by rack and pinion mounted on a wooden platform. Since this is a prototype used for simulation purpose and developing an algorithm, the dimensions and design parameters that would suit actual glider are neglected at this moment. The displacement of the rack is given by following formula:

$$x = 2 \prod r \Theta \tag{2}$$

Yuvraj N. Patil, V. S. Aneesh, M. T. Abhilash, Swati Kadlag

where,

x is the displacement of rack,

r is the radius of pinion,

 $\Theta$  is the angle of rotation of pinion.



Fig. 10 & 11. DSO showing echo received by sensor after triggering, Desired obstacle avoidance in saw toot motion.

The DC motor is turned clockwise for a duration of 5 seconds which is enough to pull the piston by full length of the cylinder. A short delay is of 2 sec. is introduced after which the motor goes in anti-clockwise direction. This pushes the piston back in its initial state depicting the surfacing action of the glider. The delay depends upon the counter being incremented per iterations which selected based on experimental observations.

A feedback is taken from ultrasonic sensors all the time to check for any obstacle. In case of an obstacle, the rotation of the motor is reversed.



Fig. 12. Flowchart for obstacle avoidance system.

Research in Computing Science 103 (2015)

76

This stops the glider from achieving predefined altitude and the glider will change its upward or downward motion. The glider does not steer itself in order to avoid obstacle hence the steering mechanism is independent of the obstacle avoidance system. Above is a flowchart of the algorithm for the obstacle avoidance system.

### 5. Orientation Control Algorithm

Many of the existing gliders that are designed rely on either rudder or movable internal mass to change its course. Assuming unavailability of the rudder on a glider and movable solid internal mass, a 3 cylinder mechanism is proposed to be used. The pistons of side cylinders are operated by two VS-2 servo motors. The motor shaft is connected to rack and pinion mechanisms that drive the pistons. The servo motors are PWM signal driven and their angle shift for respective compare values is shown in following diagram. The system clock for the PWM block in PSoC creator 3.0 is set at 2MHz. and the period is kept 20 micro seconds. The angle will depend upon the compare value which is nothing but the duty cycle of PWM pulse. The pulse width is varied in order to change the duty cycle.



**Fig. 13 & 14.** PWM for minimum angle of servo motor for piston at initial position (no water in tank), PWM for maximum angle of servo motor for piston at final position (water in tank).

At 1.53ms of the pulse width, the angle of rotation of the servo motor is close to 0 degrees. On powering on the system, this is the default PWM pulse width given to the servo motor. At 2.40ms of the pulse width the angle of rotation of the servo motor is close to +90 degrees. The on time for pulse tends to increase further on more deviation but would not result in servo motor's angle change beyond +90 degrees.

For understanding the concept behind the steering control, the angle of rotation of servo motor is kept directly proportional to the change in the difference of yaw angle from glider's initial approach vector. i.e. minimum the angle of deviation from main path lesser will the water allowed to fill up the tank and vice versa. This would result

77





Fig. 15. Flowchart for orientation control using feedback from sensors.

in steering of the glider and compensating motion for unwanted shift in the course of the glider A feedback is taken from digital compass and the gyroscope. The compare value of PWM required to drive servo motor by -90 degrees is 600 and for +90 degrees is 2400. The equations for mapping the difference in yaw angle with the compare value are.

$$map(difference, \min, \max, 600, 2400)$$

$$Compare \ value = \left(\frac{(difference \ in \ yaw + 90)}{180}\right) \times (2400 - 600) + 600$$

$$Compare \ value = [(difference \ in \ yaw + 90)]10 + 600 \tag{3}$$

This compare value is nothing but the pulse width that needs to be varied for the PWM signal to get desired angle on servo motor. The flowchart of the algorithm is shown in Figure 15.

The digital compass is a HMC5883L with i2c interface working on 2.6V-3.6V. The sensitivity of this digital compass is 2 milli-gauss Field Resolution in ±8 Gauss Fields. As the digital compass may read any value while dropping the glider, depending upon the direction at that moment the reading is recorded and stored in a temporary variable. This variable is stored as a reference for further computing. The PSoC keeps reading its new heading direction and compares its current reading with this stored value. The current reading is subtracted from the stored value and the difference is noted. Depending upon positive or negative value of the difference, the deviation in left or right direction is determined. Same is done by using a digital gyroscope. The difference in the angle is calculated and compared with the digital compass readings.

The algorithm is implemented on piston mechanisms mounted on a wooden platform. Figure 18 shows actual demonstration board with cylinders mounted with rack and pinions driven by servo motors and DC motor.



Electronic System for Orientation Control and Obstacle Avoidance for Underwater Glider ...

Fig. 16. Actuator control with feedback from sensors.



Fig. 17. Prototype set up of actuator control with PSoC5LP interface.

### 6 Conclusion

The orientation control and obstacle avoidance is achieved on a demonstration set up based on system of chip (SoC) platform with the help of sensors. The SoC approach provided better flexibility in programming and interfacing all the required sensors with minimum system resources. PSoC5LP powered by ARM-Cortex M3 has better performance than PSoC3 in terms of CPU and RAM utilization for a given program. This remains an initial stage of development of the prototype that is intended. The actuator selection for implementation is conceptual and intended to develop an algorithm. Also the mechanical parameters needed for selection are neglected at this stage. Currently the piston motion obtained by servo and DC motor based on sensor inputs is satisfactory. However the tuning of this system is significantly crude and needs precision.

The obstacle detection carried out by the ultrasonic sensors is based on a threshold set for a minimum range of 20cm. which can be changed based on the region of operation, velocity, length and weight of the glider. The intension of this system remains to detect the obstacle and not to map the terrain but can be considered for better accuracy. Similar algorithm can be implemented for underwater acoustic sensors as well. The motion of the piston in central cylinder is reversed by DC motor on an obstacle being placed before the sensors to simulate change in saw tooth profile.

79

Yuvraj N. Patil, V. S. Aneesh, M. T. Abhilash, Swati Kadlag

For orientation control, the min. and max. angles of servo motor drive the piston by its full length inside the cylinders. The difference in angles of heading vectors obtained from the orientation sensors is linearly proportional to the angle of rotation of both servo motors.

#### Acknowledgement

We would like to take this opportunity to express our heartiest gratitude to DIAT-DRDO(DU) Pune, and to all its students and faculty members of Electronics Engineering Department, Naval Wing, Navigation department, aerospace dpt. of DIAT-DRDO(DU) for their valuable suggestions and support throughout this work. We also thank Symbiosis Institute of Technology, Pune for providing valuable guidance and necessary facilities throughout the process.

## References

- Amit Ray, Singh S.N. and Seshadri V, 'Underwater Gliders Force Multipliers for Naval Roles' at the RINA conference 'Warship 2011 : Naval Submarines and UUVs', 29-30 June, 2011 at the Guildhall, Bath, UK. William P Barker, 'Analysis of undersea glider architectures and an assessment of undersea glider integration for undersea applications', Sep 12.
- 2. Danio Joe, R V Shashank Shankar, RVijay kumar and Arnab Das, 'Concept Design of Autonomous Underwater Glider.
- Arnab Das, "Naval Operations Analysis in the Indian Ocean Region-A Review", Accepted for publication in the IDSA Journal.
- Russ E. Davis, Charles C.Eriksen, Clayton P. Jones. "Autonomous Buoyancy-driven Underwater gliders". The technology and application of autonomous underwater vehicles, G.Griffiths,ed., Taylor and Fransis, London 2002.
- Charles C. Eriksen, T. James Osse, Russell D. Light, Timothy Wen, Thomas W. Lehman, Peter L. Sabin, John W. Ballard, and Andrew M. Chiodi Seaglider: A Long-Range Autonomous Underwater Vehicle for Oceanographic Research, IEEE Journal of oceanic Engineering, Vol. 26, No. 4, Oct 2001.
- Stephen Bozzone, Brian Newberry, Dominic Germana Enhancements and Performance of Seaglider Iridium Communications System Chris Yahnker, OCEANS, 2012.
- Abdallah Ismail, Project report on Programmable System-on-Chip technology from Cypress Semiconductor digital systems architecture, Dec 2007.
- M.D. Nair, Abhilash M.T., Vijay Kumar, Arnab Das "Embedded control system for underwater glider based on PSoC" et al International Journal of Computer and Electronics Research, Volume 3, Issue 2, April 2014
- Jeff Sherman, Russ E. Davis, W. B. Owens, and J. Valdes, "The Autonomous Underwater Glider Spray" IEEE JOURNAL OF OCEANIC ENGINEERING, VOL. 26, NO. 4, OCTO-BER 2001.
- Luke Stutters, Honghai Liu Senior Member, IEEE, Carl Tiltman and David J Brown, "Navigation Technologies for Autonomous Underwater Vehicles", IEEE TRANSAC-TIONS ON SYSTEMS, MAN AND CYBERNETICS - PART C: APPLICATIONS AND REVIEWS, VOL. 38, NO. 4, JULY 2008.
- R. E. Davis, "Preliminary results from directly measuring middepth circulation in the tropical and South Pacific," J. Geophys. Res., vol. 103, pp. 24 619–24 639, 1998.

Electronic System for Orientation Control and Obstacle Avoidance for Underwater Glider ...

- 12. M. J. McPhaden, "Genesis and evolution of the 1997-98 El Nino," Science, vol. 283, pp. 950-954, 1999.
- 13. S. A. Jenkins and J. Wasyl, "Optimization of glides for constant wind fields and course headings," J. Aircraft, vol. 27, pp. 632–638, 1990.
  14. http://www.mtu.edu/engineering/
- 15. http://realterm.sourceforge.net/

# Accuracy of Artificial Neural Network Models of Software Reliability Growth – A Survey

M. K. Saley<sup>1</sup> and Sasikumaran Sreedharan<sup>2</sup>

<sup>1</sup>Research Scholar, ManonmaniamSundaranar University, Thirunelveli, India saley\_mk@yahoo.com <sup>2</sup>Department of Computer Engineering, King Khalid University, KSA. drsasikumaran@gmail.com

Abstract: Software Engineering data is being analyzed by classical statistical methods and non parametric methods. Performance models are constructed using classical approach as a high maturity practice. Such practices are constrained by data quality and inadequacy of data analysis methods to treat data from real life projects. Data mining techniques can broaden the data analysis capability and improve prediction accuracy even with commonly presented data. Artificial neural networks are found as an improved prediction error estimation method against traditional parametric software reliability growth models. In this paper, we study prediction errors of Artificial Neural Networks (ANN) based Software Reliability Growth Models (ANN SRGM) with the objective of arriving at a criteria for selecting the methods having least prediction errors. All major works in ANN SRGM's are compared against that of parametric models. Then, inter-comparison of error performances of ANN SRGM's of different applications is made.

**Keywords:** Software Reliability Growth Model, Artificial Neural Networks, Prediction Accuracy, Root Mean Square Error of ANN, Learning Speed of ANN, Flexibility of ANN.

# 1 Introduction

Various attempts are being made to apply Artificial Neural Networks (ANN) to substitute traditional parametric Software Reliability Growth Models (SRGM). A new breed of ANN based SRGM's have been introduced by researchers in this field. While parametric models give clues to process composition and behavior, the nonparametric ANN SRGM's offer greater accuracy and flexibility [1].

Typical software reliability data consists of cumulative defects discovered during testing. The pattern may vary depending on how early defects are discovered, how promptly defects are reported, and how well testing process is planned. Because of these reasons data patterns vary from organization to organization, from project to

83

#### M. K. Saley, Sasikumaran Sreedharan

project. Reliability growth is concerned with the pattern by which defect discovery progresses towards the ideal plateau region of defect free state. This is a time series problem and to study reliability growth models, one does not consider software structure or software development process composition. It is a black box view on defect discovery pattern. Parametric models use mathematical equations that may use test engineering parameters like test efficiency, test effort, software size, review speed etc; such models, when constructed, provide insight into what is happening during testing. ANN SRGM's do not aim at obtaining such an insight; in fact they are incapable of seeing the hidden factors of structural aspects. The purpose of ANN SRGM is to learn from data patterns and predict pending defects.

The inability of ANN SRGM to provide insight into problem structure is richly compensated by the greater prediction accuracy, which emerges as the most important criteria by which we can judge the performance of ANN SRGM's. In this paper, we consider the pioneering work done by several researchers and study the results reported by them, providing an inter-comparison of errors.

# 2 Comparison of Performance of Parametric Models with ANN SRGM

The key question that concerns users of Artificial Neural Network Software Reliability Growth Models (ANN SRGM) is whether the proposed ANN SRGM will provide better accuracy than existing parametric models such as Goel-Okumoto Model, Delayed S Model, Logarithmic Model, Exponential Model, Power Model, and Inverse Polynomial Model. Every researcher who has proposed ANN SRGM has answered this question in favor of ANN SRGM. [2] Comments and conclusions such as "ANN provides better results" are common. Such conclusions are not objective enough.

**Hypothesis Test 1.** The first objective error data comparing ANN with parametric models has come from Karunanidhi [3] and is summarized in Table 1. The error metric data Average Error (AE) % has been selected and used to construct Table 1.

Five parametric models and six ANN SRGM's are considered in Table 1. Error data is provided for each model against three data sets. On this data, a hypothesis test has

			AE %	
CATEGORY	MODEL	DATA 1	DATA 2	DATA 3
PARAMETRIC	LOGARITHMIC	16.84	5.25	12.48
PARAMETRIC	INV POLY	19.4	4.6	13.29
PARAMETRIC	EXP	28.35	11.93	15.87
PARAMETRIC	POWER	18.35	12.16	12.95
PARAMETRIC	DELAYED S	35.78	17.71	27.1
ANN	FFN GEN ENC	26.75	14.51	30.2
ANN	ELMAN	24.73	13.24	19.6
ANN	FFN GEN	6.63	3.52	10.24
ANN	FFN	3.76	2.32	12.32
ANN	JORDAN GEN	3.05	3.11	6.96
ANN	JORDAN	2.68	3.21	9.52

Table 1. End Point Error from Karunanidhi's Data [3].

Hypothesis Testing	Hypothesis Testing								
T-Test of difference $= 0$ (	vs ≠)								
MODEL	N	AE %	AE %	SE					
MODEL	IN	Mean	StDev	Mean					
ANN SRGM	18	10.91	8.99	2.1					
PARAMETRIC	15	16.80	0 20	2.2					
MODEL	15	10.80	0.50	2.2					
Difference	μ (ANI	N) - μ (PARA	METRIC)						
Estimate for difference	-5.90								
95% CI for difference	(-12.08, 0.29)								
T-Value	-1.95								
P-Value	0.061								
Degrees of Freedom	30								

 Table 2. Two Sample t – Test Results on difference in AE % between

 ANN SRGM and Parametric Models.

been done by the authors to see if ANN SRGMs fare better than parametric models. A group is formed with error data for six ANN SRGMs and three data sets with a total of eighteen data points. Another group is formed with error data for the five parametric models and three data sets, with a total of fifteen data points. A t-Test has been done on the two groups of data and the results are shown in Table 2.

The mean values of average errors are 10.91 and 16.8 for ANN and parametric models, clearly showing an improvement of 5.90 in the mean. This is good enough in normal situations to go in for ANN; however, in the light of skepticism and criticism that prevail upon ANN a more stringent judgment is called for. The t - Test p value provides this information and is only 0.061. Had the p value been less than 0.5 we could have acknowledged, by a popular rule, significant improvement. With a borderline value of 0.061, ANN experiments have not yet delivered convincing results with statistical significance.

In addition to the t Test, a visual comparison of the two error data groups has been done using Turkey's box plot as shown in Fig. 1. The most common form of the box plot is constructed with median value at the center line and, quartiles in the box edges and whiskers that stretch from the box to connect with the extreme values. The box plot offers a more robust comparison, compared to a t-Test, and is not dependent on data normality.

**Hypothesis Test 2.** Karunanidhi's data is typical of ANN performance. Many other researchers support the view that ANN is better than parametric models. There have been many results reported in clear support of ANN. Sultan's data [4] is an example of results that are not so convincing. Sultan uses Summed Square Error (SSE), as the error metric and considers three types of projects, shown in Table 3.

A t-Test done on Sultan's data is shown in Table 4. The mean SSE values are 57.2 and 60.3 for ANN and Parametric Models. ANN scores better, with less mean error. The difference is only marginal. But the p value is 0.970 which suggests that ANN has

Table 3. Test Errors from Sultan's Data [4].

SUM SQUARE ERR	OR	
PROJECT	PARAMETRIC	ANN
Military	168.54	160.3887
Real Time Control	1.368	1.2542
Operating System	10.885	9.9623

#### M. K. Saley, Sasikumaran Sreedharan

Hypothesis Testing						
T-Test of difference = $0$ (v	rs ≠)					
MODEL		AE %	AE %	SE		
MODEL	N	Mean	StDev	Mean		
ANN SRGM	3	57.2	89.5	52		
PARAMETRIC	2	60.2	02.0	54		
MODEL	3	00.5	95.9	54		
	1					
Difference	$\mu$ (ANN)	- μ (PARAME'	TRIC)			
Estimate for difference	-3.1					
95% CI for difference	(-241.4, 2	.35.2)				
T-Value	-0.04					
P-Value	0.970					
Degrees of Freedom	3					

**Table 4.** t -Test Results on Squared Sum of Error difference betweenParametric and ANN SRGM.

not produced significantly different results from Parametric Models. With such a high p value, one would tend to consider them equal.

This is a clear example of a situation where ANN performance is equal to parametric model performance, but offers an advantage, though small.

# 3 Sensitivity of End Point Errors to Data Patterns – A Hypothesis Test

A limitation of ANN SRGM arises out of its difficulty to cope up with changes in data patterns. Its ability to forecast depends on smoothness of training data. Typically the data is the cumulative defects found during testing and its pattern can vary from project to project, depending on the quality of testing and management of testing process. There could be sudden jumps and micro plateaus; the inflexion point could change if proportion of early discovery changes. The finishing line flatness could also change depending on test case effectiveness. ANN SRGM trained with such turbulent patterns may show larger errors during testing. Also, it has been shown that the number of hidden nodes required for optimum errors vary if the data set changes [5]. As a result software reliability data sets and the actual patterns of data seem to have an influence on ANN SRGM prediction errors (end point errors).

Sultan's [4] results, shown in Table 3, exhibits dramatic variation of SSE when data set changes. SSE could vary from a value as low as 1.25 in one project dataset to a value as high as 160.39 in another project dataset. The change is so dramatic and obvious that without using a sophisticated hypothesis testing one can arrive at the judgment. The problem exists even with parametric models. Basically ANN SRGM has not solved a fundamental problem: reliability models fail to show consistent performance across datasets. If data patterns could dictate errors so much, it is not good news for those who wish switch over to ANN SRGM's, because they do not see any advantage in doing so, from the point of view of data-pattern sensitive performance.

A different situation emerges if we examine Karunanidhi's data shown in Table 1. To test this hypothesis that data sets control errors, Analysis of Variance (ANOVA) has been done on ANN SRGM AE % for three different data sets. ANOVA results are shown in Table 5.

Accuracy of Artificial Neural Network Models of Software Reliability Growth ...

Table 5. ANOVA Results of Dataset Sensitivity of Errors.

Hypothesis	Testing: A	NOVA			
Source	DF	Adj SS	Adj MS	F-Value	P-Value
DATA	2	200.7	100.33	1.28	0.306
Error	15	1174.7	78.31	-	-
Total	17	1375.4	-	-	-

The p value is 0.306, indicating that data sets do not have significant influence on errors. This means that ANN SRGMs have overcome the problem of data-pattern sensitivity.

# 4 Inter-comparison of Prediction Errors of ANN SRGM'S

**Normalized Data.** Prediction errors in six experiments with ANN Software Reliability Growth Models performed from 1992 till 2013 by various researchers have been compiled and shown in Table 6. The error metric selected for this inter-comparison is AE %

Ref.No	Name of the Author	Year published	Name of NN	AE %
		1	FFN GEN ENC DS1	26.7500
			FFN GEN ENC DS2	14.5100
			ELMAN DS1	24.7300
			ELMAN DS2	13.2400
			FFN GEN DS1	6.6300
	NachimuthuKarunanidhi,		FFN GEN DS2	3.5200
[3]	Darrel Whitley and	1992	FFN DS1	3.7600
	Yaswant K. Malaiya.		FFN DS2	2.3200
	-		FFN DS3	12.3200
			JORDAN GEN DS1	3.0500
			JORDAN GEN DS2	3.1100
			JORDAN DS1	2.6800
			JORDAN DS2	3.2100
			FFN generalization	0.0667
(7)	NachimuthuKarunanidhi,	1000	FFN prediction	0.0612
[7]	Darrell Whitley and	1992	JN generalization	0.0475
	Yaswant		JN prediction	0.0494
			PSNN 1	1.890
		2000	PSNN 2	4.650
[5]	Jun Zheng	2009	PNNE 1	1.290
			PNNE 2	1.630
			MLFFN 1	20,790
			MLFFN 2	7.350
[10]	Jung-Hua Lo	2009	MLFFN 3	8.900
			MLFFN 4	2.260
			FFNW 1	0.100
		F	FFNW 2	0.050
[11]	Yogesh Singh, Pradeep	2010	FFNW 3	0.120
[11]	Kumar		FFNW 4	0.340
		-	EENW 5	0.830
				0.0.00
			PSO	7.180
[12]	Rita G. Al gargoor, Nada	2013	PSO PSO	7.180

Table 6. Normalized Data of ANN SRGM Test Errors.

#### M. K. Saley, Sasikumaran Sreedharan

defined in Equation 2. This metric has been used in the pioneering work of Karunanidhi [3], [7] and followed by several researchers. Prediction errors (AE %) for Multi-Layer Perceptron (MLP), Elman, Jordan, PNNE (Ensemble), PSNE (Single Input Single Output Three Layer), PSO networks have been compiled. AE % data shown in Table 6 is normalized data and permits inter-comparison.

**Non-Normalized Data.** Non-normalized data (scale dependent) from nine experiments is compiled in Table 7. This data does not allow inter-comparison across different experiments performed by different authors. This data is provided just for reference. The metrics used in this data are MSE, RMSE and SSE. Metrics such as NRMSE would permit inter-comparison; however, unfortunately, it is available for only one experiment from a single paper.

N	NON-NORMALIZED DATA OF PREDICTION ACCURACY OF ANN							
Ref. No	Name of the Author	Year published	MSE	RMSE	NRMSE	NSE	SSE	
	Sultan U. Alaa Shata and		9.800	-	-	-	-	
[4]	David Rine	2001	15.826	-	-	-	-	
	David Kile	12.476 - 0.1455 - - 0.1504 - 0.1574 -	-					
			-	-	0.1455	-	-	
			-	-	0.1504	i.	i.	
			-	-	0.1579	ł.	ł.	
			-	-	0.1527	•	•	
røi	N Rai Kiran and V Ravi	2007	-	-	0.1669	-	-	
[8] 1	IN. Raj Kitali alki v Ravi	2007	-	-	- 0.1479 -			
				0.1611	-	-		
			-	-	0.1476	-	-	
			-	-         0.1611         -         -           -         0.1476         -         -           -         0.1162         -         -           -         0.1115         -         -           0.6677         -         -         -	-			
			-	-	0.1115	-	-	
			-	0.6677	-	-	-	
101	Sultan H. Aljahdali and	2008	-	0.1591	-	1	ł	
[9]	Khalid A. Buragga	2008	-	0.1394	-	0.1476     -       0.1162     -       0.1115     -       -     -       -     -       -     -       -     -       -     -       -     -       -     -       -     -       -     -       -     -       -     -       -     -       -     -       -     -		
	[9] Khalid A. Buragga		-	0.9079	-	ł	ł.	
[12]	Lu Zhao Jian-nei Zhang	2010	0.0486	-	-	-	-	
[12]	Lu Zhao, Jian-per Zhang	2010	0.0590	-	-	ł.	ł.	
[12]	Kewen LI, Lina GONG,	2010	20.5426	-	-	•	•	
[15]	Jisong KOU	2010	0.3642	-	-	1	ł.	
[14]	V.Ramakrishna,MR	2012	-	-	-	0	-	
[14]	Narasinga Rao and TM	2012	2012	-	-	0.07	-	
	Rita G. Al gargoor, Nada		-	0.0184	-	-	-	
[15]	N Saleem	2013	-	0.0186	-	-	VN SSE - - - - - - - - - - - - - - - - - -	
	in Saleelli		-	0.0180	-	-	-	
	Sultan H. Alaa Sheta and		-	-	-	-	160	
[16]	David Rine	2001	-	-	-	-	1.25	
			-	-	-	-	9.96	

Table 7. Non-Normalized Data of ANN SRGM Test Errors.

Inter-comparison of ANN SRGM with normalized data



Accuracy of Artificial Neural Network Models of Software Reliability Growth ...

Figure 1. Inter-comparison of Average Errors of Various Types of ANN SRGMs.

It is rather difficult to perform a hypothesis test to find which ANN is better, because of multiplicity of data origin and inequality of sample size. Moreover, data sets used in the six groups of data are different, introducing an element of uncertainty. Instead of the classical hypothesis test a box plot comparison, the next robust alternative, has been performed. Figure 1 shows the inter-comparison results in the form of box plots of data given in Table 6. As usual the box plot comprises of median, quartiles and whiskers and, in our cases, we have included the mean values for reference and connected the mean values by a thin line.

If we compare the medians of the box plots, the ensemble PNNE has the lowest error. If we compare the first quartiles (lower edges of boxes) then MLP shows lowest error. Clearly Elman network shows the highest error. PSO shows larger errors than PNNE and PSNN.

MLP performance requires a special analysis. The dispersion of MLP data is high. One must remember that MLP sample size is also high and data has arrived from heterogeneous sources. The mean value of MLP errors is high because of skew in the data and influence of extreme values. The median shows a fair picture. When data is skewed, mean is not a reliable indicator of central tendency. It is the quartile of the box that makes us think that MLP does show possibility of better performance. To validate this theory we can look at the raw data in Table 6 where we find Yogesh's results [8] with MLP are impressive and are responsible for the lower quartile of the box.

While one has to take the box plot results with a grain of salt because of inherent uncertainties, it is evident that without using recursion and special optimization, geed results can be obtained with simple plain MLP. This evidence motivates one to pursue MLP as an attractive candidate for ANN SRGM. There is no compelling evidence in favor of pursuit of sophistication in ANN topologies.

# 5 Conclusion

It is seen that several researchers have attempted various realizations of ANN SRGMs in the past 22 years. Each experiment is a discovery an ANN SRGM topology that works better than selected statistical models. Variability of ANN performance with data

#### M. K. Saley, Sasikumaran Sreedharan

sets is a concern. Inadequate attention to characteristics of trained networks leaves ANN less trustworthy in the critical eyes of ANN users. It is noted that input data preparation and structuring extend ANN performance to greater levels. Further refinement is obtained by the choice of judicious activation functions that resemble the empirical data patterns

This survey finds ANN SRGM is better than statistical models for reliability prediction. However, the statistical significance of results is still low, with a p value of 0.61, leaving a wide scope for further research and improvement in ANN SRGM. The errors are comparable and in some cases better than errors achieved in other domains.

#### References

- Saley M.K and SasikumaranSreedharan, "A Survey of Software Reliability Growth Models Using Non-Parametric Methods", IEEE International Conference on Computational Intelligence and Computing research", Dec. 2014.
- Doteswamy and Chanabasayya M Vastrad, "Performance Analysis of Oxazolens and Oxazoles Derivatives Description Dataset", Vol 03, No 6, International Journal of Information Sciences and Techniques, Nov 2013.
- NachimuthuKarunanidhi, Darrel Whitley and Yaswant K. Malaiya., "Prediction of reliability model using Connectionist Models," IEEE transactions on software engineering, Vol. 18, No. 7, July 1992. pp. 563–574.
- 4. Sultan H, AlaaSheta and David Rine, "Prediction of Software Reliability: A comparison between Regression and Neural Network Non-Parametric Models," Computer Systems and Applications, ACS/IEEE International Conference, pp. 470–473, June 2001.
- 5. Jun Zheng, "Predicting software reliability with neural network ensembles", ELSEVIER, Expert Systems with Applications 36, 2116–2122, 2009.
- 6. ZhigangGan, Nanfeng Xiao, "An Improved Adaptive Boosting Algorithm for Neural Network Ensemble Based on Multi-dimensional Cloud Model", Second International Symposium on Knowledge Acquisition and Modeling, IEEE Computer Society, 2009.
- 7. NachimuthuKarunanidhi, Darrell Whitley and Yaswant, "Using Neural Networks in Reliability Prediction," IEEE software, 0740-7459/92/0700/0053, July 1992.
- N. Raj Kiran and V Ravi, "Software Reliability Prediction Using Wavelet Neural Networks", IEEE computer society, pp.195–199, 2007.
- 9. Sultan Aljahdali and Alaa F sheta, "Predicting the Reliability of software System using Fuzzy Logic", IEEE 2011.
- Jung-Hua Lo "The Implementation of Artificial Neural Networks Applying to Software Reliability Modeling", IEEE Chinese Control Decision Conference, 2009.
- 11. Yogesh Singh and Pradeep Kumar, "Prediction of software reliability Using Feed Forward Neural Networks" Computational Intelligence and Software Engineering (CiSE), International Conference, 2010.
- Lu Zhao, Jian-pei Zhang et al., "Software reliability Growth Model based on Fuzzy Wavelet Neural network", IEEE 2<sup>nd</sup> International Conference on Future Computer and Communication, Vol. 1, pp. VI 664–VI 668, 2010
- Kewen LI, Lina GONG, Jisong KOU, "Predicting Software Quality by Fuzzy Neural Network Based in Rough Set", Journal of Computational Information systems, pp. 1439–148, May 2010.

- V.Ramakrishna, MRNarasingaRao and TM Padmaja "Software reliability prediction using Neural Networks", International journal of Computer Applications (0975 – 887) Vol. 60-No.&, December 2012.
- Rita G. Al gargoor, Nada N Saleem, "Software Reliability Prediction Using Artificial Techniques", IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 4, No. 2, ISSN (Print): 1694-0814, pp. 274–281, July 2013.
- Sultan H, AlaaShetta, David Rine, "Prediction of Software Reliability: A comparison between Regression and Neural Network Non-Parametric Models," Computer Systems and Applications, ACS/IEEE International Conference, pp. 470–473, June 2001.
- G.Q. Shang, C.H Sun, "Application of BP Neural Network for Predicting Anode Accuracy in ECM", International Symposium on Information Science and Engineering, IEEE Computer Society 2008. pp. 428–432.
- Qi Feng, Shengjun Wu, Yun Du, HuaipingnXue, Fei Xiao Ban, and Xiaodong Li, "Improving Neural network Prediction Accuracy for PM Individual Air Quality Index Pollution Levels", Environmental Engineering Science, Volume 30, 2013.
- Lihua Chen, Yuchen Wang, "Application of Radial Basis Function Neural Network on the Prediction of Urban Built-up Area", Applied Mechanics and Materials, Trans tech Publications, Switzerland, Vols. 556–562, pp. 5308–5311. 2014.
- Di Lu and Jing Liu, "Application of Improved Neural Network in the Automotive Engine Fault Diagnosis", International journal of Control and Application, Vol. 7, No. 4, pp. 262–355, 2014.
- 21. Fuqiang Zhou, Zhen Su, Xinghua Chai and Lipeng Chen, "Detection of Foregin Matter in Transfusion Solution Based on Gaussian Background Modeling and an Optimized BP Neural Network", Sensors ISSN 1424-8220, 2014.
- A. Noriega, D Blanco, B.J. Alvarez, A.Garcia, "Dimensional Accuracy improvement of FDM square cross-section parts using artificial neural networks and an optimization algorithm", Int J Adv Manuf Technol, Springer, 2013.
- 23. Kunwar Singh Vaisla and Ashuthosh Kumar Bhatt, "An Analysis of the Performance of Artificial Neural Network Technique for Stock Market Forecasting", Vol 02, No. 06, International Journal on Computer Science and Engineering, 2010.

# Development of Dual-Actuated Stage for Positioning Applications

Prabha Niranjan<sup>1</sup>, K. V. S. S. S. S. Sairam<sup>1</sup>, Shashikantha Karinka<sup>2</sup>

<sup>1</sup>Department of Electronics and Communication Engineering, <sup>2</sup>Department of Mechanical Engineering, NMAMIT, Nitte. prabhan@nitte.edu.in

Abstract. Piezoelectric actuators have been widely used in precision positioning applications because of their fast response, fine resolution, and large actuating forces. The displacement of piezoelectric actuators should be precisely controlled to use them in precision actuation stages. However to use these piezoelectric actuators in control systems, it is very much necessary to estimate the displacement of actuator based on their input voltage and also the inverse model. This paper presents Prandtl-Ishlinskii model for describing the nonlinearity of the actuator. Inverse Prandtl-Ishlinskii model will be used in the feed forward loop for the compensation of nonlinearity. This research work also includes development of a dual-actuated single stage to overcome the small travel range of piezoelectric stack actuators. The piezoelectric stack actuator is used for fine positioning and ball screw actuator is used for coarse positioning in the proposed dual-actuated stage. A closed loop control will be designed for the dual positioning stage, and this stage will be later applied for real-time wing flap control of a ircraft model.

**Keywords:** Piezoelectric actuator; Prandtl-Ishlinskii model; Positioning; Closed loop control system; Displacement; Dual-Stage; Ball screw actuator.

# **1** Introduction

In recent years, there is lot of scope for research in ultra precision positioning applications. It includes diamond cutting, drilling machine, wing flap control in aircraft etc. Such ultra precision position systems require displacement control in the range of micrometers to millimeters.

In ultra-precision positioning, the piezoelectric actuators are used because of their good dynamic response and high resolution. It can be used for nano-positioning drives but it is bit tedious to provide higher load capacity. Whereas ball screw provides higher load capacity and stiffness but it fails in providing accuracy in sub-micrometer range [1],[2].

Hence for high precision machines, dual positioning systems have been developed where piezoelectric actuator is used for fine positioning and ball screw actuator is used for coarse positioning [3], [4], [5].

93

Prabha Niranjan, K. V. S. S. S. S. Sairam, Shashikantha Karinka

In this paper, the various hysteresis models are discussed in section 2. The simulation model of piezoelectric actuator is presented in section 3. The proposed dual-actuated single stage for positioning applications is given in section 4, followed by conclusion and future work.

# 2 Hysteresis Modeling

Piezoelectric actuators (PEAs) are widely used in variety of positioning applications due to their advantages of high positioning resolution in the range of micrometers to nanometers, large force, high dynamic response, high stiffness and compact design [6], [7], [8]. The major drawback of the PEA is its hysteresis nonlinearity that leads to positioning inaccuracy, and even instability of closed loop systems [9]. In order to increase the positioning accuracy of the PEA, there is a need to model the PEA, taking into consideration hysteresis and creep nonlinearities, which otherwise lead to large errors.

The various mathematical models which are popularly used to describe the hysteresis phenomenon like Preisach model, Prandtl-Ishlinskii (PI) model, Maxwell-slip model, etc. are proposed by various researchers [10].

In Preisach model, much of data is required to characterize the nonlinear systems that will influence the accuracy of the model. Obtaining inverse model is not easy in this type due to double integral present in the equation [11],[12],[13].

Maxwell-slip model operates with an elasto slide element comprising of mass less linear spring and mass less block those are susceptible to coulomb friction. One of the limitations with Maxwell-slip model is about hysteresis rising curve. The smart actuators need to start from a relaxed state that is not easily acquired in case of this model [14].

Many Researchers have used the sub class of Preisach model, known as Prandtl-Ishlinskii (PI) model for hysteresis compensation of PEA. The PI hysteresis model is the super position of several backlash operators, which are similar to the relay operators used in the Preisach model [15],[16]. The Preisach model is more accurate compared to PI model, but PI model is chosen due to availability of analytical inverse. In PI model only with few backlash operators, the hysteresis can be accurately modeled.

### **3** Simulation Model

PI hysteresis model is a phenomenological operator based model, used to describe the nonlinear relationship between the applied voltage and the resultant displacement of PEAs. In this model, the output is a weighted sum of many backlash operators. This model has been widely used for real time compensation of hysteresis nonlinearity in control applications due to its reduced complexity and the analytical form of inversion as compared to Preisach hysteresis model.

The backlash operator is the primary operator of the PI model, and it is defined in the discrete domain as follows:

$$y(k) = H_{r}[x, y_{0}](k) = \max\{x(k) - r, \min\{x(k) + r, y(k-1)\}\}.$$
(1)

where, x = c1 \* xdata + c2, where c1 and c2 are constants and xdata is the applied input to the model, y is the actuator response and r is the single threshold value of the back-lash operator.

The initial condition of equation (1) is given by,

$$y(0) = \max\{x(0) - r, \min\{x(0) + r, y_0\}\}.$$
(2)

The hysteresis loop can be modeled by a linearly weighted superposition of many backlash operators with different threshold and weight values, as given in equation (3).

$$\mathbf{y}(\mathbf{k}) = \mathbf{w}^{\mathrm{T}} \cdot \mathbf{H}_{\mathrm{r}}[\mathbf{r}, \mathbf{y}_{0}](\mathbf{k}).$$
(3)

where weight vector,  $\mathbf{w}^{T} = [\mathbf{w}_{1} \ \mathbf{w}_{2} \ \mathbf{w}_{3} \ \dots \ \mathbf{w}_{n}]$  and initial value vector,  $\mathbf{y}_{0}^{T} = [\mathbf{y}_{01} \ \mathbf{y}_{02} \ \mathbf{y}_{03} \ \dots \ \mathbf{y}_{0n}]$ , and threshold vector,  $\mathbf{r} = [\mathbf{r}_{1} \ \mathbf{r}_{2} \ \mathbf{r}_{3} \ \dots \ \mathbf{r}_{n}]^{T}$ ;  $\mathbf{0} = \mathbf{r}_{1} < \mathbf{r}_{2} < \mathbf{r}_{3} < \dots < \mathbf{r}_{n}$ .

The threshold values of the backlash operators are selected within the applied voltage range, with equal intervals. The weight values decide the shape and size of the hysteresis loop, which are found by the experimental data. Also it is assumed that the actuator starts with its de-magnetized state. So the initial value is always zero.

#### 3.1 Parameter Estimation

First the PEA response is measured by driving the actuator up to the full displacement range. In this study, 0–150V, input with 1Hz sinusoidal signal is given to PEA and the corresponding displacement readings are noted. Then voltage and displacement readings are imported to MATLAB least-squares optimization program for parameter calculation with lsqcurvefit function.

This requires a user defined function to compute the vector valued function F(x, xdata). This function will find coefficients x that best fit the equation, F(x, xdata). In this model, a total of 17 backlash operators are used in order to demonstrate the hysteresis loop. Geometric progression method is implemented, in order to choose the threshold parameters. The values of constants c1 and c2 are calculated based on equation x = c1 \* xdata + c2, they are, c1 = 0.973693604078441 and c2 = -2.88447749468157.

Threshold Value  $r_i$  (V) and Weight Value  $(w_i)$  are calculated based on equation (1) - (3). The Inverse PI model can be obtained from the identified PI model. Because of the analytical inverse, the design of inverse PI model becomes more straightforward. The parameters  $r_i$  and  $w_i$ , for the inverse PI model are determined based on equations (4)–(7).

$$H^{-1}[y(k)] = w^{T} \cdot H_{r}[y, y_{0}](k) = \sum_{i=1}^{n} w'_{i} \cdot \max\{y(k) - r'_{i}, \min\{y(k) + r'_{i}, y'_{i}(k-1)\}\}.$$
 (4)

where, 
$$\mathbf{r}'_{i} = \sum_{j=1}^{i} W_{j}(\mathbf{r}_{i} - \mathbf{r}_{j}), i = 1, 2, ... n.$$
 (5)

$$y'_{oi} = \sum_{j=i+1}^{i} w_j y_{0i}, i = 1, 2, ... n$$
 (6)

Prabha Niranjan, K. V. S. S. S. S. Sairam, Shashikantha Karinka

$$w_{1}' = \frac{1}{w_{1}}, w' = -\frac{w_{i}}{\left(w_{i} + \sum_{j=2}^{i} w_{j}\right)\left(\sum_{j=1}^{i-1} w_{j}\right)}, i = 2, 3, \dots n.$$
(7)

The values of constants are found based on equation, x = c1 \* xdata + c2 they are, c1 = 3.02098764835687 and c2 = 0.676064852409702.

#### 3.2 Simulation Results

The hysteresis nonlinearity of PEA is mathematically modeled using PI hysteresis model for sinusoidal signal of 150V, 1Hz. The output of forward and inverse PI model for sinusoidal input signal is given in Figs.1 and 2. Results show an error of 0.5 % for forward model and 0.8% for inverse PI model, which is shown in Fig. 3. This shows that the PEA can be accurately modeled using sinusoidal input signal, due to fine variation at the peak and valley points of the signal.



Fig. 1. a) PI hysteresis model output of PEA with 150V, 1Hz sinusoidal signal (b) Hysteresis loop.



**Fig. 2.** a) PI inverse hysteresis model output of PEA with 150V, 1Hz sinusoidal signal (b) Inverse hysteresis loop.

Development of Dual-Actuated Stage for Positioning Applications



Fig. 3. Modeling error (a) PI forward model (b) PI inverse model.

# 4 Proposed Method

The block diagram of the experimental setup for the proposed work is as shown in Fig. 4. This setup has a piezoelectric stack actuator with very small travelling range within 20 micro meters, which can be used for fine positioning. And a ball screw with built in linear encoder is also used in the stage for coarse positioning. A single control system will be developed for the position control of the entire system. This system can be used for both micro and macro applications.



Fig. 4. Block diagram of proposed dual-actuated stage.

# 5 Conclusion and Future Work

The effectiveness of the hysteresis model is validated by comparing the experimental data with the simulation results. The modeling error is comparatively very small so that the simulated model can be further utilized in the feed forward loop for control design.

In the later part of the work, a dual-actuated stage will be developed with piezoelectric actuator for fine positioning and ball screw for coarse positioning. Control system will be developed for the real-time control of the dual stage. Developed scheme will be applied for the position control of wing flap of aircraft model as one of the example.

# References

 Matsubara, A., Fujitha, T., Kono, D., Tanaka, N., Watanabe, Y.: Nano-positioning drive with Piezoelectric Actuator Integrated into support bearing unit of Ball screw, pp. 323–326 (2007).

97

Prabha Niranjan, K. V. S. S. S. S. Sairam, Shashikantha Karinka

- Watanabe, S., Furutani, T., Yoshida, S.: Nanometer Positioning and its Micro-dynamics, Nanotechnology, Vol. 1, pp. 31–37 (2008).
- Moriyama, S., Harada, T., Takanashi, A.: Precision X-Y Stage with a piezo-driven fine table, in the Bulletin of the Japan Society of Precision Engineering, Vol. 50, pp. 718–723 (1984).
- Lee, C., Kim, S.: An Ultraprecision Stage for Alignment of Wafers in Advance Microlithography, in Precision Engineering, Vol. 21, pp. 335–345 (1997).
- Pahk, H.J., Lee D.S., Park, J.H.,: Ulta Precision Positioning System for Servo Motor Piezo Actuator using DualnServo Loop and Digital Filter Implementation, in the International Journal of Machine Tools and Manufacture, Vol. 41, pp. 51–63 (2001).
- Woronko, A., Huang, J.,: Piezoelectric Tool Actuator for Precision Machining on Conventional CNC Tuning Centers, Precision Engineering, Vol. 45, pp. 335–345 (2003).
- Leang, K., and Devasia, S.: Design of Hysteresis Compensating Iterative Learning Control for Piezo Positioners: Application to Atomic Force Microscopes, Mechatronics, vol. 16, no. 4, pp. 141–158 (2006).
- Gawthrop P., Bhikkaji B, and Mohimani S.: Physical Model Based Control of a Piezoelectric Tube for Nano Scale Positioning Applications, Mechatronics, vol. 20, no. 1, pp. 74–84 (2010).
- 9. Salapaka, S.: Scanning Probe Microscopy, IEEE Control Systems, vol. 28, no. 2, pp. 65–83 (2008).
- Yang, M., G. Gu and Zhu, L.: High Bandwidth Tracking Control of Piezoactuated Nano Positioning Stages Using Closed Loop Input Shaper, Mechatronics, vol. 24, no. 6, pp. 724–733 (2014).
- Ge, P., and Jouaneh, M.,: Tracking Control of a Piezoceramic Actuator, IEEE Trans. Control Syst. Technology, vol. 4, no. 3, pp. 209–216 (1996).
- Mayergoyz, I. D and Friedman, G.: Generalized Preisach Model of Hysteresis, IEEE Transactions on Magnetics, vol. 24, no. 1, pp. 212–217 (1998).
- Song, G., Zhao, J., X. Zhou and J. A. De Abreu-Garcia: Tracking Control of a Piezoceramic Actuator with Hysteresis Compensation Using Inverse Preisach Model, IEEE/ASME Transactions on Mechatronics, vol. 10, no. 2, pp. 198–209 (2005).
- Michel G. and Nikola C.: Modeling Piezoelectric Stack Actuators for Control of Micromanipulation, IEEE Control Systems, vol. 17, no. 3, pp. 69–79 (1997).
- Bashash S., Jalili N.: Robust Multiple Frequency Trajectory Tracking Control of Piezoelectrically Driven Micro/Nano Positioning Systems, IEEE Transactions on Control Systems Technology, vol. 15, no. 5 (2007).
- Kuhnen, K.: Modeling, Identification And Compensation Of Complex Hysteretic Nonlinearities: A Modified Prandtl-Ishlinskii Approach, European Journal of Control, vol. 9, no. 4, pp. 407–421 (2003).

# Chronological Advancement in Compiler Design: A Review

Amit Verma<sup>1</sup> and Nikita Bakshi<sup>2</sup>

Department of Computer Science and Engineering, Chandigarh University, Mohali, India. <sup>1</sup>amit.verma@cumail.in <sup>2</sup>bakshinikita24@gmail.com

Abstract. Compiler is a set of instructions that translate the source code into binary format usually known as object code. Compiler is used to convert a language readable in user domain into the tasks which is understood by the machine. For example C++ compiler compiles program written in a language easily understandable by human which compiles task that can be executed by a computer's processor. In this paper, we discuss about the computers and their evolution. Algorithms and tools are used for compiler design. Further our study carrys a survey on key properties of compiler courses in some universities.

**Keywords:** Computer, Compiler, Programming language, Memory Aware Mapping, Automatic Parallelizing, Lemon.

# **1** Introduction

Computer is an electronic device which is competent of receiving information in a particular form and performing a set of instructions with predetermined but variable set of instruction to produce a result in the form signals. Sequence of operations can be easily changed in computer so it can solve more than one kind of problem like logical operations etc.

### 1.1 Evolution of Computers

*1822:* English mathematician Charles Babbage conceives of a steam-driven calculating machine that would be able to compute tables of numbers. The project, funded by the English government, is a failure. More than a century later, however, the world's first computer was actually built. [15]

*1890:* Herman Hollerith designs a punch card system to calculate the 1880 census, accomplishing the task in just three years and saving the government \$5 million. He establishes a company that would ultimately become IBM (IBM was founded in 1911). [13]

*1937:* J.V. Atanasoff, a professor of physics and mathematics at Iowa State University, attempts to build the first computer without gears, cams, belts or shafts. [14]

99

#### Amit Verma, Nikita Bakshi

*1941:* Atanasoff and his graduate student, Clifford Berry, design a computer that can solve 29 equations simultaneously. This marks the first time a computer is able to store information on its main memory. [15]

*1943–1944:* Two University of Pennsylvania professors—John Mauchly and J. Presper Eckert—build the Electronic Numerical Integrator and Calculator (ENIAC). Considered the grandfather of digital computers, it fills a 20 foot by 40 foot room and has 18,000 vacuum tubes. [16]

*1946:* Mauchly and Presper leave the University of Pennsylvania and receive funding from the Census Bureau to build the UNIVAC, the first commercial computer for business and government applications. [16]

*1953:* Grace Hopper develops the first computer language, which eventually becomes known as COBOL. Inventor Thomas Johnson Watson, of IBM CEO Thomas Johnson Watson, conceives the IBM 701 EDPM to help the United Nations keep tabs on Korea during the war. [15]

1954: The FORTRAN programming language came into existence.

*1960–1962:* In 1960, COBOL became an early high-level programming to be compiled on different architectures. In 1962, the first self-hosting compiler was assigned for Lisp by Tim Hart and Mike Levis at MIT. [15]

*1980–1999:* The term **Wi-Fi** becomes part of the computing language and users begin connecting to the Internet without wires.

2000–2015: Apple unveils the **iPad**, changing the way consumers view media and jumpstarting the dormant tablet computer segment.

#### 1.2 Compiler

Compiler is a program that translates source code into object code. The compiler derives its name from the way it works, looking at the entire piece of source code and collecting and reorganizing the instructions. Thus, a compiler differs from an interpreter, which analyzes and executes each line of source code in succession, without looking at the entire program. [17]

#### 1.3 Phases of Compiler

A compiler efficiently generates object code by confirming code syntax. At run time, output is formatted according to the rules of linker and assembler. A compiler consists of:

*a. Lexical Analysis:* Lexical analysis is used to remove irrelevant information from the program source. Irrelevant information contains things like blanks and comments. Besides eliminating irrelevant information, lexical analysis determines the lexical tokens of the language.

*b. Syntax Analysis*: Syntax Analysis is responsible for looking syntax rules of the language (often as a context-free grammar), and construction of an intermediate representation of the language.

Chronological Advancement in Compiler Design: A Review



Fig. 1. Phases of Compiler.

*c. Semantic Analysis:* Semantic analysis takes the representation made from the analysis of syntax and semantic rules apply for representation to ensure that the program meets the requirements of semantic rules of the language.

*d. Code Generation:* This final stage of a typical compiler converts the intermediate representation of program into an executable set of instructions. This last stage is the only step in the compilation that is machine dependent. You can also do optimization at this stage of compilation that makes the program more efficient.

This paper is organized as follows: Section II provides the related work. In Section III, we briefly discuss the algorithm for compiler design. Section IV presents the compiler construction tools. Section V gives overview of the Compiler Project at Leading Computer Science Universities. Finally, Section VI provides the conclusion of this paper.

# 2 Related Work

Yunsik Son *et al.* [4] provided a brief overview on symbol table. Symbol table is an essential module in compiler construction. It includes phases like lexical analysis, syntax analysis, semantic analysis and code generation. In this paper, they deal with reverse technique for the verification of the symbol table in objective C compiler.

#### Amit Verma, Nikita Bakshi

They also discuss the design and implementation of a reverse translator that verifies and analyses the symbol table designed during the development stage objective C compiler. Furthermore, based on the symbol table verification, a correct code can be generated by examining the use of identifiers and attribute in the code generation step.

I. Budiselic *et al.* [5] discussed the experiences with the programming language instruction over the last three years and tool based assignment used before and quantitative differences in results. They also discuss the compiler design courses providing an overview on compiler project at seven different computer science universities in Europe and US. They also provide an overview on programming language translation courses, organization of PTL courses and describe the evolution of programming from its initial stage to its current design. Two important classifications of compiler design project and courses are explained. They roughly divide compiler design into two courses front end heavy and back end heavy.

Chengyong Wu et al. [6] presented an overview of the design of the main components of ORC, especially new features in the code generator. The Open Research Compiler (ORC) was jointly developed by the Intel Microprocessor Laboratory of Technology and the Institute of Computer Technical Academy of Science of China. It has become the leading open source compiler in processor family Itanium TM. ORC development methodology which is important for achieving the objectives is discussed. Performance comparison with other IPF compiler and a brief summary of research based on ORC are also presented.

John S. Mallozzi et al. [7] talked about one semester course in compiler design presents difficulties to an instructor who want to assign a project in which object oriented techniques are used. This paper describe a method that uses the tool developed by the author to generate a parser that encourages an object-oriented approach, clearly related code written by the student which automatically generate code with intended students to increase understanding.

Miodrag Djukic et al. [8] described a technique in which significance of controllability and speed is placed upward the retarget ability and cycle-accuracy to provide a better platform for software development. Many simulation instructional approaches place the retarget ability and cycle precision as the key functions to facilitate the exploration and performance of architecture and also estimate early in the development phase of hardware. The main idea of this work is to associate the simulator effort compiled with the development of the C and build target language compiler for the processor using knowledge related to compiler and reusing some common software elements.

Mirko Viroli [3] provided a brief description about EGO compiler (Extract Generic On-Demand). This is the result of a project developed in partnership with Sun Microsystems in order to evaluate a smooth support for generic time function, which does not require changes in the JVM or any other component of the Java Runtime Environment. We conceive and develop solution which is a sophisticated translation based on the type style step also known as reification of type parameters, where the type information is at runtime and automatically create as on per code and cached for future use. The main aspects of development are presented, from the basic design to implementation and deployment issues.

Johgheen Youn *et al.* [9] presented a new coding scheme and instructions based on the dynamic implied addressing mode (DIAM) to solve the limited space coding and side effects by trimming. Also introducing two versions of architectures to support our approach is based on DIAM. They also suggest a generation of code algorithm to fully utilize DIAM. In their work, architecture with DIAM exhibition shows code size reduction up to 8% and 18% on average speed compared with the basic architecture without DIAM.

Hankjin Lee *et al.* [10] provided a well established algorithm more over a methodology that is used for detection of design patter. In this paper, reclassification of GoF pattern takes place. Gang of Four (GoF) is known to be very useful for the detection of projects with reverse engineering methods. He also proposed GoF pattern detection technique. After that, evaluation of new technique is done and paper is concluded with the pros and cons of new approach, and what other work is to be done in terms of future research.

Ivan Keimek *et al.* [11] provided a brief description that reverse engineering is used in many fields of IT every day like binary code patching, legacy compatibility, network protocol analysis, malware analysis, rapid prototyping or in debugging. Despite its widespread use, reverse engineering is not actively taught as part of computer courses. This paper attempts to provide an overview of real life scenario of reverse engineering. Analysis of skills, ways of thinking that can be developed by reverse engineering and provides example that you can teach reverse engineering by resolution of practical problems. They also focus on the importance of reverse engineering as a tool to turn the self motivation in students and systematically build your logical thinking skills and analytical skills.

Cristina Cifuentes *et al.* [12] presented different type of reverse engineering based on level of code abstraction, which was used to reengineer assembly code, CASE code, machine code and source code. In this paper they elaborate various type of reverse engineering and protection for copyright software. Common uses of reverse engineering were explained. Comparative overview of the legal standing reverse engineering are provided. They also propose the existing and future challenges of the global electronic community for the protection of digital works.

# **3** Algorithm for Compiler Designing Process

Compiler design and related set of classic algorithms provides a pretty flexible software architecture that can be called "abstract machine" architecture. Sometimes using this architecture and adapting it to a particular task can make design more transparent and more easily debugged.

Amit Verma, Nikita Bakshi

```
a. Memory Aware Mapping for Compiler
SOS: Set Operations to Schedule
QRO: Query Ready to Schedule Operations
G: Application DDG
```

i secure code for memory Aware mappin		Pseudo	code	for	Memory	Aware	Mappin
---------------------------------------	--	--------	------	-----	--------	-------	--------

Begin Priorities Assign (G); p=Highest Priority // Minimum mobility while (SOS  $\neq \phi$ )  $\dot{Q}RO$  = queue ROP (p) do {  $O_{p} = dequeue QRO$  $(DDP_s, RTime) = Predecessors (O_n)$  $(IDDP_s) = Predecessors - R (O_p)$ do Choice=GetCost(DDP<sub>s</sub>, IDDP<sub>s</sub>, RTime); RTime++ } while ( Resources\_Cong (Choices)); Dscn = DecidesScheduleTime (Choices) RsvResources (Decision) Sch(Op)  $SOS = SOS - O_p$ } while (QRO  $\neq \phi$ ); p = p + 1End

The pseudo starts with DDG to represent the priority. DDG is initialized by setting the minimum value of mobility. QRO queue takes ROP function which has a value of mobility less than or equal to the value of variable p. After that do while loop schedule each operation once at a time until it become empty. DDPs set the earliest clock cycle at which operation Op can be schedule. The preprocessor returns Op where IDDPs were executed. Subsequently, GetCost return choice variables. After that RTime is incremented and the GetCost function is repeated until available Op are found. The decision schedule time function analyze the mapping costs from choice variable and select the most efficient operation.

#### b. Distributed Shared Memory Automatic Parallelizing Compiler

 $\begin{array}{l} T_a: \text{Target Array} \\ P_O: \text{Parallelizing Loop} \\ K_O: \text{Kernel Loop} \\ \text{DDM: Data Distributed Method} \\ \text{IK}_O: \text{Intraprocedural Kernel Loop} \\ \text{iK}_o: \text{Interprocedural Kernel Loop} \\ \text{FTC: First Touch Control} \end{array}$ 

104

Pseudo code for Distributed Shared Memory Automatic Parallelizing Compiler

```
Begin
Detect
          \rightarrow P_o and T_a
For
            DDM=Determination of the shape
            Determination of the IK<sub>o</sub>
            if
                        T_{a} \rightarrow Arg
                       return determination of iKo
            else
                       no change
            endif
end for DDM
      →check distribution
Err
if
            Code→ FTC
            Analysis of data redistribution
else
            B→ Analysis of reference pattern
            Return B
end if
End
```

In this pseudo code, we start with the detection of parallelizing loops and target arrays. Parallelizing loop for each shape of data distribution is determined. Kernel loop is distributed and data redistribution analysis is done. In the end, analysis of first touch control for data redistribution and reference pattern.

#### c. Demand Driven to Detect Parallelism in Irregular Code of Compiler

Pseudo Code For Demand Driven to Detect Parallelism in Irregular Code

```
Begin
Case1:
Check SCC (X<sub>1</sub>,....,X<sub>n</sub>)
when dependence is found
SCC(Y_1,\ldots,Y_m) —
                              \rightarrow SCC(X<sub>1</sub>,...,X<sub>n</sub>)

→ start classification

Ŀ
            if
            SCC is already in stack
            Return
            SCC(X_1,\ldots,X_n) SCC(Y_1,\ldots,Y_m)
            else
            B 🔶 Generate Error
            endif
Case2:
SCC(X_1)

    trival component

            inherit CLASS<sub>X1</sub>(e)
            X<sub>1</sub>=e
          if
          SCC(X_1,...,X_n) on trival component
          A CLASS<sub>XK</sub> (X_K(S_K)=e_K)
         Return - classified component
         else
         SCC← return unclassified
         endif
End
```

Amit Verma, Nikita Bakshi

In this pseudo code, non classified component is pushed into stack and process is started. If SCC is independent then its classification is found. Classification process starts when SCC ( $Y_1, \ldots, Y_m$ ). It reaches a deadlock when mutually dependent SCC exists in the loop. If dependence is found, stack is checked before starting classification. If already in stack, dependence exists. In case 2 if stack belong to the same class then component inherit otherwise remains unclassified.

# **4** Compiler Construction Tools

There are so many tools for compiler design; few of them are listed below [18] :

*i. Lex & Yacc:* Lex and Yacc are the most classic UNIX tools for the compiler construction. Lex does tokenization which helps to create programs whose control flow is handled by instances of regular expression in the input stream. Yacc provides a parsing tool to illustrate the input to a computer program. The Yac user specifies the grammar of the input with its code to be invoked as each structure in that grammar is renowned. Yacc provides specification into a subroutine to process the input.

*ii. Lemon:* The lemon program is an LALR parser generator. It takes a context free grammar and converts it into a subroutine that will parse a file using that grammar. Lemon is analogous to much more programs like "BISON" and "YACC", but the lemon is not companionable with either bison or yacc.

*iii. GCC-RTL:* RTL store text in a file as an interface between the language front end and GNUCC. GNUCC was designed to use RTL internally only.

*iv. ANTLR:* ANother Tool for Language Recognition. It is a powerful parser generator for processing, reading, executing, translating binary files or structure files. It's widely used to build tools, languages and frameworks.

v. *ML-RISC*: MLRISC is a customization optimization back-end written in Standard ML and has been successfully retargeted to multiple architectures like PPC, Sparc, Alpha, MIPS.

# 5 Overview of The Compiler Project at Leading Computer Science Universities

In this section, we briefly summarize the compiler design courses and accompanying programming assignments from some of the leading computer science universities in the US and Europe. The specifics of each course are shown in Table I.

urses in Some Computer Science Universities.	ImplementationLectureStudentProjectLanguageFront end %Toolsper groupImpact	Java 10 ANTLR 3–4 60	SML, Ocaml, 10 Lexer and 1–2 70 Haskell, Java parser and others generators	Ocalm 25 ocamllex, 5 40 ocamlyacc	Java 30 JLex, CUP 2 66	MIPS 30 Lexer and 1-2 50 parser	Java 30 JLex, CUP 1-2 40	Ocalm 15 ocamilex, N/A N/A
Properties of Compiler Courses in Sor	e Target Implement ge Language Langua;	x86-64 Java	X86-64 SML, Ocar Haskell, Ja and others	Student Ocalm Choice	X86 or Java similar	a COOL MIPS	MIPS Java	like Keiko/ARM Ocalm
Table 1. Key Prol	Course Source ss name Language	6.035 Decaf Computer Language	Engineering 15-411 L1-L4 Compiler Design	COMS W4115 Student Programming designed Languages and Translators	ch Compiler JavaLi Design I	CS 143 C++, Java Compilers	CS 164 COOL Programming Languages and Commilers	Compiler Oberon-like
	Universitie	MIT	CMU	Columbia	ETH Zuric	Stanford	Berkeley	Oxford

Chronological Advancement in Compiler Design: A Review

Amit Verma, Nikita Bakshi

#### 6 Conclusion

In this paper, we conclude that compiler is a program that translates a source-code written in programming-language (like C or Pascal) to an object-file. Afterwards a linker links the object-file with other object-files and libraries to make them executable (like COM or EXE). Many compilers will perform both the compiling and linking steps in one operation. A compiler must do much more checking about the legality of the statements, make calls to functions, import from libraries, and manage variables of different scopes and so on. Today, compiler design is a vast field of research. Every programming language need different compiler to run a program. So, we need generic compiler to make programmer's work easy. The current paper gives a prerequisite environment for the construction and design of an efficient generic compiler.

# References

- 1. Takashi Hirooka et al. "Automatic Data Distribution Method Using First Touch Control For Distributed Shared Memory Multiprocessor", pp. 147-161, Springer 2003.
- 2. Manuel Arenaz et al. "A Compiler Framework to Detect Parallelism in Irregular Code", pp. 306-320, Springer 2003.
- 3. Mirko Viroli "Effective and efficient compilation of run-time generics in Java", Electronic Notes in Theoretical Computer Science, vol. 2, pp. 95-116, 2005.
- 4. Son, Yunsik, Seman Oh, and Yangsun Lee, "A Reversing Technique for Symbol Table Verification on Compiler Constructions" 2014.
- 5. Budiselic, I., D. Skvore, and S. Srbljic "Designing the programming assignment for a university compiler design course", 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), IEEE, 2014.
- 6. Wu, Chengyong, et al. "An overview of the open research compiler", Languages and Compilers for High Performance Computing Springer Berlin Heidelberg, pp. 17-31, 2005.
- 7. Mallozzi, John S. "Thoughts on and tools for teaching compiler design", Journal of Computing Sciences in Colleges, vol. 21.2, pp. 177-184, Springer 2005
- 8. Djukic, Miodrag, et al. "An Approach to Instruction Set Compiled Simulator Development Based on a Target Processor C Compiler Back-End Design", First IEEE Eastern European Conference, IEEE, 2009.
- 9. Youn, Jonghee M., et al. "Two versions of architectures for dynamic implied addressing mode", Journal of Systems Architecture, vol. 8, pp. 368-383, 2010.Leupers, Rainer. "Compiler design issues for embedded processors", Design & Test of Computers, vol. 4, pp 51-58, IEEE, 2002.
- 10. Lee, Hakjin, Hyunsang Youn, and Seunghwa Lee, "Automatic detection of design pattern for reverse engineering", 5th ACIS International Conference on Software Engineering Research, Management & Applications, IEEE, 2007.
- 11. Ivan Klimek, Marián Keltika and František Jakab, "Reverse Engineering as an Education Tool in Computer Science", 9th IEEE International Conference on Emerging eLearning Technologies and Applications, pp. 123-126, IEEE2007.
- 12. Cifuentes, Cristina, and Anne Fitzgerald. "The legal status of reverse engineering of computer software", vol. 2, pp. 337-351, Springer, 2000.
- 13. Kim Ann Zimmerman June 04, 2012 ,"History of computing", Retrieval 21st May 2015, http://www.livescience.com/20718-computer-history.html.
- 14. Jeremy Meyers "A Brief History Of The Computers(B.C.-1993 A.D)", Retrieval 20 May 2015, http://www.jeremymeyers.com/comp.
- 15. Mary Bellis "History of Computers" Retrieval 15<sup>th</sup> May 2015, http://inventors.about.com/ library/blcoindex.htm.
- Scribd, Feb24,2012 "Computer History Table", Retrival 10<sup>th</sup> May 2015, http://www.scribd. com/doc/82649888/Computer-History-Table#scribd.
- 17. Vangie Beal "Compiler", Retrieval 22th May 2015, http://www.webopedia.com/TERM/C/ compiler.html
- Christopher B. Browne's "Compiler Construction Tools", Retrieval 22<sup>th</sup> May 2015, http:// linuxfinances.info/info/compilingtools.html

109

## **Editorial Board of the Volume**

Rafael Aschoff, IFPE - Federal Institute of Education, Science and Technology of Pernambuco, Brazil Kyller Gorgônio, Federal University of Campina Grande, Brazil Rafael Kunst, Federal University of Rio Grande do Sul (UFRGS), Brazil Cristian Machado, Federal University of Rio Grande do Sul, Brazil Ramachandran Venkatesan, Memorial University of Newfoundland, Canada John Pantoja, University of Los Andes, Colombia Jovana Zoroja, University of Zagreb, Croatia Anna Fagertun, Technical University of Denmark, Denmark Sherief Hashima, Engineering dept, Nuclear Research Center, EAEA, Cairo, Egypt Benoit Darties, Université de Bourgogne, France Moufida Maimour, Lorraine University, France Nikolaos Doukas, Hellenic Army Academy, Greece Malamati Louta, University of Western Macedonia, Greece Charalampos Pitas, National Technical University of Athens, Greece Hengky Susanto, Hong Kong University of Science and Technology, Hong Kong Navab Abid Ali Khan N, Vasavi College of Engineering, India Sonali Agarwal, Indian Institute of Information Technology, Allahabad, India Naveen Aggarwal, Panjab University, India Ms Alam, AMU, Aligarh, India K. Mani Anandkumar, Anna University, Chennai, India Mohammad Bazaz, National Institute of Technology Srinagar, India Manoj Bs, Indian Institute of Space Science and Technology, India Lakshmi Devasena C, IBS, Hyderabad, IFHE University, India Chinmay Chakraborty, Birla Institute of Technology, Mesra, India Monish Chatterjee, Asansol Engineering College, India Neetirajsinh Chhasatia, Charusat University, Changa, India Abhijit Das, Indian Institute of Technology, Guwahati, India Sujata Deshmukh, LTCOE, Mumbai University, India Sheetal Dhande, Amravati University, India Anuj Gupta, CGC College of Engineering, India Munish Jindal, Panjab Univ. Rural Centre, Kauni (Sri Muktsar Sahib), Punjab, India Namita Kale, University of Pune, India Ekram Khan, Aligarh Muslim University, India Mohammed Khan, Indian Institute of Technology, Hyderabad, India Naveen Kolla, Geethanjali Institute of Science and Technology Nellore, India Vaishali Kulkarni, NMIMS University, India Navin Kumar, Amrita University, India Saravanan Logan, Coimbatore Institute of Technology, India Ruchika Malhotra, Delhi Technological University, India T Manjunath, HKBK College of Engineering Bangalore Karnataka, India Asoke Nath, St. Xavier's College, India Ashalatha Nayak, Manipal University, India Swati Nikam, University Of Pune, India

Research in Computing Science 103 (2015)

Sanjaya Kumar Panda, Veer Surendra Sai University of Technology, Burla, India Vikas Pareek, Banasthali University, India Maulika Patel, G H Patel College of Engineering & Technology, India Sanjeev Patel, Jaypee Institute of Information Technology, India Ankita Pramanik, IIEST, Shibpur, India Sudhir Rupanagudi, WorldServe Education, India Debarshi Kumar Sanyal, KIIT University, Bhubaneswar, India Rajeshwar Singh, Punjab Technical University, Jalandhar, Punjab, India Sanjay Singh, Manipal Institute of Technology, India Anjini Tiwary, Birla Institute of Technology, Mesra, India Amarsinh Vidhate, University of Mumbai, India Sanjeev Wagh, University of Pune, India Hamid Reza Arkian, University of Guilan, Iran Habib Rasi, Shiraz University of Technology, Iran Raid Daoud, Hawija, Iraq Christian Callegari, University of Pisa, Italy Daniele Tarchi, University of Bologna, Italy Ali Bashir, Osaka University, Japan Alex James, Nazarbayev University, Kazakhstan Vu Van Hiep, International of Ulsan, Korea Rize Jin, KAIST, Korea Rong Ran, Ajou University, Korea Jalil Desa, Telekom Research and Development, Malaysia Nik Azlina Nik Ahmad, Universiti Kuala Lumpur, Malaysia Go Yun II, Heriot-Watt University Malaysia, Malaysia Mohammed Bouhorma, Faculté des Sciences et Techniques de Tanger, Morocco Ji Changpeng, Liaoning technical university, P.R. China Liang Huang, Zhejiang University of Technology, P.R. China Xin Huang, Xi'an Jiaotong-Liverpool University, P.R. China Datong Liu, Harbin Institute of Technology, P.R. China Xin Liu, China University of Petroleum, P.R. China Sive Wang, Institute of Information Engineering, Chinese Academy of Sciences, P.R. China Zhenyu Zhou, North China Electric Power University, P.R. China Fawaz Bokhari, PUCIT - University of the Punjab, Pakistan Shahid Manzoor, UCSI Universiti Kuala Lumpur, Pakistan Slawomir Ambroziak, Gdansk University of Technology, Poland Marcin Kowalczyk, Warsaw University of Technology, Poland Zbigniew Leonowicz, Wroclaw University of Technology, Poland Filipe Caldeira, University of Coimbra, Portugal Sudhir Routray, Instituto de Telecomunicações, Portugal Vasile Bota, Technical University of Cluj Napoca, Romania Lucian Perisoara, University Politehnica of Bucharest, Romania George Suciu, Politehnica University of Bucharest, Romania Belal Abuhaija, University of Tabuk, Saudi Arabia Haikal El Abed, Technical Trainers College (TTC), Saudi Arabia

Wajeb Gharibi, Jazan University, KSA, Saudi Arabia

Editorial Committee

Nazeeruddin Mohammad, Prince Mohammad Bin Fahd University, Saudi Arabia Al-Sakib Khan Pathan, Islamic University in Madinah, Saudi Arabia Arun Kumar, Nanyang Technological University, Singapore Francisco Bellido Outeiriño, University of Córdoba, Spain Luis Sequeira, University of Zaragoza, Spain Hsing-Chung Chen, ASIA University, Taiwan Y-Chuang Chen, Ming Hsin University of Science and Technology, Taiwan Chi-Ming Wong, Jinwen University of Science and Technology, Taiwan Zribi Amin, Higher Institute of Communication Technology, Tunisia Ghilen Aymen, University of Gafsa, Tunisia Pınar Kırcı, İstanbul University, Turkey Sharmini Enoch, University of Wollongong in Dubai, UAE Mohamed Hassan, American University of Sharjah, UAE Deepayan Bhowmik, Heriot-Watt University, United Kingdom Kezhi Li, Imperial College London, United Kingdom Cathryn Peoples, University of Ulster, United Kingdom Basit Qureshi, University of Bradford, United Kingdom Edward Jung, Kennesaw State University - Marietta Campus, USA Feng Lin, University at Buffalo, SUNY, USA Junjie Lu, Broadcom Corp., USA Mehrgan Mostowfi, University of Northern Colorado, USA Shuang Wang, University of California, San Diego, USA Mohammad Zahidul Bhuiyan, Finland, Finland Santosh Majhi, Veer Surendra Sai University of Technology, India Dhiraj Sunehra, Jawaharlal Nehru Technological University Hyderabad, India Thuan Do, HoChiMinh City University of Technology and Education, Vietnam A. F. M. Sajidul Qadir, Samsung R&D Institute-Bangladesh, Bangladesh Philipp Glatz, Ifm Electronic, Germany Alin Ananyase, Robert Bosch(RBEI), India Ajay Bhushan, MahamayaTechnical University Noida, INDIA, India Joy Bose, Samsung R&D Institute India, Bangalore, India Pethuru Raj Chelliah, IBM, India Senthil Kumar Mani, Imagination Technologies, India Arindam Pal, TCS Research, India Arun Kumar Sp, IBM, India Mukesh Taneja, Cisco Systems, India S Zafaruddin, Ikanos Communications, inc., India Stylianos Basagiannis, United Technologies Research Centre, Ireland Riccardo Brama, CMC Labs, Italy Scott Trent, IBM Research - Tokyo, Japan Chia-Yu Lin, National Taiwan University, Taiwan Sebastian Robitzsch, InterDigital Europe, United Kingdom Stephen Chasko, Landis + Gyr, USA Qunfeng He, Qualcomm Technologies Inc., USA Songqing Zhao, Apple Inc., USA Gustavo Fernández Domínguez, AIT Austrian Institute of Technology, Austria

Sofiane Maza, University of Bordj Bou Arreridj, Algeria

Research in Computing Science 103 (2015)

Zohair Abu-Shaban, The Australian National University, Australia Nafees Mansoor, Universiti Teknology Malaysia, Bangladesh Carlos Astudillo, State University of Campinas, Brazil Adilson Chinatto, Universidade de Campinas, Brazil Amin Banitalebi-Dehkordi, University of British Columbia, Canada Veria Havary-Nassab, University of Toronto, Canada Tejinder Singh, University of Waterloo, Canada Navikkumar Modi, CentraleSupelec/IETR, France Abbass Nasser, Université de Bretgane Occidentale, France Ali Haqiqatnejad, University of Isfahan, Iran Mehdi Mohammadkhani, Payame Noor University, Tehran, IRAN, Iran Afshin Shaabany, University of Fasa, Iran Atieh Rajabi Khamesi, University of Padova, Italy Saleem Shahid, Politecnico di Milano, Italy Fadhil Taher Alawe, Universiti Kebangsaan Malaysia, Malaysia Mangging Guo, Beijing Unicersity of Posts and Telecommunications, P.R. China Mu-Qing Lin, Northeastern University, P.R. China Ting Wang, Hong Kong University of Science and Technology, P.R. China Umair Rafique, Mohammad Ali Jinnah University, Pakistan Firooz Saghezchi, Instituto de Telecomunicações, Portugal Pratya Nuankaew, Mae Fah Luang University, Thailand Abdullah Almuttiri, De Montfort University, United Kingdom Mithileysh Sathiyanarayanan, University of Brighton, United Kingdom Iman Tavakkolnia, University of Edinburgh, United Kingdom Shiqiang Wang, Imperial College London, United Kingdom Awais Khawar, Virginia Tech, USA Jian Lin, Georgia Institute of Technology, USA Javed Mohammed, Newyork Institute of Technology, USA Quamar Niyaz, University of Toledo, USA

Impreso en los Talleres Gráficos de la Dirección de Publicaciones del Instituto Politécnico Nacional Tresguerras 27, Centro Histórico, México, D.F. Octubre de 2015 Printing 500 / Edición 500 ejemplares