From the Editor’s Desk

Increasingly, we are dependent on information that is available only by accessing a network. This dependency implies that users would need access to this information even while on the move. Hence, mobility is an additional parameter that needs to be considered in the design of networks, network protocols, applications and information services. When moving from fixed-wired database technology to architectures including mobile clients, mobile devices face unforeseeable disconnections and lose connections to mobile clients. They are likely to have scarce resources, and they are required to react to frequent changes in the environment. To accommodate these new requirements imposed by mobility, middleware platforms for mobile computing must be capable of both deployment-time configurability and run-time reconfigurability. The paper titled “Performance Comparison of Mobile Computing Reflective Context Aware Middleware in Adhoc Smart space with Cross Layering Approach” authored by N.Radhika and Dr.S.Arumugam throws more light on importance of context aware middleware. Though middleware meets the requirements imposed by mobility it does not solve the traffic grooming problem. This problem is handled by graph models that are developed for multihop logical topologies. Authors Dr.B.Sathyanarayana and Dr.G.Narasimhan in their research work titled “Modified De Bruijn Graphs as Regular Logical Topologies in Fiber Optic Networks” have proposed a shortest path routing algorithm which takes smaller hop length than what the Bruijin graphs achieve. Mobile/Portable applications needs a system which should provide significant noise cancellation, fast algorithmic convergence in colored noises, short group delay, and minimal introduction of artifacts into the speech signal. Furthermore, it should have low computational cost and complexity, low memory usage, low power requirements, and small physical size. These issues are especially important in portable applications, where processing power must be conserved. As a result, subband adaptive filters (SAF) become a viable option for many adaptive systems. G.A.E.Sathish Kumar, Dr.P.Chandrasekhar Reddy and Dr.B.Stephen Charles in their paper “Design Criteria for Subband Adaptive System” have proposed three design criteria for subband adaptive modelling. Any information sent in the network is subjected to possible security and privacy attacks. Most of these problems can be solved relatively easily by means of standard encryption and digital signature techniques. In recent years many researchers are focusing on developing MDA transformation techniques for security services. The paper “Design and Development of MDA Transformation Techniques for Security Services in Embedded Environment” authored by G.R.Karpagam and S.N.Sivanandam portray a transformation technique for security services, which provide a comprehensive solution to the contemporary problems viz., inability to achieve reusability and interoperability, and tool migration in the software industry.
Although computing power and network bandwidth have increased dramatically in recent years, the design and implementation of networked applications remains expensive and error-prone. Much of the cost and effort stems from the continual re-discovery and re-invention of core patterns and framework components throughout the software industry. Systematic software reuse is a promising means to reduce development cycle time and cost, improve software quality, and leverage existing effort by constructing and applying multi-use assets like architectures, patterns, components, and frameworks. The authors U. Suman and Dr. M. Ingle’s topic, “Towards Development of Quality Software using Reuse Reengineering Approach”, a twofold software reuse process produces a high quality and flexible software architecture in a timely and cost effective way. Spatial data mining is the process of discovering interesting and previously unknown, but potentially useful patterns from large spatial datasets. Extracting interesting and useful patterns from spatial datasets is more difficult than extracting the corresponding patterns from traditional numeric and categorical data due to the complexity of spatial data types, spatial relationships, and spatial autocorrelation. Much more complexity is 3D data modeling and 3D data management in mobile.

Authors N. Subash Chandra and Dr. A. Govardhan in their paper titled “Three Dimensional (3D) Data Types for Spatial Abstract Data Model” have described the complex inherent features of 3D data types in data model.

Genetic sensorineural hearing loss includes a broad range of disorders that impact infants, children, and adults who may have unilateral or bilateral hearing loss that ranges from mild to profound. Several investigations on speech intelligibility have demonstrated that subjects with sensorineural loss need 5 to 15db higher SNR (Signal to Noise Ratio) than the normal hearing subjects. Today’s Digital Hearing Aids are not up to the expectation for sensorineural loss patients. Hearing-impaired patients applying for hearing aid reveal that more than 50% are due to sensorineural loss. The paper “FFT-based DCT-LMS Speech Enhancement for Sensorineural Loss Patients” authored by S. L. Sunitha and V. Udayasankar describes a FFT based DCT-LMS algorithm to improve the SNR and to reduce the convergence rate of the Least Means Square (LMS) for sensorineural loss patients. If protein structure, even secondary structure, can be accurately predicted from the now abundantly available gene and protein sequences, such sequences become immensely more valuable for the understanding of drug-design, the genetic basis of disease, the role of protein structure in its enzymatic, structural, and signal transduction functions, and basic physiology from molecular to cellular, to fully systemic levels. In short, the solution of the protein structure prediction problem (and the related protein folding problem) will bring on the second phase of the molecular biology revolution. This fact has stimulated the authors K. Saravanan and T. Sivaraman in creating a software
tool for predicting of secondary structures of proteins. Their work “PRISM: A Software for the prediction of secondary structures in protein” is an innovative effort in the field of Bioinformatics.

A major goal of the manufacturing industry is increasing product quality. Hence much research is focused on tool wear prediction systems and fault detection systems. Integrating a tool condition monitoring system within the machine allows online, real-time monitoring to reduce the dependence on human judgment. Among the many possible machining conditions that could be monitored, tool wear is the most critical for ensuring uninterrupted machining. This triggered the authors V. Kalaiselvi, D. Sivakumar and R. Karthikeyan in developing a monitoring system for tool wear prediction and about which they have described in their paper “Adaptive Network Based Fuzzy Inference System for Tool Wear Prediction”. The basic objective of a fault detection methodology for dynamic systems is to provide techniques for detection and isolation of failed components in an attempt to prevent catastrophic malfunctions in the system liable to cause destruction of the equipment, endangering the crew or the operational personnel, or in any way constituting a potential threat to the safety or the natural environment. Efficient detection algorithm that can be used to alert an operator to any changes in steady-state characteristics of a monitored system. Authors S. Abraham Lincon, D. Sivakumar and J. Prakash who are doing research in fault detection methodology have conducted a simulated and presented the result of the study in their research paper “Simulation Studies On State and Bias Estimation of Continuous Stirred Tank Reactor using Augmented State Kalman Filter”

S.N. SIVANANDAM
<table>
<thead>
<tr>
<th>CONTENTS</th>
<th>PAGE NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design Criteria for Subband Adaptive Systems</td>
<td>92</td>
</tr>
<tr>
<td>G.A.E. Satish Kumar, Dr. P. Chandrasekhar Reddy, Dr. B. Stephen Charles</td>
<td></td>
</tr>
<tr>
<td>Design and Development of MDA Transformation techniques for security services in Embedded environment</td>
<td>99</td>
</tr>
<tr>
<td>S.N. Sivanandam, G.R. Karpagam</td>
<td></td>
</tr>
<tr>
<td>Modified De Bruijn graphs as Regular Logical Topologies in fiber Optic Networks</td>
<td>109</td>
</tr>
<tr>
<td>Dr. B. Sathyanarayana, Dr. G. Narasimhan</td>
<td></td>
</tr>
<tr>
<td>Performance comparison of Mobile reflective Context aware middleware in Adhoc Smart space with Cross layering Approach</td>
<td>116</td>
</tr>
<tr>
<td>N. Rathika, Dr. S. Arumugam</td>
<td></td>
</tr>
<tr>
<td>Adaptive - Network - Based Fuzzy Inference System for Tool Wear Prediction</td>
<td>125</td>
</tr>
<tr>
<td>V. Kalaichelvi, D. Sivakumar, R. Karthikeyan</td>
<td></td>
</tr>
<tr>
<td>Towards Development of Quality Softwares using Reuse Reengineering Approach</td>
<td>135</td>
</tr>
<tr>
<td>U. Suman, Dr. M. Ingle</td>
<td></td>
</tr>
<tr>
<td>FFT based DCT-LMS Speech Enhancement for Sensorineural Loss Patients</td>
<td>146</td>
</tr>
<tr>
<td>Sunitha S L, Dr. V. Udayashankara</td>
<td></td>
</tr>
<tr>
<td>Simulation Studies On State And Bias Estimation Of Continuous Stirred Tank Reactor Using Augmented State Kalman Filter</td>
<td>155</td>
</tr>
<tr>
<td>S. Abraham Lincon, D. Sivakumar, J. Prakash</td>
<td></td>
</tr>
<tr>
<td>PRISM : A software tool for the prediction of secondary structures in proteins</td>
<td>161</td>
</tr>
<tr>
<td>K. Saravanan, T. Sivaraman</td>
<td></td>
</tr>
<tr>
<td>Three Dimensional (3D) Data Types for Spatial Abstract Data Model</td>
<td>168</td>
</tr>
<tr>
<td>N. Subhash Chandra, Dr. A. Govardhan</td>
<td></td>
</tr>
</tbody>
</table>
Design Criteria for Subband Adaptive Systems

G.A.E. Satish Kumar¹ Dr. P. Chandrasekhar Reddy² Dr. B. Stephen Charles³

Abstract
Multirate schemes such as the subband adaptive filter have been a topic of interest for many years now. They are used to identify high-order FIR systems and are a good alternative to the classical LMS algorithm. LMS adaptive filter is less attractive to this type of applications because it has a larger complexity and its convergence behavior is normally worse. Three design criteria for subband adaptive systems, which deal with frequency selectivity, perfect reconstruction and perfect path modelling, are presented in this paper. It will be proven that the design criteria on perfect reconstruction and perfect path modelling for the subband adaptive filter are fulfilled for the Partitioned Block Frequency-Domain Adaptive Filter algorithm.

Keywords: Subband Adaptive Filter, DFT Modulated Filter Banks, Design Criteria, Partitioned Block Frequency-Domain Adaptive Filter

1. Introduction
Subband Adaptive Systems
Subband adaptive filtering schemes have been a topic of interest for many years now [1]. They are used to identify long FIR systems because they provide reduced complexity solution for high-order problems and in this way do better than the standard fullband adaptive approaches. A general subband adaptive filtering system is shown in figure 1. Both input signals x and d are fed into identical M-band analysis filter banks, with d being a filtered version of x by the unknown system w[k]. In most applications a near-end signal s is added to wx such that d = s + wx. The aim is to suppress wx at the output e and to retain a non-distorted version of s. After N-fold subsampling, adaptive filtering is done in each subband separately. Basically any kind of adaptation algorithm can be used for the update. It is however common to use (N) LMS to adapt [11], the subband filters fm[k]. It is observed that in contrast to classical adaptive filtering structures [13], this setup w[k] is estimated using a set of parallel, independently adapted filters fm[k]. The outputs of the subband adaptive filters are recombined in the synthesis filter bank leading to the final output e. The ideal frequency amplitude characteristics of the analysis filters $H_m(z)$ and synthesis filters $G_m(z)$ are also shown (ideal bandpass filters).

As a cheaper alternative to the LMS algorithm, the Frequency-Domain Adaptive Filter (FDAF) was introduced, being a direct translation of Block-LMS to the frequency domain [1]. Instead of linear convolutions and correlations component wise operations are performed in the frequency domain. The performance and the convergence properties of the FDAF algorithm are comparable to those of the LMS algorithm. It appears that the FDAF algorithm is computationally attractive.
only if the block length $L$ has the same order of magnitude as the filter length $L_{fb}$. In practice, however, this leads to unacceptable input/output delays. Partitioning may then be used to reduce the overall delay. This leads to the so-called Partitioned Block Frequency Domain Adaptive Filter [12] [15] [16].

2. Design Criteria

In this paper, three design criteria for subband adaptive systems are formulated, which should produce subband adaptive filters exhibiting better properties, similar with frequency-domain techniques [8]. Three design criteria are defined. Any analysis/synthesis filter bank set must be designed such that these design criteria are met. First of all, the filter banks have to be frequency selective. The second condition is about perfect reconstruction and the third requirement deals with perfect path modelling. These conditions are necessary requirements to ensure satisfactory performance of the subband adaptive filter. In the following sections each of these conditions are discussed in more detail.

2.1 Frequency Selectivity

To reduce the amount of aliasing components that are occurred in the subbands the analysis filters should be frequency selective. The residual error, which occurs in any standard subband adaptive setup basically consists of four parts: (1) an excess mean-squared error because of near-end source activity, (2) an excess mean-squared error due to aliasing components, (3) an undermodelling error and (4) lag error as the adaptive system fails to track time-varying setups and will lag behind.

The frequency characteristics of the synthesis bank do not have an influence on the convergence of the subband adaptive filters (see figure 1). Highly frequency selective filter banks initiate a considerable processing delay. This puts a constraint on the frequency selectivity, and hence on the downsampling factor $N$, makes subband schemes unattractive. On the other hand, computational savings appear to be more or less proportional to $N$.

2.2. Perfect Reconstruction

Both signal $x$ and $d = s + w \times x$ pass through the filter bank system (see figure 1). Ideally, signal $x$ is completely suppressed by the adaptive filtering operations. The desired part $s$ of signal $d$ passes to the output of the subband scheme without distortion, which is an exact (delayed) copy of $s$. Perfect reconstruction filter banks are used to minimize the distortion of $s$ (nearly). It is observed that overall aliasing cancellation does not keep out aliasing insertion in the individual subbands, which leads to degraded convergence behavior of the adaptive filters. Perfect reconstruction is obtained when

$$JG^T(z)H(z) = z^{-\delta}I_N, \delta \in IN$$

(1)

The smallest $\delta$ is determined such that Eq. 1 is fulfilled such that both the analysis and synthesis filters are causal and hence implementable. It can be shown that if Eq. 1 holds the perfect reconstruction conditions Eq. 2 and Eq. 3 are fulfilled for

$$C(z)B(z) = z^{-\delta}I_N, \delta \in IN$$

(2)

The design of critically downsampld perfect reconstruction filter banks is well-known and is treated in the literature [2] [3] [4] [14]. However, these maximally decimated filter banks are not suitable for subband adaptive filtering applications. So, oversampled subband systems should be applied. Some references on perfect reconstruction design for oversampled DFT modulated filter banks can be found in more recent publications [5] [6] [7] [10]. Condition 1 can be relaxed to

$$JG^T(z)H(z) \approx z^{-\delta}I_N, \delta \in IN$$

(3)
leading to nearly perfect reconstruction up to a delay.

The approximation in Eq. 3 implies that

$$V = \| JG^T (z) H(z) - z^{-\delta} I_N \|_2$$

(4)

is small.

As a result, the amplitude, phase and aliasing distortion will be small. In the same way condition 2 can be relaxed to

$$C(z) B(z) = z^{-\delta} I_N, \delta \in IN$$

(5)

Finally, adaptive filtering operations should be done in each subband. Perfect reconstruction will only hold for the desired part s of d (upper branch of figure 1), as the adaptive filters in between the analysis and synthesis filter bank (lower branch of figure 1) do not protect perfect reconstruction.

2.3. Perfect Path Modelling

To guarantee appropriate time-invariant system modelling, the perfect reconstruction or nearly perfect reconstruction, another constraint has to be imposed on the filter banks. It is known from the literature [2] that the lower branch can only model a time invariant system if

$$JG^T (z) \text{diag} \left\{ F_i(z) \right\} H(z) =$$

$$\begin{bmatrix}
  \hat{W}_0(z) & \hat{W}_1(z) & \cdots & \hat{W}_{N-1}(z) \\
  z^{-1} \hat{W}_{N-1}(z) & \hat{W}_0(z) & \cdots & \hat{W}_{N-2}(z) \\
  \vdots & \ddots & \ddots & \vdots \\
  z^{-1} \hat{W}_1(z) & \cdots & z^{-1} \hat{W}_{N-2}(z) & \hat{W}_0(z) \\
\end{bmatrix}$$

(6)

is a pseudo-circulant matrix.

In case of DFT modulated filter banks Eq. 6 leading to

$$C(z) F^{-1} \text{diag} \left\{ F_i(z) \right\} FB(z) =$$

$$\begin{bmatrix}
  \hat{W}_0(z) & \hat{W}_1(z) & \cdots & \hat{W}_{N-1}(z) \\
  z^{-1} \hat{W}_{N-1}(z) & \hat{W}_0(z) & \cdots & \hat{W}_{N-2}(z) \\
  \vdots & \ddots & \ddots & \vdots \\
  z^{-1} \hat{W}_1(z) & \cdots & z^{-1} \hat{W}_{N-2}(z) & \hat{W}_0(z) \\
\end{bmatrix}$$

(7)

is a pseudo-circulant.

Theorem: It is observed that if Eq. 6 or 7 is fulfilled the following time-invariant path can be modeled

$$W(z) = z^{-(N-1)-1} \sum_{n=0}^{N-1} z^{-n} \hat{W}_n(z)$$

(8)

Proof: The proof can be found in [2]. Observe that Eq. 8 is an Nth order polyphase decomposition of \( \hat{W}(z) \).

For complete modelling of an unknown system \( W(z) \) using a set of subband adaptive filters \( F_i(z) \) with the help of equation 6, it should be fulfilled and the order of \( \hat{W}(z) \) has to be sufficiently high to model \( W(z) \). In general, condition 6 will only be satisfied if the adaptive filters \( F_i(z) \) are infinitely long. It is not clear however which constraints [9] have to be imposed on the filter bank polyphase matrices \( \{H(z),G(z)\} \) such that any finite-order system \( W(z) \) can be modelled with finite-length filters \( F_i(z) \) following equation 6. For the frequency-domain adaptive filter Eq. 7 is fulfilled for finite-length filters \( F_i(z) \).

3. Partitioned Block Frequency-Domain Adaptive Filter: Design Criteria

Three design criteria on frequency selectivity, perfect reconstruction and perfect path modelling were defined for the subband adaptive filter. These conditions are necessary requirement for the satisfactory performance of the subband adaptive filter. It will be verified whether these conditions are fulfilled for the Partitioned Block Frequency-Domain Adaptive Filter.
The first criterion requires that the filter banks are frequency selective. A frequency selective analysis filter bank reduces the amount of aliasing components inserted in the subbands. Aliasing is known to have a negative impact on the convergence of the adaptive filter. The Partitioned Block Frequency-Domain Adaptive Filter uses poor filter banks with a sinc-like frequency amplitude response. So, it can be concluded that the first criterion is not fulfilled for the Partitioned Block Frequency-Domain Adaptive Filter, although the algorithm doesn’t seem to suffer from slow convergence due to aliasing.

The analysis prototype polyphase matrix $B(z)$, which characterizes an $M$-band $L$-fold downsampled DFT modulated analysis filter bank, is a structured matrix. By seeing figure 2, It is observed that for the Partitioned Block Frequency-Domain Adaptive Filter

$$
B(z) = \frac{Mz}{Mz - 1} \begin{bmatrix}
I_L & \cdots & I_L \\
\vdots & \ddots & \vdots \\
-I_L & \cdots & -I_L
\end{bmatrix}
$$

(9)

From figure 2, for the synthesis part it is observed that the synthesis prototype polyphase matrix $C(z)$ is given by

$$
C(z) = \frac{Lz}{Lz - 1} \begin{bmatrix}
I_L & 0 & \cdots & 0 & 0 \\
L & 0 & \cdots & 0 & 0 
\end{bmatrix}
$$

(10)

By filling the equations 9 and 10 in equation 2, it is easily verified that the perfect reconstruction condition is fulfilled for the Partitioned Block Frequency-Domain Adaptive Filter:

$$
\begin{bmatrix}
I_L & 0 & \cdots & 0 \\
\vdots & \ddots & \vdots & \vdots \\
-I_L & \cdots & -I_L
\end{bmatrix} = I_L
$$

(11)

As very simple filter banks are used no extra algorithmic delay is introduced by the filter banks. For the time-invariance condition ensuring perfect path modelling equation 7 has to be satisfied. Equations 9 and 10 are therefore filled in equation 7.

4. CONCLUSION

The identification of high-order FIR systems requires efficient adaptive algorithms such as the subband adaptive filter, which do better than standard fullband approaches both from a complexity and a performance point of view. In this paper three design criteria for appropriate subband modelling were presented. They are necessary requirements to ensure satisfactory performance of the subband adaptive filter. It can be concluded that the Partitioned Block Frequency-Domain Adaptive Filter fulfils the above mentioned conditions of perfect reconstruction and perfect path modelling. The first criterion requiring frequency selective filter banks is not fulfilled. Nevertheless, the Partitioned Block Frequency-Domain Adaptive Filter shows fast convergence.

REFERENCES


Design Criteria for Subband Adaptive Systems


Figure 1: General subband adaptive filter with ideal filter banks

Figure 2. Partitioned Block Frequency-Domain Adaptive Filter as an oversampled DFT modulated subband system: intermediate scheme
Author’s Biography

**G.A.E. Satish Kumar** was born in Andhra Pradesh, South India in 1971. He received the B.Tech. Degree in Electronics & Communication Engineering from Sri Krishnadevaraya University and M.E. Degree in Communication Systems from Gulburga University in 1995 and 1999 respectively and he is pursuing his Ph.D in Signal Processing in Jawaharlal Nehru Technological University, Hyderabad.

From 1995 to 2000 He was a Lecturer and from 2000 to 2005 as an Assistant Professor at R.G.M. College of Engineering, Nandyal, A.P. respectively. Now, working as Professor and Head of the Department in ECE at St. Johns College of Engineering & Technology, Yemmiganur, A.P. His research interest includes Signal & Image Processing, Filter Bank Design and Speech Processing. He published 3 Papers in International Conferences, 2 Papers in Journals.

**Dr. P. Chandra Sekhar Reddy** was born on 15-08-1966 at Chinnamanchupalle, Kadapa, Andhra Pradesh, South India. He did his B.Tech (ECE) from JNTU, Anantapur in the year 1983, M.Tech Applied Electronics from Bharathiar University, Coimbatore, Tamilnadu. He did his Ph.D. from JNTU, Anantapur in the year 2001 on “Routing in Adhoc Networks”.

He worked as Trainee at ISRO for 10 months. He has a total of 15 years of experience. Currently he is Professor in ECE Department at JNTU, Hyderabad. He has published 4 papers in Journals, 12 in Conferences and he has also written 1 text book.

**Dr. B. Stephen Charles** was born in Andhra Pradesh, South India in 1965. He received the B.Tech. Degree in Electronics & Communication Engineering from Nagarjuna University in 1986, M.E. Degree in Applied Electronics from Bharatir University in 1992 and Ph.D Degree in Digital Signal Processing from Jawaharlal Nehru Technological University, Hyderabad in 2001.

His research interest includes Adaptive Signal Processing, Wavelet Based Image Compression and Filter Bank Design. He has 20 years of teaching experience in various Engineering Colleges at various levels. He published 27 Research Papers (7 in Journals, 10 in International Conferences and 10 in National Conferences).
Design and Development of MDA Transformation techniques for security services in Embedded environment

S.N.Sivanandam¹  G.R.Karpagam²

ABSTRACT
The intention of the Model Driven Architecture initiative is to automate the generation of platform specific models (PSM) from platform independent models (PIM) and code from PSMs, and to ensure synchronization between models. These activities, together with other model engineering activities, such as, refactoring and pattern application, can be described in terms of model transformations. A model transformation takes one or more source models as input and produces one or more target models as output, following a set of rules. This paper deals with design of MDA based design of transformation techniques for security in embedded Environment. It proposes three types of transformation techniques namely Platform Independent Model in XML to PSM, PIM in UML to PSM, and Design of Entity Transformation Language. These approaches aim at standardizing the process of model transformation and portability of models across different platforms.

Keywords: Model Driven Architecture (MDA), Platform Independent Model (PIM), Platform Specific Model (PSM), and Model Transformation

1. INTRODUCTION
MODEL Driven Architecture (MDA) is an approach to IT system specification that separates the specification of system functionality, from the specification of the implementation of that functionality, on a specific technology platform [1]. The MDA treats models as proper artifacts during software development and model transformation process. There is a growing need for model transformation languages and tools that would be highly acceptable by users. This paper proposes three different approaches for model transformation. These approaches aim at standardizing the process of model transformation and portability of models across different platforms.

2. MODEL DRIVEN ARCHITECTURE
Models in MDA are of two types – Platform Independent Model (PIM) and Platform Specific Model (PSM). PIM describes a platform independent specification of the functionality of the application. PSM defines a platform specific model of the system incorporating technology-specific details. It is derived from PIM by applying platform specific translation rules.

Key standards in MDA are UML, XMI, MOF and CWM. UML (Unified Modeling Language) has become the language of choice for representing software designs and architectures. XMI (XML Metadata Interchange) is an XML-based representation of the UML models. Saving in XMI provides for vendor interoperability of the models. CWM (Common Warehouse Meta-model) is a data warehouse standard that address entire lifecycle of
3. RELATED WORK

In the review of the different OMG QVT RFP [3], [7] submissions, Gardner et al. proposed a unified terminology to enable a comparison of the different proposals. It focused on the 8 initial QVT submissions and discusses model queries, views, and transformations, whereas we focus on transformations methodology in more detail.

In addition to providing the basic unifying terminology, Gardner et al. described practical requirements on model transformations such as requirements scalability, simplicity, ease of adoption and the need to handle transformation scenarios of different complexities, such as transformations with different origin relationships between source and target model elements. Finally, they make some recommendations for the final QVT standard.

In particular, they recommend a hybrid approach, supporting declarative specification of simpler transformations, but also allowing for an imperative implementation of more complex ones [5].

The paper by Krzysztof Czarnecki and Simon Helsen proposes a possible taxonomy for the classification of several existing and proposed model transformation approaches. The taxonomy is described with a feature model that makes the different design choices for model transformations explicit. Based on the analysis, they propose a few major categories in which most model transformation approaches fit [4]. Embedded systems are constrained by the environments they operate in, and by the resources, they possess. For such systems, there exist several factors like security against software, side-channel attacks, computational demands, tradeoffs between factors such as security, cost and performance, operations under stringent resource constraints, rapid evolution of security mechanisms, standards etc. This requires moving security considerations from a function centric perspective into system architecture.

The work of Thomas Wollinger introduces the basic concepts, characteristics, and goals of various cryptographic algorithms. It is shown that embedded systems are essential parts of most communications systems and how this makes them especially attractive as a potential platform to implement cryptographic algorithms. Furthermore, although a challenging task, previous implementations of arithmetic intensive cryptographic algorithms seem to indicate that they can achieve acceptable performance on embedded processors and constrained platforms. Thus, designing and implementing efficient cryptographic algorithms on embedded systems will continue to be an active research area [22].

4 SCADA EMBEDDED SYSTEMS (SUPERVISORY CONTROL AND DATA ACQUISITION)

SCADA is an Embedded system and a supervisory level software package that is positioned on top of hardware to which it is interfaced, through Programmable Logic controllers. In SCADA system, client layer interacts with user and data server layer handles process data control activities. The data servers communicate with devices through process controllers. Data servers are connected to each other and to client stations through network. SCADA makes use of multi-tasking and real-time database located in one or more servers. Servers are responsible for acquisition, alarm checking, calculations, logging and archiving.
The possible methods of attacks on SCADA systems include Eavesdropping, Masquerading, Tampering and replaying. Eavesdropping is obtaining copies of messages without authority like learning the information from the data server like substation ids, login information of SCADA or near by terminals. Masquerading is Sending/receiving messages using identity of another without their authority. Tampering is Intercepting messages and altering their contents before passing them to intended recipient, trying to modify corporate data, process set points in SCADA server for malicious purposes. Replaying is Storing messages and sending them later. Impersonation is passing information to a person who poses as intended recipient. Spoofing is pretending as a user, or a computer identifying itself as a site when it isn’t. Cryptographic algorithms that are in existence for SCADA security are 3DES, AES; bit length of 128 minimum for encryption, RSA (1024 bit minimum), ECDSA (160 bit minimum) for Digital Signing and SHA-1, Key exchange for Integrity hashing [20].

5 EXPERIENCES WITH APPLYING MDA FOR SECURITY SERVICES

SCADA Embedded system is taken as a case study to prove the applicability of MDA in Cryptographic security evolution [17]. The attacks and possible solutions for attack were analyzed [20]. Attacks are tabulated in Table 1. MDA tool Generic Modeling Environment (GME) [6] is used for development of the models. SCADA environment and its security services are represented in the Metamodel (Meta PIM, Meta PSM). Transformation is achieved by developing an interpreter. Figure 1 depicts the phases of the proposed scheme [19].

<table>
<thead>
<tr>
<th>Attack</th>
<th>Impact of Attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Denial of service</td>
<td>Hanging of the server or shutdown</td>
</tr>
<tr>
<td>Username and Password attacks</td>
<td>Information mining</td>
</tr>
<tr>
<td>System set point attacks</td>
<td>Information mining</td>
</tr>
<tr>
<td>Mine Corporate data for personal gain or to sell to competitor</td>
<td>Information Tampering/Replaying</td>
</tr>
<tr>
<td>Change of Data Points, Set point(s) System or in Data server</td>
<td>System shutdown</td>
</tr>
<tr>
<td>Modify Data points on graphics to deceive Operators that system is out of control and must Shut Down</td>
<td>Information Tampering</td>
</tr>
<tr>
<td>Capture, Modify, or Delete Data Logged in Operational Database, Locate Maintenance Database, modify or delete information regarding reliability tests, Calibration for equipment</td>
<td>Information Tampering</td>
</tr>
</tbody>
</table>
**Forward Engineering Process: (PIM – PSM)**

Meta PIM is designed to provide ECC based cryptographic services to the SCADA applications [9]. It simulates the working environment of the SCADA thereby considering and applying the security concerns wherever needed. The process of MetaPIM development involves Identification of the entities., Defining the attributes, Designing the constraints and functions. The metaPIM templates of SCADA ECC using GME were generated.

**Logic of PIM to PSM Transformation:**

The three tasks involved in the transformation are: First, it takes an input PIM model as a XML document. Second, it verifies the correctness of the marked PIM. Third, it performs a transformation by activating a matching mapper corresponding to the marked stereotypes in the input model. The mapper takes the marked model and its corresponding annotation to generate the target PSM. The output result can directly be generated into an executable code. PSM is based on Java Platform. However, a given PIM can be transformed to any PSM. The PSM can be transformed to code using code generators.

6 **PROPOSED TECHNIQUES**

This paper proposes three types of transformation techniques namely Platform Independent Model in XML to PSM, PIM in UML to PSM, and Design of Entity Transformation Language. These approaches aim at standardizing the process of model transformation and portability of models across different platforms [15].

**Platform Independent Model in XML to PSM**

The transformation of the Platform Independent Model in XML to the Platform Specific Model is developed using the State transition approach [12]. Since the generated XML follows a particular Document Order, this approach is used. The diagrammatic representation of the workflow and the state transition diagram are shown in Figure 2 and Figure 3 respectively. The steps to be followed in the transformation are given below.

**Transformation logic:**

1. Define the MetaPIM by identifying the entities, defining the attributes and constraints.
2. Formulate the PIM using the designed MetaPIM.
3. Using an XML code generator represent the PIM in XML format.
4. Using the Tuner eliminate unnecessary information from the raw XML code. Analyze the generated XML code.
5. Extract the key features (e.g. Attribute), Filter it and define the XML code in a format, which Follows a particular document order (DTD)
6. Transform the PIM (in XML) to PSM using the Model transformer.
7. Define the state transition for the Specific features represented in XML (e.g. `<atom>`).
8. Process the XML fragment by defining the appropriate feature and map to the specific PSM by associating it with the corresponding representation.

![Figure 2. Transformation approach for PIM in XML to PSM](image-url)
Generated XML code for the PIM

```xml
<atom id="id-0066-00000007" kind="SCADA" role="SCADA" relid="0x6">
  <name>SCADA</name>
  <regnode name="PartRegs">
    <value></value>
    <regnode name="secure">
      <value></value>
      <regnode name="Position" isopaque="yes">
        <value>156,191</value>
      </regnode>
    </regnode>
  </regnode>
  <attribute kind="+app_program:string" status="meta">
    <value>host</value>
  </attribute>
  <attribute kind="+password:string" status="meta">
    <value>xxxxx</value>
  </attribute>
  <attribute kind="+username:string" status="meta">
    <value>ganesh</value>
  </attribute>
</atom>
```

Figure 3 State transition Diagram for the generated XML code

Generated PSM in Java:

```java
Class SCADA {
  public String app_program="host";
  public String password="xxxxx";
  public String username="Scada User1";
}
```

Platform Independent Model in UML to PSM

In this approach the various types of syntactic representation of objects, attributes, operations, etc. in UML are analyzed and a corresponding state transition approach is proposed. Figure 4 shows the transformation logic for Platform Independent Model in UML to PSM.

1. Define the PIM in UML by analyzing the requirements of the system.
2. Formulate the entities, attributes, and functions using the UML representation.
3. Represent the constraints, using OCL in UML.
4. Transform the PIM (in UML) to PSM using the model transformer.
   a. Classify the Language features in the PIM in UML.

Tuned Code

```xml
<atom id="0066-00000007" kind="SCADA" role="SCADA" relid="0x6">
  <name>SCADA</name>
  <attribute kind="+app_program:string">
    <value>host</value>
  </attribute>
  <attribute kind="+password:string">
    <value>xxxxx</value>
  </attribute>
  <attribute kind="+username:string">
    <value>ganesh</value>
  </attribute>
</atom>
```

Figure 4 Transformation Approach from PIM to PSM in UML

```
Platform Independent Model in UML to PSM

1. Define the PIM in UML, by analyzing the requirements of the system.
2. Formulate the entities, attributes, and functions using the UML representation.
3. Represent the constraints, using OCL in UML.
4. Transform the PIM (in UML) to PSM using the model transformer.
   a. Classify the Language features in the PIM in UML.
b. Define the State Transition for the specific Language Feature.

Table II Language features and the corresponding representations

<table>
<thead>
<tr>
<th>Language feature &amp; its UML representation</th>
<th>PIM</th>
<th>Java Specific PSM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Constructor</strong> visibility mode create (): class Name</td>
<td><code>+create(): student</code></td>
<td><code>Public student()</code></td>
</tr>
<tr>
<td>Attribute visibility name [ multiplicity ordering ] [optional] : type = initial value [optional]</td>
<td><code>–Email_Address [ 1…5 ] : string</code></td>
<td><code>private String Email_Address [5]</code></td>
</tr>
<tr>
<td><strong>Operation</strong> visibility mode operation name (Parameter list): return type</td>
<td><code>+ Append(p1,p2): string</code></td>
<td><code>Public String Append(p1,p2)</code></td>
</tr>
<tr>
<td>Parameter kind[optional] name : type = default value [optional] kind -&gt; in</td>
<td><code>– encryption ( in key integer )</code></td>
<td><code>private encryption ( integer key )</code></td>
</tr>
</tbody>
</table>

In this approach, the language features such as constructors, attributes, operations, and parameters are analyzed and the corresponding transformation logic is used to obtain the PSM. Table II shows features and respective representations.

Figure 5-8 shows the state Transition diagrams for constructor, attribute, operation and parameter respectively.

Figure 5 State transition diagram for Constructor

Design of Entity Transformation Language

Models developed using MDA tools become dependent on the tool and hence such a model cannot be used in another tool. This technique aims to propose requirement specifications for a transformation language called ETL (Entity Transformation Language), for transforming a PIM to a PSM and a markup language called Entity Markup Language (EML), for representing a PSM.

Figure 6 State transition diagram for Attribute

Figure 7 State transition diagram for Operation

Figure 8 State transition diagram for Parameter

Interpretation of this EML generates PSM. EML can be used across different platforms that encompass a modeling environment and interpreters (ETL). The ETL has to be
well equipped with the standard library functions that map the PIM to the desired PSM. The library functions are designed after careful analysis of the target platform. The ETL scripts are embedded within the EML tags. An ETL transformation should map a source model instance to a target model instance by matching a pattern in a source model instance and creating a collection of objects with given properties in the target model instance. Separate libraries should be created for different platforms. Functions to access the XML form of the PIM should be designed properly. The transformation logic is depicted in Figure 9.

4.2 Embed the ETL scripts for model transformation in appropriate places.
5 Interpret the EML to get the PSM
6 Using the code generator, convert the PSM to code.

**Generic Modeling Environment**

An atom is a basic entity without any internal structures. An atom can be associated with another atom or it can be contained by a model as shown in Figure 10. A model is a basic entity, which can contain a model or an atom. A model can be associated with an atom or a model or it can contain another model or an atom as shown in Figure 11.

**Figure 9 Transformation of a PIM to a PSM**

Modeling is done using a tool called GME (Generic Modeling Environment). The transformation can be done to any platform, language. The generalized template shows the EML structure for a PSM and few entities like the Atom and the Model and the connection between them. The steps to be followed for the transformation are shown below.

1. Define the Meta PIM by identifying the entities, defining the attributes and constraints.
2. Formulate the PIM using the designed MetaPIM.
3. Using an XML code generator represent the PIM in XML format.
4. Analyze the generated XML.
4.1 Format a EML structure for the PSM.

**Figure 10 Atom associated with another Atom and Atom contained in a model**

**Generated Java Specific PSM Fragment in EML**

The simulated results of the logic of transformation using Entity Transformation Language is shown in the fragment below.

```xml
<etl:declare>
  <class id="id-0066-00000008">
    <name>EllipticCurve</name>
    <etl:build>
      <attribute kind="Attributes">
        <value>
          coefficientA:java.lang.Integer
          coefficientB:java.lang.Integer
          OrderOfCurve:java.lang.Integer
          fieldOrder:java.lang.Integer
        </value>
      </attribute>
      <attribute kind="IsAbstract">
        <value>false</value>
      </attribute>
      <attribute kind="Stereotype">
        <value>class</value>
      </attribute>
    </etl:build>
  </class>
</etl:declare>
```
Design and Development of MDA Transformation techniques for security services in Embedded environment

The proposed transformation techniques provide a comprehensive solution to the contemporary problems viz., inability to achieve reusability and interoperability, and tool migration in the software industry. These techniques not only provide solution to the above problems but also aim in standardizing the transformation process. Nevertheless, developing such techniques requires an in depth analysis and understanding of the syntax and the semantics of the various modeling tools. This work is intended to support design time interoperability and model management. The key feature of these techniques is to represent the model in a standard form. These techniques can be refined in future which might produce MDA components that are independent of any given proprietary platform, specific CASE TOOL which can be stored and retrieved in a uniform way and can be deployed in the context of different software process or method.

REFERENCES


[7] Interactive Objects and Project Technology, ”MOF Query/Views/Transformations, Revised Submission.” OMG Document: ad/03-08-11, ad/03-08-12, ad/03-08-13, 2003


Design and Development of MDA Transformation techniques for security services in Embedded environment


Author’s Biography

Dr.S.N.Sivanandam is Professor and Head of Department of Computer science and Engineering with 35 years of experience in PSG College of Technology. He has won the Who’s who award and Best Teacher award. He has published papers in National/International Journals and Conferences and authored many books. He has guided 15 PhD candidates. He is reviewer of International Journals and has chaired National and International Conferences

Ms.G.R.Karpagam is Assistant professor with a decade of experience in Department of Computer science and Engineering in PSG College of Technology. She is doing her research in the field of Model Driven Architecture and Security Services. She has published papers in National/International Journals and Conferences and reviewer of National/International Journals She has co-authored monograph for Database management systems and is the coordinator of Open source movement in PSG College of Technology
Modified De Bruijn graphs as Regular Logical Topologies in fiber Optic Networks

Dr. B. Sathyanarayana¹ Dr. G. Narasimhan²

¹Prof. in ECE Dept., St. Johns College of Engg & Tech, Yemmiganur E-mail: gae_sathi@rediffmail.com
²Prof. in ECE Dept., JNTU College of Engg. Hyderabad,
³Principal, St. Johns College of Engg. & Tech., Yemmiganur

ABSTRACT
The present paper proposes Modified De Bruijn graphs as attractive logical topologies for multihop lightwave networks. Logical topologies included in literature are De Bruijn graphs, Shufflenet, and Manhattan Street Network. The Researchers have constructed the Modified De Bruijn graphs which is a improved version of De Bruijn graphs [3][8][9]. The researchers have developed a shortest path routing algorithm for this Modified De Bruijn graphs, the results of which shows that for a given nodal in-degree and out-degree, a network based on Modified De Bruijn graph takes smaller average hop length than De Bruijn graphs. Further the Modified De Bruijn graphs are free from self-loops. Also for Modified De Bruijn graphs, G*(d), there are node-disjoint paths between every pair of nodes. Therefore networks based on Modified De Bruijn graphs can tolerate up to (d-1) node/link failures without any disruption in communication between any pair of nodes.

Keywords: De Bruijn graphs, Multihop Lightwave networks, Regular graphs, Logical topologies, shortest path routing algorithm, and Computer communication networks.

1. INTRODUCTION
New applications such as digital audio, visualization, telecommunications and mechanical imaging have resulted in a growing interest in lightwave networks. Yet, due to some fundamental limitations on the peak electronic processing speed (limited to a few Gigabit per sec), optical fibers alone cannot satisfy the growing thirst for higher bandwidths. Parallelism and concurrency techniques must be employed by end-users operating with electronic front ends in order to tap the huge optical bandwidth of a single fiber. Wavelength Division Multiplexing (WDM) has emerged as the potential candidate for providing concurrency in lightwave networks [1],[2],[6],[14] which are classified into single-hop and multi-hop networks. There are technological difficulties in implementing WDM in single-hop approach [11], and [12], such as, requirement of rapid tune ability or wavelength ability and pre-transmission coordination. A multihop network employs a multitude of fixed tuned transmitters and receivers in the NIU (Network Interface Unit) of each node in the network. Each transmitter in the network transmits at a different wavelength and each node is assigned one or more channels (wavelengths) to which its lasers or filters are to be tuned. Wavelength assignments cause the network node to appear logically connected irrespective of the physical layout in which optical fibers and NIUs are laid out. A packet from a source node to a destination node may have to hop through one or more intermediate nodes. At each intermediate node, optical data is converted into electronic form, packet-headers are decoded and the packet is switched.
electronically and transmitted on the appropriate wavelength towards its final destination.

This paper deals with the construction of Modified De Bruijn graphs, which is a regular logical topology. Also a shortest path routing algorithm is developed for routing of packets between any pair of nodes. The eminent [4] distinguishes between two types of topologies – the physical topology and logical topology. By the physical topology, it means the actual underlying network topology, which is commonly, a broadcast star or bus. On any underlying physical topology one can impose a carefully selected connectivity pattern that provides dedicated connections between certain pairs of nodes. The logical topology of a multihop WDM lightwave network refers to the connectivity / wavelength assignment pattern and can be superimposed on any given physical topology. In the recent past, several different logical topologies proposed as candidates for multihop WDM lightwave networks [6], [8], [9].

In designing a “good” multihop system, the important issues which the system architect have to address are

1. the vital structure chosen must be close to optimal in some sense, for instance, the structure’s average (hop) distance between nodes must be small, or the average packet delay must be minimal, or the maximal flow on any link in the virtual structure must be minimal

2. the nodal processing complexity also must be less because, the high-speed environment allows very little processing time, which requires a simple routing mechanism. An excellent review of multihop networks is presented in [7], [11].

3. It is possible to create the logical topology to reflect the traffic patterns in the network

4. Although the physical topology may be arbitrary, the logical topology can be regular, so that routing and flowing control are simpler and traffic balancing is improved.

Regular structures, because of their structured node-connectivity pattern, are having simplified routing schemes. Regular structures that have received a significant amount of attention in the literature are (i) the perfect shuffle (called ShuffleNet) [1], [2], [7] (ii) the De Bruijn graph [8], [9], [14] (iii) the toroid (Manhattan Street Network, (MSN) [10]. The De Bruijn graphs as logical topology for multihop lightwave networks were proposed by Kumar Sivaraj et. al. [8], [9].

This paper proposes the Modified ‘De Bruijn graphs’ as logical topologies which are completely regular graphs without self-loops.

Construction of a new regular logical topology i.e., Modified De Bruijn graphs are explained in the following section.

2. CONSTRUCTION OF MODIFIED DE BRUIJN GRAPHS

The construction of Modified De Bruijn graph $G^*(\Delta, d)$ is based on the De Bruijn graph [5], [8], [9]. The network for Modified De Bruijn graph is defined for ‘N’ nodes, where $N = \Delta^d$ for $\Delta \geq 2$ and $d \geq 2$. A typical representation of De Bruijn graph is given in fig.1. Here, connection criteria of edges between nodes of the De Bruijn graphs consists of self-loops at nodes $(0_{1}, 0_{2}, ... , 0_{d})$, $(1_{1}, 1_{2}, ... , 1_{d})$, $(\Delta-1, \Delta-1), ... , (\Delta-1, \Delta-1)$, which are not actually used to transmit packets. This makes the De Bruijn graphs not completely regular, self-loops reduces the routing option, as they are not used for transmission. It is important to note that in graph i.e., fig.1. $\Delta$ self-loops which are present in the graph but which are not physically present in the actual network. In the Modified De Bruijn...
graph (network), the self-loops are removed and the nodes 
\((0_1, 0_2, \ldots, 0_d), (1_1, 1_2, \ldots, 1_d), \ldots, ((\Delta -1)_1, (\Delta -1)_2, \ldots, (\Delta -1)_d)\),
are connected to one another in a cyclic manner as shown in figures 2. An edge from a self-node \((a_1, a_2, \ldots, a_d)\) to another self-node \((b_1, b_2, \ldots, b_d)\) can be defined as \((a_1, a_2, \ldots, a_d, a_{d+1})\) where \(a_1 = a_2 = \ldots = a_d\), and \(b_1 = b_2 = \ldots = b_d\) and also \(b_i - a_i = 1\). The ‘\(\Delta\)’ self edges that are physically added to the Modified De Bruijn network reduces the distance between nodes for a significant number of node pairs in the graph. Subsequently, reduces the average hop length between the nodes of the graph.

On similar lines of the De Bruijn graph, a node in the Modified De Bruijn graph is represented by a string of ‘\(d\)’ digits. An edge from node ‘\(X\)’ to node ‘\(Y\)’ can be represented by a string of \(d+1\) digits, the first ‘\(d\)’ digits represent the node ‘\(X\)’ and the last ‘\(d\)’ digits represent node ‘\(Y\)’. Similarly, any path in the graph of length ‘\(k\)’ hops can be represented by a string of \(d+k\) digits. In the case of self-edges, an edge from self-node ‘\(X\)’ to self-node ‘\(Y\)’ can be represented by a string of \(d+1\) digits, all of which representing the node ‘\(X\)’, and there exists a self-edge between self-node ‘\(X\)’ and self-node ‘\(Y\)’ if only if

\[ b_i - a_i = 1. \]

The following section deals with the development of shortest path routing algorithm for Modified De Bruijn graphs. It is used to compute the distance, in number of hops from a given source node to a destination node.

3. **Shortest-Path Routing Algorithm for Modified De Bruijn graphs**

Modified De Bruijn graphs have a self-routing property, namely, packets can be routed from a source (\(\text{Src}\)) to a destination (\(\text{Dest}\)) using distributed routing algorithm in which the packets are forwarded to intermediate nodes, based on a routing decision that is given only on the label of the destination node of the packet.

We explain the notations and the definitions used in the development of shortest path routing algorithm

3.1 **Notations and Definitions used in the algorithm:**

\(\Delta\) — Degree of the network

\(d\) — (Diameter of the Network) string length of each node label, which denotes the maximum length between any two nodes.

\(N\) — The total number of Nodes in the Network.

\(\text{Src}, \text{Dest}\) — be the any source and destination nodes in the network and denoted as \(\text{Src} = (a_1a_2\ldots a_d)\) and \(\text{Dest} = (b_1b_2\ldots b_d)\) where \(a, 's\) and \(b, 's\) can take values ‘0’, ‘1’, ‘2’, ‘\(\Delta\)’.

The nodes of the Modified De Bruijn graphs (Network) are classified into two types.

\(\downarrow\) **Self-node**, in which all symbols in the label are same.

\(\downarrow\) **Non-self node**, in which all symbols in the label are not same.

The source-destination pairs are categorized into the following four types.

\(\downarrow\) **Self-node to Self-node**

\(\downarrow\) **Non-self node to Non-self node**

\(\downarrow\) **Self-node to Non-self node**

\(\downarrow\) **Non-self node to Self-node**

\(\text{IsSelf (Node)}\): Finds whether it is a Self-node or not.

\(\text{BothSelf} : \) Finds the shortest distance in number of hops between a Self-node (source) to another self-node(destination).

\(\text{BothNonself} : \) Finds the distance between two non-self nodes

\(\text{SourceSelf} : \) Finds the distance from self-node (\(\text{Src}\)) and non-self node (\(\text{Dest}\))
**Modified De Bruijn graphics as Regular Logical Topologies in fiber Optic Networks**

DestSelf : Finds the distance non-self node (Src) to self-node (Dest)

DistBothSelf : Finds the distance from Src and Dest through self-nodes (as intermediate nodes)

Length, Length1, len : All these indicate the number of hops.

PreSelfToDest : Refers to the preceding self-node to the destination node.

AdjSelfToSrc : Refers to the adjacent self-node to the source node.

ShiftMatch (i, Src, Dest) : An operation on the two strings Src and Dest to be TRUE iff Src = (b₁, b₂, ..., bₙ₋₁) = (aᵢ₊₁, aᵢ₊₂, ..., aₙ) and FALSE otherwise.

FindAdjacent(Src, Dest) : Returns the next adjacent node from Src on the shortest path towards Dest.

Merge (Src, Dest, i) : is a string (or sequence) of length d+i given by (a₁,a₂,...,aᵢ,bₙ₋₁,...,bₙ).

FindNormalLength : Find the distance between source node and the destination node using the De Bruijn routing

**Example:** ∆=2, d=3, Total Nodes= N = 8, total Edges= ∆ * N = 2*8 = 16. Since ns = 2, there are 2 symbols namely, ‘0’ and ‘1’ are used for labeling the network nodes. Hence the nodes of the network are ‘000’, ‘001’, ‘010’, ‘011’, ‘100’, ‘101’, ‘110’, and ‘111’.

Algorithm for the shortest path routing in Modified De Bruijn graphs is as follows:

3.2 Algorithm for the Shortest path routing in Modified De Bruijn Graph (logical topology)

Procedure Modified-DeBruijn-route (Src, Dest, ∆, d)

Begin
if((NOT IsSelf(Src)) AND (NOT IsSelf(Dest))) then
  len=BothNonself(Src, Dest);
else if ((IsSelf(Src) AND IsSelf(Dest))
  len=BothSelf(Src, Dest);
else if ((IsSelf(Src) AND IsSelf(Dest)))
  len=SourceSelf(Src, Dest);
else if (IsSelf(Src) AND IsSelf(Dest))
  len=DestSelf(Src, Dest);

Src=FindAdjacent(Src, Dest);
return len; /* len is the shortest hop distance */

End.

Procedure FindNormalLength (Src, Dest, ∆, d)

Begin
for (i=0 to d) Begin
  if (ShiftMatch(i, Src, Dest) = False)
    break;
End

Merge (Src, Dest, i);
return i; /* i is the pathlength */

End

Procedure BothSelf(Src, Dest, ∆, d)

Begin
Length=FindNormalLength(Src, Dest, ∆, d);
if (Length < d) return Length;

Length1=DistBothSelf(Src, Dest);
if (Length < Length1) return Length;

return Length1;

End

Procedure BothNonself(Src, Dest, ∆, d)

Begin
Length=FindNormalLength(Src, Dest, ∆, d);
if (Length < d) return Length;

Length1=DestSelf(Src, AdjSelfToSrc)+
  SourceSelf(PreSelfToDest, Dest)+

  DistBothSelf(AdjSelfToSrc, PreSelfToDest);
if (Length > Length1)
  return Length1;

return Length;

End
4. DERIVATION OF AVERAGE HOP LENGTH

In this section, to consider the derivation of Average number of hops ($H$) for a $G^*\text{(}\Delta,d\text{)}$ is the distance, measured in number of hops between any source and destination nodes averaged over total source destination pairs. Hence, the average number of hops $H$, is given by

$$H = \frac{1}{N} \sum_{i=1}^{N} H_i \quad (1)$$

where ‘$N$’ is the number of nodes in the network, and ‘$H_i$’, the hop distance of the $i^{th}$ source-destination pair.

A comparison of the results of average hop lengths $[H]$ for De Bruijn and Modified De Bruijn Graphs (networks) for different values of $\Delta$, $d$ are shown in Table (1).

### Table 1: Comparison of Average Number of Hops of De Bruijn and Modified De Bruijn Graphs

<table>
<thead>
<tr>
<th>$\Delta$</th>
<th>$d$</th>
<th>$N$</th>
<th>De Bruijn $H$</th>
<th>Modified De Bruijn $H$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>8</td>
<td>2.107143</td>
<td>1.964286</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>16</td>
<td>2.833330</td>
<td>2.733330</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>32</td>
<td>3.649194</td>
<td>3.584600</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>64</td>
<td>4.532242</td>
<td>4.492559</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>128</td>
<td>5.460138</td>
<td>5.436516</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>256</td>
<td>6.416973</td>
<td>6.403247</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>512</td>
<td>7.391695</td>
<td>7.383687</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>9</td>
<td>1.666670</td>
<td>1.625000</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>27</td>
<td>2.478632</td>
<td>2.448718</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>81</td>
<td>3.386111</td>
<td>3.370833</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>243</td>
<td>4.344047</td>
<td>4.337160</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>729</td>
<td>5.326120</td>
<td>5.323221</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>2187</td>
<td>6.318803</td>
<td>6.317634</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>16</td>
<td>1.750000</td>
<td>1.733330</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>64</td>
<td>2.639881</td>
<td>2.630952</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>256</td>
<td>3.598529</td>
<td>3.595037</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>1024</td>
<td>4.584448</td>
<td>4.583257</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>2556</td>
<td>5.590529</td>
<td>5.590321</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>25</td>
<td>1.800000</td>
<td>1.791667</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>125</td>
<td>2.727742</td>
<td>2.724194</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>625</td>
<td>3.705897</td>
<td>3.704795</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>3125</td>
<td>4.699964</td>
<td>4.699664</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>36</td>
<td>1.833330</td>
<td>1.828571</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>216</td>
<td>2.782300</td>
<td>2.780620</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>1296</td>
<td>3.769412</td>
<td>3.768980</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>7776</td>
<td>4.766500</td>
<td>4.766405</td>
</tr>
</tbody>
</table>

5 RESULTS AND CONCLUSIONS:

From the figures 1, 2 and table 1, the following conclusions can be drawn:

i) The proposed logical topology in this paper i.e., Modified De Bruijn graphs (network) (from Fig.2 ) is absolutely a regular structure network whereas in De Bruijn graphs (networks), the self-loops makes it not absolutely regular.
ii) The shortest path routing algorithm developed in this paper for Modified De Bruijn graphs has been used to compute the \( H(\text{average hop lengths}) \) for different values of \( \Delta, d, N \), as shown in columns 4, and 5 (of Table 1). On comparing corresponding results of De Bruijn graphs, it is found that the Modified De Bruijn graphs have shown significant improvement.

iii) The edges formed between self nodes in the Modified De Bruijn graphs improves the reliability of transmission of packets between any source and destination nodes, can tolerate upto \( \Delta-1 \) link failures.

Further scope of the study: The above-proposed Modified Debruijn graphs (logical topologies) with the shortest path routing algorithm can be used to find the edge loading. Since Modified Debruijn graphs, are regular logical topologies which could be implemented in fiber optic networks (WDM lightwave networks), the speed and enormous bandwidth of the lightwave networks can be exploited by developing deflection routing techniques for Modified Debruijn graphs (\( G'(\Delta, d) \)).

REFERENCES


Author’s Biography

Dr. B. Sathyanarayana, working as Associate Professor & Head of Department of Computer Science and Technology with 16 years of experience in Sri Krishnadevaraya University, Anantapur 515003 (AP). He has presented many valuable papers in National/International Journals and Conferences. He is supervising for 10 Doctoral theses in the areas of computer networks, mobile computing, Data warehouse management and image processing.

Dr. G. Narasimhan has been working as Associate Professor in Sri Krishnadevaraya University, Anantapur since 1989. He has published many articles in lead journals and participated in several seminars. His research interests include Linguistics, Phonetics, Philology and Communication Skills.
Performance comparison of Mobile reflective Context aware middleware in Adhoc Smart space with Cross layering Approach

N. Radhika¹ Dr. S. Arumugam²

ABSTRACT

The aim of our paper is to design a context aware middleware that senses the context by verifying the profile of the user. Services are fetched seamlessly from Adhoc spaces. The context is fetched based on three factors- environment, activity and location of the user. Smart spaces are composed of sensors, attenuators and other mobile devices that are capable of offering service to the user. Multiple smart spaces are created and seamless communication and service discovery are performed across smart spaces.

Cross layering is an important factor added to the context aware middleware that provides a link with the network routing protocol in the network layer and service discovery protocol in the transport layer. Reflective middleware supports dynamism in the service offered. Multiple numbers of reflective services could be offered to the user at a particular instant of time. The performance of our reflective middleware with cross layering is compared with the already existing Context aware middleware and found that the performance was better.

Keywords: Context aware, Middleware, Seamless, Reflective, Cross Layering

1. INTRODUCTION

Smart spaces are adhoc spaces composed of smart and dummy nodes capable of sensing the profile of the user and offering the required service to the user. Middleware is the connectivity software that joins applications through communication mechanism creating transparency, scalability and interoperability and lies between software applications it assists and the platform it is based upon [1]. The context of the user is analyzed based on the environment, activity and location factor. The environmental factor includes lightening, rain and other Climatic condition of the day. The location factor includes the position of the user and the activity includes the work he is performing. Depending on the preference and the context, the profile of the user is analyzed. Relevant services are delivered from multiple smart spaces and the best matching service is selected from all the available services. Middleware is reflective as large number of services are offered to the user.

Cross layering is the phenomena to link routing protocols i.e. in turn to bring interoperability between the service discovery protocol like slp and pdp and network routing protocol like proactive and reactive protocols.

2. CONTEXT AWARE MIDDLEWARE

The context aware middleware is designed in between the network and routing layer where the interoperability of service discovery protocol occurs. The middleware also supports the cross link between the network routing
protocol and the service discovery protocol in the transport layer. The transmission from the user to the sender is through the routing protocols.

3. NETWORK ROUTING PROTOCOL

Proactive and reactive protocols are chosen depending on whether the path of routing is decided prior or at the time of routing. If the path of transmission is decided prior, the DSDV Protocol is chosen and if it is decided at the time of routing then reactive protocol as AODV is chosen. This protocol in our experimental situation is chosen depending on the number of nodes in the network and the distance of the nodes from the network. At the first step interoperability is setup between AODV and DSDV protocol in the network layer and in the second step interoperability is set up between service discovery protocols in the transport layer and the network layer protocol.

4. LAYERED HIERARCHY

Our Cross layer approach is divided in to five layers. Mac and physical layer is capable of maintaining physical connection. Network and routing layer establishes routing between sender and receiver. Middleware acts as hardware between application and the software. Reflective context aware middleware is designed in between network and routing layer i.e. it can function in either of the layers. In the transport layer new services are discovered and transmitted to the user. Smart spaces are designed in the transport layer. Each smart space runs on a service discovery protocol which helps in the discovery of service. The application layer and transport layer put together helps in providing service.

Layered approach is mostly chosen as a standard for protocol stack. The key design is the low design complexity, modularity and improved maintainability. Layered protocols are easily maintainable as error could be traced out. Cross layering is not a simple replacement of layered architecture nor is it a simple functionality.

Figure 1: Cross Layer Architecture

5. SERVICE DISCOVERY

In our architecture the service discovery is through peer to peer system with no client – server concepts. All the node in the network has equal rights to act as service provider or the consumer of service. Clustering of peer nodes are formed in the smart space. Smart space is nothing but the smart networks where nodes register their service and then the service is broadcasted to all other nodes in and across the network.

Layered protocols are easily maintainable as error could be traced out. Cross layering is not a simple replacement of layered architecture nor is it a simple functionality.

Figure 2: Communication across smart space in peer to peer clustering
6. Service Discovery Protocol

The two protocols under study is the service location protocol and pervasive discovery protocol. Services are detected seamlessly using these protocols without any look up server. Seamless discovery of service occurs without any query request or response which is an innovative idea in our paper.

Usually service discovery is selected by finding a match between the description and the service location. But our model deals with the seamless service discovery by interoperability.

7. Readings

By using cross layer approach interaction occurs between network routing protocol in the network layer and the service discovery protocol in the transport and application layer.
8. EXPERIMENTAL RESULTS

a. cross layering (Time Vs delivery ratio) using DSDV in network layer and SLP as service discovery protocol
The above graph shows the usage of DSDV protocol on the network layer and SLP for service discovery in the transport layer. The experimental set up is done on a SLP Smart Campus. Cross layering is carried out and the result is compared with the already existing context aware architecture and found from the graph that the existing architecture of cross layering had higher delay compared to our architecture. The rate of delay in delivery in our architecture was found to be 50% less than at the existing architecture.

It is in the context aware middleware that the actual cross layer is established and thereby routing and service Discovery protocols are made to interoperate. In the experimental set up we have taken in to consideration the admission policy of the smart campus. The user profile is sensed and depending on the interested activity of the user the particular course is fetched from and across the smart spaces and seamlessly integrated in the device of the user. If the relevant match is not attained the next nearest match is provided to the user.

b. cross layering (Rate Vs delay) using Aodv in routing layer and PDP as service discovery protocol

![Graph showing performance comparison](image)
The above graph shows the usage of AODV protocol on the routing layer and PDP for service discovery in the transport layer. The experimental set up is done on a PDP Smart Campus. Cross layering is carried out and the result is compared with the already existing context aware architecture and it is found from the graph that the existing architecture of cross layering had higher delay compared to our architecture so the rate of delay in delivery in our architecture was found to be 55% less than at the existing architecture.

In the experimental set up above we have taken into consideration the Job seeking of the student. As the students become eligible for placement his profile is sensed and depending on the mark and his interest the best matching job codes and vacancies are displayed in the device of the user. The search of the job and placement of the student in the campus becomes easier. The student depending on his interest will appear for the campus recruitment.

c) cross layering (Time Vs delivery ratio) using DSDV in network layer and SLP as service discovery protocol
In the above experimental setup the admission module is considered. DSDV protocol of the network layer and the service location protocol of the transport layer are made to interoperate. Here the time taken for delivery is taken into account and it could be inferred from the graph that the cross layer in the proposed context aware architecture seems to give better delivery ratio than the already existing context aware architecture.

d) Cross layering (Time Vs delivery ratio) using Aodv in the network layer and pdp as service discovery protocol

In the above experimental set up the job module is considered and cross layering is attained using the pdp smart space in the transport layer and Aodv protocol in the network layer.
8. CONCLUSION

Context aware middleware is designed and the functionality is to sense the context based on the location, activity and environment of the user. The profile is analyzed and the seamless interaction between service discovery protocol and network layer protocol occurs. This is brought about by the cross layering approach. The services are discovered and routed through the network layer to the user. The performance was evaluated and it was found that in case of delay and delivery ratio our models seems to perform better than the existing model.

9. FUTURE WORK

The drop rate of the packet has to be compared and security feature have to be added to make the Architecture a perfect referential model. The Context aware middleware has to be tested with the combination of other service discovery protocols and network routing protocols.

REFERENCE


123
Performance comparison of Mobile reflective Context aware middleware in Adhoc Smart space with Cross layering Approach


[18] http://user.it.uu.se/henrikl/aodv/

Author’s Biography

Miss N. Radhika is currently pursuing PhD degree in Computer Science at Government College of Technology, Coimbatore. She has obtained her Bachelor of Science from P.S.G.R Krishnammal College for Women, Coimbatore, Bharathiar University in the year 1996 and Bachelor of Education (Physics and Maths) from Sri Sathya Sai Institute of Higher learning, Ananthapur, Sri Sathya Sai University in the year 1997. She has obtained Master of Computer Applications from P.S.G.R Krishnammal College for women, Coimbatore, Bharathiar University in the Year 2000. She is currently working as a Sr.Lecturer, in the department of Computer Science and Engineering at Amrita School of Engineering, Coimbatore. She has served as a lecturer in the department of Computer Science and Engineering at Sri Krishna College of Engineering and technology in the year 2000-2001 and then onwards worked at various positions of lecturer and Sr. Lecturer at Amrita School of engineering. She has published 9 technical paper in international and national conferences and two technical papers in national journals. Her area of interest includes mobile computing, pervasive computing, Mobile-Adhoc Networks, Software engineering and operating systems.

Dr. S. Arumugam, received the PhD degree in computer science and in engineering from Anna University, Chennai in 1990. He also obtained his B.E. (electrical and electronics engg) and msc. (engg)(applied electronics )degrees from p.s.g college of technology, coimbatore, university of madras in 1971 and 1973 respectively. He is working in the directorate of technical education government of Tamil nadu from 1974 at various positions from associate lecturer, lecturer, assistant professor, professor and principal; presently he is working as additional director of technical education. He has guided 4 PhD scholars and guiding 10 PhD scholars. He has published 70 technical papers in international and national journals and conferences. His area of interest is including network security, biometrics and neural networks.
Adaptive - Network - Based Fuzzy Inference System for Tool Wear Prediction

V. Kalaichelvi1  D. Sivakumar2  R. Karthikeyan3

ABSTRACT
Due to the complexity of machine tool structure and the cutting process, the dynamics of machining processes are still not completely understood. This is especially true for high speed machining processes. To model these complex processes, new techniques that can represent complex phenomenon combined with learning ability are needed. The combined neuro fuzzy approach appears to be ideally suited for this purpose. An Adaptive Network based Fuzzy Inference System (ANFIS) network is used to predict tool wear states in turning operation. Neuro fuzzy adaptive network has both the learning ability of a neural network and the linguistic representation of complex and vague phenomenon. An appropriate model representing the influence of tool wear states is first established. This model is then improved upon by learning with the given set of training data. The improved models are verified by the use of test data obtained from actual experiments. The effectiveness of this approach is ascertained by comparing with results obtained through neural modeling.

Keywords: Neural network, ANFIS, Machining process, Tool wear states, modeling.

1. INTRODUCTION
Machining, the cost of which amounts to more than 15% of the total value of all products of the entire manufacturing industry [1], plays a central role in modern manufacturing. Modeling with the help of experimental results forms an integral part in the investigation of the complicated dynamic mechanisms of machining operations. Various approaches have been proposed to model and to simulate the machining process. Merchant [1] and Ehmann et al. [2] have provided a comprehensive review of the various modeling and simulation approaches. For the ease of discussion, Ehmann et al. classified the existing methods into three categories: analytical, experimental, and mechanistic and numerical methods. It appears that almost all the experimental methods were carried out before the 1990s, and the mechanistic and numerical approach was adopted mainly during the last two decades [2]. Analytical methods, which are generally based on established scientific principles, are probably the first modeling approach. A good example is the very early work by Merchant [3, 4] on the shear angle model by the use of minimum energy principle. Other typical works based on analytical methods are reported in [5, 6]. Experimental or empirical approaches use experimental data as the basis in formulating the models [7-9]. Mechanistic and numerical methods integrate the analytical and the empirical
methods, generally, by the use of modern computer techniques. These methods are very powerful for simulating the various machining processes and have been used by various investigators [10-12]. As pointed out by Ehmann et al [2], machining process modeling has entered into significant industrial applications instead of just being an academic curiosity.

Thus, the various practical applications and on-line diagnosis such as process monitoring, on-line fault diagnosis, closed loop analytical solutions, and adaptive control require much more detailed and exact machining process modeling. In order to carry out these practical applications without more detailed modeling, some radically new techniques are needed.

A distributed neural network can be applied to tool wear monitoring. The network is a feed forward multilayer network under the back-propagation learning used for pattern recognition. The network stems from the study of human system which serves as an associative memory [13]. Neural networks are said to be low-level computational structure that can offer good performance in dealing with sensory data. In comparison, fuzzy logic can provide higher level reasoning. However it has limited learning capacity. It would be difficult for a human operator to tune fuzzy rules and membership functions from the training data set [14]. As suggested by S.N.Sivanandam et al., Neural networks and fuzzy systems individually have reached a degree of maturity where they are applied to real world situations. Researchers often utilize these two technologies in series using one as a preprocessor or postprocessor for the other. Examples include the use of fuzzy inputs and outputs for neural networks, the use of neural networks to quantify the shape of a fuzzy membership function. Hence a promising approach is to fuse fuzzy systems and neural networks into a single system that would combine the benefits of the two [15].

2. Objectives of the Present work

The main objectives of the present work can be summarized as follows:

(i) A neural network based tool wear monitoring algorithm uses spindle speed, feed rate and depth of cut to predict the tool wear states. As a first step, the neural network is trained for a set of cutting conditions which covers a range of cutting speeds and feeds and the corresponding tool wear values are predicted.

(ii) Tool wear states are also predicted by using ANFIS modeling which is an integrated system that uses neural network as a tool in fuzzy systems. Finally the simulation results of different modeling techniques are compared.

3. Experimental Procedure

The turning experiments are carried out in a CNC lathe. The current of the spindle motor in the lathe is used to estimate the tool wear states. The spindle motor current depends on the cutting conditions and tool wear. The cutting conditions used for the experiments are listed in Table 1.

<table>
<thead>
<tr>
<th>Cutting conditions</th>
<th>Spindle speed</th>
<th>Feed rates</th>
<th>Depth of cut</th>
<th>Work piece</th>
<th>Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>250, 740 and 1150 rpm</td>
<td>0.05 and 1.07 mm/rev</td>
<td>0.5, 0.75 and 1.0 mm</td>
<td>Al +10% of SiC particulate reinforced composite material</td>
<td>K10 cemented carbide</td>
</tr>
</tbody>
</table>

Table 1 Experimental conditions
The work piece material used for the experiment is Al + 10% of SiC particulate reinforced composite material. In this composite, Aluminium alloy is a major component and silicon carbide is a reinforced component. The material is prepared through stir casting and can be machined using K10 cemented carbide tool.

In order to establish the model of the spindle motor current as a function of cutting parameters and tool wear, the effects of cutting parameters and tool wear based on current signal are first examined. Experiments are conducted on a turning lathe for various sets of cutting conditions like spindle speed, feed rate and depth of cut. For each set of cutting conditions, machining is done starting with insertion of a fresh tool continuing until the tool is worn out. Cutting tests are performed on a machining center. The four axes (spindle, X, Y, and Z) of the machine are directly driven by a permanent magnet DC motor. DC motor current of the machine center is measured with a multimeter. The cutting conditions are given through computer and the cutting operation is performed automatically. The tool wear at various time intervals are measured using an optical microscope.

4. ANN FOR TOOL WEAR ESTIMATION

An artificial neural network (ANN) does not need a mathematical formulation. By studying sample data, ANN not only simulates highly accurate non-linear data but also keep the data in memory. In addition, ANN possesses a strong ability to deal with discrete data. Moreover, ANN can adjust the state of the old network to fit the new experimental data instead of abandoning or redoing the old data or network. A back propagation (BPN) algorithm with multilayered feed forward network used for turning operation is shown in Figure 1 [16].

Figure 1 Multilayered Feed Forward Neural Network with one Hidden Layer

The network has three layers, an input layer with 4 input variables, a hidden layer and an output layer which gives tool wear. The weight connection between input layer and hidden layer is \( W_{ji} \) where \( j = 1 \) to \( 4 \) and \( i = 1 \) to \( 4 \). The weight connection between hidden layer and output layer is \( W_{mj} \) where \( m = 1 \) and \( j = 1 \) to \( 4 \).

In the proposed structure two additional nodes of unit input are used for threshold adjustments and their outputs are connected to the nodes in hidden and output layers. The weight of the connection between additional nodes and others are denoted by \( \theta_{1} \) where \( j = 1 \) to \( 4 \) and \( \theta_{2} \) where \( m = 1 \) and \( j = 1 \) to \( 4 \).

The ANN structure considered here has four neurons in the input layer with \( \text{logsig} \) transfer function, four neurons in the hidden layer with \( \text{tansig} \) transfer function and one neuron in the output layer with \( \text{purelin} \) transfer function. Especially these are advantageous for use in neural network trained by BPN algorithm. The reason is that, simplest relationship between the value of the function at a point and the value of the derivative at the point
Adaptive Network Based Fuzzy Inference System for Tool Wear Prediction

reduces the computational burden during training. The weights which are the key in predicting the tool wear states are unknown for a process. They are calibrated by training the algorithm on cutting test results for a work material with a specified cutting tool. The cutting tests are conducted under different feeds, cutting speeds and depth of cut and tool wear is measured for each case. The weights are identified through iterative learning process. Backpropagation is an optimization procedure based on gradient descent that adjusts the weights to reduce the system error. During the training process, input patterns are presented to the network in some sequence. Each training pattern is propagated forward, layer by layer until an output pattern is computed. The computed output is then compared with the desired output and an error value is computed. The errors are used as inputs to feedback connections from which adjustments are made to the weights, layer by layer in backward direction.

5. SIMULATION RESULTS OF ANN MODELING
The performance of neural modeling for tool wear prediction is tested by using measured data. Their performance is assessed by evaluating the scatter between the observed and simulated results through Percentage Mean Error (PME) and Root Mean Square Error (RMSE). The variations of PME and RMSE values with number of hidden neuron are shown in Fig. 2 (a) and 2(b).

Figure 2 (a) Percentage mean error Vs Hidden neurons

Figure 2 (b) Mean square error Vs Hidden neurons
The least error corresponds to the hidden layer with 4 neurons. Hence this is chosen as the optimized ANN structure. Fig. 3 (a) and 3(b) shows the trained and tested output of neural modeling. Fig. 4 shows the comparison between measured and predicted tool wear (mm) by neural modeling.

Figure 3 (a) Trained output of neural modeling

Figure 3 (b) Test output of neural modeling

Figure 2(a) Percentage mean error Vs Hidden neurons

Figure 3(a) Trained output of neural modeling

Figure 3(b) Test output of neural modeling
6. ANFIS MODELING

The neuro fuzzy systems are considered with Takagi Sugeno Kang (TSK) fuzzy rules whose consequents are linear combinations of their preconditions [17]. The TSK fuzzy rules are in the following forms.

Rule 1: If \( x \) is \( A_1 \) and \( y \) is \( B_1 \), then

\[
1 \cdot r_1 \cdot q_1 \cdot x_1 + p_1 \cdot f_1 = 0
\]

Rule 2: If \( x \) is \( A_2 \) and \( y \) is \( B_2 \), then

\[
2 \cdot r_2 \cdot q_2 \cdot x_2 + p_2 \cdot f_2 = 0
\]

For given input values of \( x_1 \) and \( x_2 \), the inferred output is calculated as follows:

\[
f = \frac{w_1 \cdot f_1 + w_2 \cdot f_2}{w_1 + w_2}
\]

This can be expressed as

\[
f = w_1 \cdot f_1 + w_2 \cdot f_2
\]

Fig. 5 shows the equivalent architecture for the given input values of \( x \) and \( y \) in the ANFIS nodes. Each layer has similar functions as listed below. The output of node in layer 1 is denoted as \( O_{1,i} \) [18].

Layer 1: Every node in this layer is an input node that just passes external signal to the next layer.

Layer 2: Every node in this layer is an adaptive node with node output defined as

\[
O_{2,i} = \mu_{B_i}(y) \quad \text{for } i = 1, 2
\]

Where \( x \) and \( y \) are input nodes and \( A_i \) and \( B_i \) are fuzzy sets.

Layer 3: Every node in this layer is fixed nodes labeled \( \Pi \), whose output is the product of incoming signals. For instance

\[
O_{3,i} = \mu_{B_i}(y) \quad \text{for } i = 3, 4
\]

Output from each node represents the firing strength of a rule.

Layer 4: Every node in this layer is a fixed node labeled \( N \) and calculates the normalized firing strength of a rule. The \( i \)th node calculates the ratio of \( i \)th rule’s firing strength to the sum of all rule’s firing strength.

\[
O_{4,i} = \frac{w_i}{w_1 + w_2} \quad \text{for } i = 1, 2
\]

Layer 5: Every node in this layer is an adaptive node with a node function.
Adaptive - Network - Based Fuzzy Inference System for Tool Wear Prediction

Layer 5: The output layer of this Layer is a fixed node labeled \( \Sigma \), which computes the overall output as a summation of all the incoming signals.

\[
O_{5,i}^{\Sigma} = \sum_{i=1}^{2} w_i f_i = \sum_{i=1}^{2} \left( p_i x + q_i y + r_i \right) \quad \text{for } i = 1, 2 \quad (9)
\]

This layer calculates the weighted consequent value, with \( w_i \) = output of layer 4 and \( \{p_1, q_1, r_1, p_2, q_2, r_2\} \) are the consequent parameters.

Layer 6: The single node in this layer is a fixed node labeled \( \Sigma \), which computes the overall output as a summation of all the incoming signals.

\[
O_{6,i}^{\Sigma} = \text{overall output} = \sum_{i=1}^{2} w_i f_i = \frac{\sum_{i=1}^{2} w_i f_i}{\sum_{i=1}^{2} w_i} \quad (10)
\]

Thus an adaptive network that is functionally equivalent to a Sugeno fuzzy model has been formulated. It is observed that the structure of this adaptive network is not unique. For example, the weight normalization can be performed in the last layer. The adaptive network can tune the fuzzy system with a back propagation algorithm based on the collection of input-output data. This provides the fuzzy system with the ability to learn. It is clear that the network just described can easily be extended to a Sugeno fuzzy model with multiple input and rules. Again for a system with multiple output, models for each output can be built separately and then combined together to form an overall model.

7. HYBRID LEARNING ALGORITHM

From the proposed ANFIS architecture in Fig.5, it is observed that given the values of premise parameters, the overall output can be expressed as a linear combinations of the consequent parameters. More precisely, the output \( f \) can be rewritten as

\[
f = \frac{\omega_1}{\omega_1 + \omega_2} f_1 + \frac{\omega_2}{\omega_1 + \omega_2} f_2 \quad (11)
\]

\[
= \omega_1 f_1 + \omega_2 f_2 \quad (12)
\]

\[
= (\omega_0 x)p_1 + (\omega_0 y)q_1 + (\omega_0)\theta_1 + (\omega_1 x)p_2 + (\omega_1 y)q_2 + (\omega_1)\theta_2 \quad (13)
\]

which is linear in the consequent parameter \( \{p_1, q_1, r_1, p_2, q_2, r_2\} \). As a result, expression becomes

\[
H(\text{output}) = \quad (14)
\]

\[
H = \quad (15)
\]

Where,

\( S \) is the set of total parameters
\( S_0 \) is the set of premise parameters
\( S_i \) is the set of consequent parameters

In equation (15), \( H \) and \( F \) are identity function and the function of the fuzzy inference system, respectively. Therefore, the hybrid-learning algorithm can be applied directly. More specifically, in the forward pass of the hybrid learning algorithm, functional signals go forward till layer 5 and the consequent parameters are identified by the least square estimation. In the backward pass, the error rates propagate backward and the premise parameters are updated by the gradient descent.

As mentioned earlier, the consequent parameters thus identified are optimal under the condition that the premise parameters are fixed. Accordingly, the hybrid approach is much faster than the strict gradient descent and it is worthwhile to look for the possibility of decomposing the parameters set in the manner of equation (15). It should be noted that the computational complexity of the least square estimation is greater than that of the gradient descent method.

8. SIMULATION RESULTS OF ANFIS MODELING

An ideal approach to model a complex and vague dynamics in the machining operation is the recently developed ANFIS type of modeling. The tool wear is predicted by using this technique. Fig.6 (a) and Fig.6 (b) show the trained and checked output of ANFIS modeling. Fig.7 shows the comparison between measured and predicted tool wear(mm) by ANFIS modeling.
Figure 6 (a) Trained output of ANFIS modeling

Figure 6 (b) Checked output of ANFIS modeling

Figure 7 Degree of accuracy of tool wear prediction by ANFIS modeling

The performance of simulation results of various prediction methods by employing ANN and ANFIS are compared and presented in Table 2.

Table 2 Comparison of prediction errors

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Performance</td>
<td>ANN</td>
</tr>
<tr>
<td>1.</td>
<td>Training set size</td>
<td>35</td>
</tr>
<tr>
<td>2.</td>
<td>Checking set size</td>
<td>21</td>
</tr>
<tr>
<td>3.</td>
<td>Training RMSE</td>
<td>$1.3328 \times 10^{-4}$</td>
</tr>
<tr>
<td>4.</td>
<td>Checking RMSE</td>
<td>0.0296</td>
</tr>
<tr>
<td>5.</td>
<td>Training PME</td>
<td>$3.1214 \times 10^{-4}$</td>
</tr>
<tr>
<td>6.</td>
<td>Checking PME</td>
<td>0.6872</td>
</tr>
</tbody>
</table>

9. RESULTS AND DISCUSSION

A total of 56 tool wear tests are conducted under various cutting conditions. 35 samples are randomly picked as training samples and 21 samples are selected as test samples. The neural network technique is used for tool wear monitoring in turning. The backpropagation algorithm is used with spindle speed, feed rate, depth of cut and spindle motor current as input vectors and the tool wear as the output vector. The network is trained off-line using 35 patterns. The major advantage of the neural network predictions is that the models can estimate tool wear progress very fast and with accuracy. Once the inputs are known, it is found that the ANN algorithm successfully mapped the cutting conditions to the appropriate classes of tool wear states [19]. But the following problems need to be investigated in ANN.

In ANN, no theoretical guidance for the choice of the number of hidden layer in neural network is available. The accuracy of the predicted results varies greatly with different structures of the ANN and the accuracy of tool wear states does not improve when hidden layers are increased. When the number of hidden layers is more than 1, the learning speed of the ANN decreases, but the accuracy of the predicted tool wear states does not increase. Thus it is not appropriate to choose more than one hidden layer. For the choice of the number of neurons...
in the hidden layer, the principle is to choose a number as low as possible to simplify the network on the basis that the data from the input to output is mapped well. The method of selection is to set less number of neurons and measure the error in the output layer, then add a number of neurons according to the error until there is no appreciable reduction in error. Based on this principle, a maximum of four neurons is selected in the hidden layer. The learning efficiency (\(\mu\)) cannot be too small. Otherwise the learning speed will be too slow. A large values of \(\mu\) may cause the network to diverge. It is found that a suitable vale for \(\mu\) lies in the range of 0.85 to 0.9.

It must be emphasized that large number of training cycles does not mean a greater accuracy for the prediction of tool wear states. The purpose of training is to find an essential relationship between input and output data in the samples so that the parameters in the output layer can be predicted reasonably with the parameters in the input layer. However, because of unavoidable noise in the sample data, the larger the number of training cycles, the greater will be noise data in the network that will be recorded. Also the noise will affect the prediction of validated data. Therefore it is necessary to fix an upper limit to the training cycle number. When the upper limit is exceeded, the process of training the ANN will stop. When the value of RMSE increase with the increase of epochs, the number of training cycles at this moment is defined as upper limit.

The unification method of the data also influences the predicted results. The accuracy of the predicted tool wear states can be obtained when all of the data in the input layer and output layer range evenly from 0 to 1.

Neural networks can be trained but it is extremely difficult to use a prior knowledge about the system under consideration and it is almost impossible to explain the behaviour of the neural system in a particular situation. The training may sometimes cause temporal instability. To overcome this problem, ANFIS modeling is used. The integration of fuzzy systems and neural networks can combine the merits of both systems and offer a more powerful tool for modeling.

In ANFIS, no expert is available and number of membership functions assigned to each input variable is chosen empirically by combining the desired input and output data. It is found that effective partition of the input space decrease the corresponding rule and thus increase the speed in both learning and application phases. Without a priori knowledge, the initial parameter settings of ANFIS are intuitively reasonable. This leads to fast learning and increases the accuracy of predicted results. Because of the complexity of the process, modeling techniques are implemented for limited data. ANFIS modeling is especially suited for modeling with large amount of data. Furthermore, the data can be in any irregular or random form such as the daily operating data of the machining process. Firstly, it summarizes past data and there is no need to repeat the calculations of past data whenever new data becomes available. Secondly, it has the learning ability of neural networks with the newly available daily operating data. Thirdly, this type of modeling can estimate many parameters and even tune or modify the network structure. Thus the proposed approach is a versatile tool for modeling of the machining processes.

10. CONCLUSION

The present study has used two powerful learning algorithms ANN and ANFIS for predicting tool wear states under different cutting conditions. It is observed that a greater utility can be made of hybrid method,
combining the strength of both neural networks and fuzzy systems. The simulation results show that the proposed ANFIS modeling can effectively be employed for tool wear prediction.

REFERENCES


[16] R.H. Wu, J.T. Liu, H.B. Chang, T.Y. Hsu and X.Y. Ruan, “Prediction of Flow Stress of 0.4C-19Cr-
Adaptive - Network - Based Fuzzy Inference System for Tool Wear Prediction

1.5Mn-1.0Ni-0.2Mo Steel during Hot Deformation”, Journal of materials processing technology, Vol.116 (2001); pp.211-218.


Authors Biography

V. Kalaichelvi is working as Lecturer (SG) in the Department of Electronics & Instrumentation Engineering, Annamalai University. She is currently doing her Ph.D work in the area of control systems applied to material processing. Her research interest includes Control Systems, Process Control, Neural Networks and fuzzy Logics. She has published several research papers in National/International Conferences & Journals.

Dr. D. Sivakumar is working as Professor in the Department of Electronics & Instrumentation Engineering. He obtained his Ph.D in Control Systems from the PSG College of Technology, Bharathiyar University, Coimbatore. His research interest includes Control Systems, Process Control, Digital Signal Processing and Image Processing. He has published several research papers in National/International Conferences & Journals.

Dr. R. Karthikeyan is working as Professor in the Department of Manufacturing Engineering. His research interest includes Material Processing and Optimization Techniques. He has published several research papers in National/International conferences & Journals.
Towards Development of Quality Softwares using Reuse Reengineering Approach

U. Suman¹ Dr. (Mrs.) M. Ingle²

¹Lecturer, International Institute of Professional Studies, Devi Ahilya University, Indore (M.P.), India.
E-mail: ugrasen123@yahoo.com

²Reader, School of Computer Science, Devi Ahilya University, Indore (M.P.) - 452001, India.
E-mail: maya_ingle@rediffmail.com

ABSTRACT
The maintenance of existing procedural systems has gained momentum in the information society. These systems sometimes are unable to satisfy user’s requirements and at the same time their performance has been observed to be degraded as compared to modern technologies. The reengineering of obsolete procedural systems like legacy systems onto modern platform (e.g. object-oriented technology) enhances the reusability, maintainability, testability and hence the quality of software. On the other hand, development with reuse of existing knowledge has now become a promising practice for software development. It improves the quality as well as productivity, predictability and maintainability of the product. Hence, reuse reengineering has gripped a significant attention to software practitioners. In this paper, we propose a reuse reengineering process, which is mainly twofold; Firstly, to streamline the obsolete systems by making them reusable for future software development. Secondly, to design a candidate system with reuse of existing tested and certified artifacts. Thus, reuse reengineering process produces a high quality product with flexible software architecture in a timely and cost-effective way. Candidate components from existing or developing system are identified, reengineered, qualified, categorized and then managed in a single reuse repository system. The elected components can be used to develop a candidate system.

Keywords: Quality, Reengineering, Software reuse, RRP, Component, Reuse repository, qualification, classification, software architecture, domain analysis.

1. INTRODUCTION
The development and maintenance of softwares in an organization is very costly and time consuming. There are several procedural systems (e.g. legacy systems) exist in the world as the critical assets for most of the organizations. These systems are too difficult to modify and too important to be discarded. Therefore, these systems need to be reengineered onto modern platform for better operating efficiency and maintenance. Reengineering of a system can make it more reusable and at the same time generates an alternative view about software product to work under the new environment [17]. The goal of Reuse Reengineering Process (RRP) is to produce a high quality product with reduced cost and time. The RRP involves two aspects: One is to reengineer software products for reuse and other is to develop new softwares with reuse of existing tested and certified artifacts. Many reuse and reengineering processes have been discussed with phases such as candidature, election, qualification, classification and storage, search and display [2][12].
Towards Development of Quality Softwares using Reuse Reengineering Approach

We propose RRP consisting of six phases, namely: identification, election, qualification, classification and management, retrieval and integration phase. This process creates reusable components in a cost effective way, which implements various levels of abstraction within existing systems. A component is an encapsulated, distributed and executable piece of software that provides and receives services with defined interfaces [11]. Domain engineering plays an important role to identify highly reusable components. RRP involves the identification and extraction of reusable components through reengineering i.e. reverse engineering techniques. The object oriented reverse engineering approach for procedural system can be used in RRP to transform procedural systems onto modern platform [17]. This approach uses XML (i.e. Extensible Markup Language) as a portable source code representation in which source code is represented at an abstract level [5]. This kind of representation requires analyzing syntax and semantic rules as well as the domain model of the language being used in earlier systems. It identifies the functional equivalent language constructs and aggregates them at a higher level of abstraction. A language domain model representation of source code can be represented in the form of Abstract Syntax Tree (AST) for an expression syntax tree. From domain model, a logical structure of the entities of an AST is constructed in terms of XML elements and Document Type Definition (DTD). These logical structures are directly mapped to objects and corresponding properties to model the system for modern platforms. The reengineering phase thus takes great effort to transform onto different platform.

Reengineered components can be coupled with the components of candidate system with clear interface specifications. An interface of a component is used to pass information to and from other components. An interface of a component can be specified by the data type, signature of the operations, etc. The degree of coupling among components may be measured through the complexity of interfaces. Components with well-defined interfaces need to be managed for further use. The reuse repository system can be designed to manage these qualified and classified components. However, These managed components can be used for further development in an effective manner [10][18].

The proposed RRP is a promising practice of developing new software systems from existing software knowledge. This process covers domain engineering, reverse engineering, reuse repositories, reuse certificate, generic architecture and reuse measurement. Development with reuse is widely considered as a way to simplify and speedup the software development. At the same time, it also reduces the cost and time to market, to improve the quality and reliability of software [18]. The set of components that are reusable exists in a variety of forms, i.e. requirement document, design models, programs, test cases and test procedures, documentation, prototype architectures, etc.

The RRP takes the considerable amount of efforts to identify reusable components thereby improving the quality and reliability of the product [1]. In Section 2, we describe the RRP and the working mechanism of various phases with the help of a case study. The impact of RRP and reuse of the artifacts extracted from the used case study is also covered in this section. Reuse reengineering and software architecture together shape a standard and reliable products, which is discussed in Section 3. RRP can be employed from a small scale to large-scale projects to produce a high quality product with more advantageous form of existing system and is described in Section 4. Section 5 discusses certain factors such as managerial, economical, legal and technical related to RRP. Finally, we conclude with concluding remarks in Section 6.
2. REUSE REENGINEERING PROCESS

Reuse Reengineering is a systematic process of producing a high quality product in a cost effective manner. The success of RRP depends on the use of a well-defined organizational paradigm to identify and classify the theoretical, methodological and technological problems involved in it. Therefore, the advantages and disadvantages of any solution can be identified and evaluated. This process uses some existing source code for reengineering and produces reusable artifacts for future development or new requirement that can be developed with reusable artifacts. The reusable artifacts are stored and retrieved as and when required for the development of new software to produce a high quality product in a cost effective manner [14][3].

The RRP takes existing systems or new requirements as an input and produces a high quality reengineered product. The phases involved in RRP are namely: identification of candidates, election of candidates, qualification, classification and management of reusable components, retrieval from repository and integration in software development. These components can be integrated from the various kinds of similar problem areas. RRP cycle is shown along with its phases and the mechanisms performed in Figure 1.

![Figure 1: Reuse Reengineering Process (RRP)](image-url)
2.1 A CASE STUDY: AirportSimulation System

We consider a procedural program for AirportSimulation written in Turbo Pascal as an input, which is shown belows. This program uses queuing system for landing and taking off planes. Planes arrive ready to land or to take off at random times, so at any given unit of time, the runway may be idle or a plane may be landing or taking off and there may be several planes waiting either to land or take off.

```pascal
program AirportSimulation(input, output)
uses PlaneQueue, RandomGen;

var
landing, takeoff : queue;
curplane : plane;
expectarrive, expectdepart : real;
curtime, nplanes, endtime, idletime,
land, takeoff, nrefuse,
landwait, takeoffwait, counter : integer;

begin
CreateQueue(landing);
CreateQueue(takeoff);
Start(endtime, nplanes, nland, ntakeoff, nrefuse,
landwait, takeoffwait, idletime, expectarrive,
expectdepart);

for curtime := 1 to endtime do begin
for counter := 1 to Prandom(expectarrive) do begin
NewPlane(curplane, nplanes, curtime, arrive);
if Queuefull(landing) then
Refuse(curplane, depart, nrefuse)
else
Append(curplane, takeoff)
end;

for counter := 1 to Prandom(expectdepart) do begin
NewPlane(curplane, nplanes, curtime, depart);
if Queuefull(takeoff) then
Refuse(curplane, depart, nrefuse)
else
Append(curplane, takeoff)
end;
end;
end;

if not QueueEmpty(landing) then begin
Serve(curplane, landing);
Land(curplane, curtime, nland, landwait)
end
else if not QueueEmpty(takeoff) then begin
Serve(curplane, takeoff);
Fly(curplane, curtime, ntakeoff, takeoffwait)
end
else
idle(curtime, idletime);

Conclude(nplanes, nland, ntakeoff, nrefuse, landwait,
takeoffwait, idletime, takeoff, landing)
end.

Identification: In this phase, the existing system or new requirements acts as an input and analysis is performed to identify a set of potential candidates. The candidates can be in the form of requirement specifications, systems, subsystems, design abstractions, modules, procedures, members, data (scalar items), structures (compound variables), test procedures, libraries, etc. It involves domain analysis and other methods to identify reusable candidates [4]. These candidates are input to election phase to reverse engineering onto different platform as object oriented platform. The potential candidates identified in AirportSimulation program are shown in Table 1.

Election: The identified candidates act as an input and reverse engineering is performed to transform them onto different platform in this phase. There exists various reverse engineering approaches and metrics that can be
used to identify potential candidates for reverse engineering and generate reusable components. The existing candidates must be transformed at an abstract level of representation with its boundaries and interfaces [10][21]. Most of the systems, which have been developed, were written in procedural languages and now they need to be transformed onto modern object oriented technologies. Object oriented reverse engineering approach for procedural systems can be used to reengineer AirportSimulation program onto object-oriented platform [17]. In this approach, we use XML as an intermediate code representation whereby a generic code can be produced. The program must be written in generic fashion instead of writing for the specific system because the artifacts may be used for similar kinds of problem area.

<table>
<thead>
<tr>
<th>Potential candidates</th>
<th>Identified Candidates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modules/Files</td>
<td>AirportSimulation, PlaneQueue, RandomGen.</td>
</tr>
<tr>
<td>Procedures</td>
<td>CreateQueue(), Refuse(), Start(), NewPlane(), Append(), Serve(), Land(), Fly(), Idle(), Conclude()</td>
</tr>
<tr>
<td>Data</td>
<td>landing, takeoff, counter, landwait, expectdepart, expectarrive, curtime, nplaned, endtime, idletime, curplane, nland, ntakeoff, nrefuse, takeoffwait.</td>
</tr>
<tr>
<td>Enumerated data</td>
<td>Action: arrive, depart</td>
</tr>
</tbody>
</table>

The AirportSimulation module written in Pascal can be reengineered onto object-oriented paradigms such as C++, Java, etc. We have used C++ representation for AirportSimulation problem. There are various approaches for reengineering and transforming modules in a generic fashion. Object oriented paradigm provide template like facilities for writing programs in a generic form. All the constructs, properties and operations used in the procedural program need to be transformed onto another platform. In some approaches, scalar data items can be transformed directly from procedural paradigm to C++. The compound data items such as records, enumerated data and other user-defined items can be transformed into C++ using constructors of a class and objects [10][21].

```c++
/*AirportSimulation Record Transformation*/

#include"PlaneQueue.cpp"
#include"RandomGen.cpp"
#include<iostream.h>
class AirportSimulation
{  private:
    int nplane, nland, ntakeoff, idletime, nrefuse, landwait, takeoffwait;
    float expectarrive, expectwait;
    public:
        Queue landing, takeoff;
        AirportSimulation(int endtime, int nplane)
        {
            /* Assign initial values to the variables used*/
        }
        float curtime, endtime;
        void Append_land(int x);
        void Append_takeoff(int x);
        void Refuse(int x, int c);
        int Serve(int c);
        void Land(int x);
        void Fly(int x);
        void Idle();
    ~AirportSimulation()
    {
        cout<<endtime<<nplane<<nland<<ntakeoff<<nrefuse;
    }
}
Towards Development of Quality Softwares using Reuse Reengineering Approach

In AirportSimulation program, the Start() and Conclude() procedures can be transformed as constructor and destructor. PlaneQueue and RandomGen units can be designed separately and included here to generate queues and random numbers. The prototypes declared in this class can be used in the main program as follows.

```c
#define TRUE 1
#define FALSE 0

void main()
{
    int plane;
    AirportSimulation APS;
    for(int i = curtime; i <= endtime; i++)
    {
        for(int j = 0; j <= expectarrive; j++)
        {
            plane = getvalue(); /* get values from users and initializes all variables specified as parameters */
            if(APS.Landing.isfull() == FALSE)
                APS.Append_land(plane);
            else
                APS.Refuse(plane, 0);
        }
        for(int j = 0; j <= expectdepart; j++)
        {
            plane = getvalue();
            if(APS.takeoff.isfull() == FALSE)
                APS.Append_takeoff(plane);
            else
                APS.Refuse(plane, 1);
        }
    }
    while (APS.Landing.isempty() != TRUE)
    {
        plane = APS.Serve(0);
        APS.Land(plane);
    }
    while (APS.takeoff.isempty() != TRUE)
    {
        plane = APS.Serve(1);
        APS.Fly(plane);
    }
    APS.Idle();
}
```

The other procedures used in this program can be reengineered in the same way as the main program. The important considerations should be given on data variables in transformation.

**Qualification:** This phase groups together the activities that produces the specifications of each one of the candidates elected in the previous phase. The interface and functional specifications must be produced [11]. Interface specifications describe the external behavior of a class, object or other entities such as modules, subsystems, etc. Interface specification also measures the coupling of the module. In case of a class or object, the interface includes the signatures of the operations and access specifications. The signature of an operation maps the number of arguments, return types, and types of arguments. Object oriented programming provides private, public and protected access specifiers for the visibility of members and operations of one class to another [15]. Qualification specifies the pre conditions and post conditions of an expression or a statement. It has a major effect on extension for subclasses such as library calls and other functions. In AirportSimulation problem, the following library facilities observed:

```c
#include"PlaneQueue.cpp"
#include"RandomGen.cpp"
#include<sstream.h>
```

**Classification and management:** In this phase, the elected and qualified components can be used as input and classified according to the reference taxonomy of the application domain. The classification and management of components is important for easy maintenance, storage, retrieval and searching. These components can be stored in a single reuse repository system instead of managing them elsewhere. The repository system keeps separate tracks of components.
classified according to the functional description or application areas for future purpose. The indexing, sequencing, hashing and storage mechanisms play a major role for efficient classification and management of reusable components [16].

**Retrieval:** The user’s queries for intended component acts as an input and searching techniques can be used to find the required components from the reuse repository as easy as possible in this phase. For the component to be used, one can also provide a visual support for the developers to search and verification of the components through the reuse repository system. An active reuse repository system that uses a prototype called Codebroker can be used to retrieve reusable components effectively [22]. Active reuse repository system involves the listener, fetcher and presenter to retrieve proper information for the component.

**Integration:** In this phase, the retrieved components are assembled with the system development, as and when required. These components can be used at any stage of system development life cycle, i.e. requirement analysis, design, development, testing and documentation. The integration phase greatly emphasizes on signature matching compatibility. Signature matching is the process of determining the compatibility of two components. In this stage, certain type constraints are checked for reliability of the candidate system [11][10].

### 2.2 Impact Of RRP On Software Quality

Simulation systems are useful for analyzing the behavior of a system with the help of computation of related statistics/parameters, finding bottlenecks, detecting scheduling errors, etc. The *AirportSimulation* system described in the previous section is analyzing the dynamics of planes at airport. The operations involved in this system can be reused in the implementation of any event driven simulated environments such as time-sharing systems, traffic system, grocery checkout system, etc.

Thus, entire *AirportSimulation* system with reused modules PlaneQueue, RandomGen can be utilized for another systems. Specifically, the operations CreateQueue(), Refuse(), Start(), NewPlane(), Append(), Serve(), Land(), Fly(), Idle(), Conclude() can be reused to observe the behavior of any other systems.

The execution of RRP reflects the quality of products. The reengineering of procedural system, as shown in *AirportSimulation* system on to object-oriented code makes the maintenance task easier. Object oriented code can be made reusable by designing them in generic fashion. Developments with reuse products are much reliable because of usage of tested and certified components thereby saving development cost and time to market the product and hence productivity. Also, it is possible to estimate the cost, time and effort before development. Software products developed with reusable components are able to operate in a distributed environment such as Java, CORBA, etc. thereby increasing the efficiency and the reliability of the product.

### 3. SOFTWARE ARCHITECTURES AND REUSE REENGINEERING

Software architecture provides a pattern whereby the understanding and integrity of both existing and future systems is shared and maintained. Standardized architectures define frameworks to meet requirements of similar systems with increasing predictability of the product and process at early stages of development, while decreasing defects, costs and time. But architectural decisions are made in the early stages of the development life cycle. If suitable architectures are not chosen, it may neither be possible to satisfy systems requirements nor to effect major design changes. Good architectures are necessary to produce high quality software and at the same time increasing development, productivity, and
Towards Development of Quality Softwares using Reuse Reengineering Approach

Software architecture provides a means to structure knowledge of systems within a given domain to include requirement, design and implementation. A domain is an area of knowledge or activity characterized by a family of related systems. Domain specific software architectures with standard interfaces are central to the domain specific reuse, thereby providing a framework to create reusable components and constructing systems within a specified domain. Software architecture can be reused among the various contexts of a single system through its maintenance cycle. The various styles of architectures can be produced through RRP and managed in a single reuse repository system. It is necessary to design architectures in a general fashion rather than writing code for specific software. The most frequently used architectures are: communication processes, main program and its modules, layered system, event system, sequential execution steps, transactional databases and so on [13]. A suitable architecture increases the reliability and the quality of the reengineered products.

4. REUSE REENGINEERING BASED SYSTEMS

It is true that each approach has its own benefits and limitations. RRP has also its advantages and some problems. There exists several systems, which have been developed in earlier days using conventional method of development has many limitations. Most of them are procedural systems, i.e. legacy systems developed with procedural paradigms and languages. Now these systems are unable to satisfy users requirements, difficult to maintain and difficult to modify and enhance. The maintenance cost and resources required increases with repeated modification and decreases its reliability and quality. This is tedious and time consuming also. These systems are unable to reuse thereby decreasing the productivity and predictability also. Such systems have been becoming obsolete with rapid growth of technology [19].

RRP based development uses an efficient and modern technology that produces a highest quality products with increasing productivity and predictability. At the same time it also decreases maintenance cost and time, enhance reusability, maintainability, testability and scalability of the system. Formal reuse reengineering requires common standards, procedures and produces across a given domain. In RRP, the reusable components are systematically analyzed, well reengineered and well qualified. In addition, these components must be well managed in a repository system and one can retrieve them through well searching mechanisms. The retrieval stage uses well-defined criterions such as isolation, aggregation and generalization to extract suitable component from the repository system [9]. The identification phase uses two well-defined criteria: completeness and adequacy. Completeness is the extent to identify all the potentially reusable components. Adequacy measures its ability to identify only those components that will actually be elected for reuse [8][20].

However, RRP based development has some problems. There is a lack of required standardization in the different areas of the software life cycle. It needs full support of management at all phases of development. The extra efforts employed for training the reuse groups and require well skilled expertise to manage reusable components, system understanding for reverse reengineering and proper integration of components in the development [20].
5. Reuse Reengineering Affecting Factors

There are various issues that can affect the success of RRP. The major factors are: managerial, economical, legal and technical [18][7]. RRP needs management support at each phase whereby developers and reuse group share their skills with management group and learn habitual things about the organization. It needs training of management and technical expertise of the RRP team. Therefore, development with RRP requires management support from IT and other departments of the organization for the success of reuse reengineering. Economic factors measure the reuse rate of highly reusable components and domain of related systems. It involves various cost-benefit analysis, performance measurements to identify for highly reusable components, pricing and other support costs. Legal factors permit the rights and responsibilities for common use of reusable things to the people. It involves the software copyright and proprietary issues, liabilities and responsibilities, contractual requirements for reuse. There are various technical factors that must be required for RRP such as: domain analysis, methodologies, creation, environment, repository, transformation and languages. Hence these factors play their important roles to the success of RRP.

6. Conclusion

It has been observed that RRP has many benefits that have been practiced in several areas. Reengineered systems also provide potential benefits and streamlined to modern technologies. We have discussed about various phases of RRP, the quality of product produced, and the role of software architecture for giving standardization to reusable products. We observed that the domain analysis is the heart of RRP that leads to the successive phases. It identifies potential candidates in existing systems. Reverse engineering is a crucial technical phase that takes too much effort for producing an efficient and suitable representation of the source code to manufacture good quality reusable components. Another important aspect of RRP is the classification and management of reusable components. There are various techniques that can be used for managing reusable components. One of them is a Codebroker, an efficient technique for managing a single repository system whereby reusable components can be retrieved and used as simple way as possible. RRP requires careful planning and techniques to integrate reusable components in systems development for easy understanding and reliability. It also requires skills of developers and an intelligent mechanism.

Software architectures provide standardization to produce a quality and reliable product. Thus, designing a suitable architecture, RRP provides a major role to produce components and their interfaces to integrate them with other components. A well management team, skilled developers, resource availability, etc. are prerequisites to achieve a successful reuse reengineering based development. The RRP has many benefits but its implementation may involve risk. If it is not implemented successfully, it may cost of an organization precious time and resources. On the other hand, RRP may be the key to improving the quality of software and reducing the cost of development and provide the competitive advantages. But this approach requires initial investment and regular maintenance of reuse repository system. It doesn’t provide mechanisms for reuse effort estimation, classification and storage and integration.
REFERENCES


Author’s Biography

Dr. (Mrs.) M. Ingle has received Ph. D. in Computer Science and M. Sc. in Statistics from Devi Ahilya University. She also received M.Tech in computer science from IIT, Kharagpur. She has around 25 publications in National and International Conferences and Journals. She is guiding 3 research scholars. She has also guided many PG students for projects since last 20 years.

Mr. U. Suman holds MCA and pursuing Ph. D. in Computer Science from Devi Ahilya University. He has published 5 research papers in National and International Conferences. He has guided around 20 students for project since last 5 years.
FFT based DCT-LMS Speech Enhancement for Sensorineural Loss Patients

Sunitha S. L. 1  Dr. V. Udayashankara 2

1. INTRODUCTION

Hearing impairment is the number one chronic disability affecting many people in the world. Background noise is particularly damaging to speech intelligibility for people with hearing loss especially for sensorineural loss patients. Several investigations on speech intelligibility have demonstrated sensorineural loss patients need 5-15 dB higher SNR than the normal hearing subjects. This paper describes FFT based Discrete Cosine Transform Power Normalized Least Mean Square (Fast DCT-LMS) algorithm to improve the SNR and to reduce the convergence time of the LMS for sensorineural loss patients. This type of fast algorithm requires \[ N \left( \frac{1}{4} \log_2 N + 1 \right) \] real multiplications and \[ N \left( \frac{3}{4} \log_2 N - 3 \right) \] real additions. The computer simulations results show superior convergence characteristics of the proposed algorithm by improving the SNR at least 10 dB for input SNR’s less than and equal to 0 dB, faster convergence speed and better time and frequency characteristics.

Keywords: Hearing Impairment, Fast DCT Adaptive filter, Sensorineural loss, Convergence rate, SNR improvement.

1Research Scholar, JSS Research Foundation, SJCE Campus, Mysore-570006
E-mail: Sunithanov27@yahoo.co.in

2Research Scientist, JSS Research Foundation, SJCE Campus, Mysore-570006
E-mail: udaya@sjce.ac.in

ABSTRACT

Hearing impairment is the number one chronic disability affecting many people in the world. Background noise is particularly damaging to speech intelligibility for people with hearing loss especially for sensorineural loss patients. Several investigations on speech intelligibility have demonstrated sensorineural loss patients need 5-15 dB higher SNR than the normal hearing subjects. This paper describes FFT based Discrete Cosine Transform Power Normalized Least Mean Square (Fast DCT-LMS) algorithm to improve the SNR and to reduce the convergence time of the LMS for sensorineural loss patients. This type of fast algorithm requires \[ N \left( \frac{1}{4} \log_2 N + 1 \right) \] real multiplications and \[ N \left( \frac{3}{4} \log_2 N - 3 \right) \] real additions. The computer simulations results show superior convergence characteristics of the proposed algorithm by improving the SNR at least 10 dB for input SNR’s less than and equal to 0 dB, faster convergence speed and better time and frequency characteristics.

Keywords: Hearing Impairment, Fast DCT Adaptive filter, Sensorineural loss, Convergence rate, SNR improvement.

1Research Scholar, JSS Research Foundation, SJCE Campus, Mysore-570006
E-mail: Sunithanov27@yahoo.co.in

2Research Scientist, JSS Research Foundation, SJCE Campus, Mysore-570006
E-mail: udaya@sjce.ac.in

1. INTRODUCTION

Hearing impairment is the number one chronic disability affecting many people in the world. Background noise is particularly damaging to speech intelligibility for people with hearing loss especially for sensorineural loss patients. Several investigations on speech intelligibility have demonstrated sensorineural loss patients need 5-15 dB higher SNR than the normal hearing subjects. This paper describes FFT based Discrete Cosine Transform Power Normalized Least Mean Square (Fast DCT-LMS) algorithm to improve the SNR and to reduce the convergence time of the LMS for sensorineural loss patients. This type of fast algorithm requires \[ N \left( \frac{1}{4} \log_2 N + 1 \right) \] real multiplications and \[ N \left( \frac{3}{4} \log_2 N - 3 \right) \] real additions. The computer simulations results show superior convergence characteristics of the proposed algorithm by improving the SNR at least 10 dB for input SNR’s less than and equal to 0 dB, faster convergence speed and better time and frequency characteristics.

Keywords: Hearing Impairment, Fast DCT Adaptive filter, Sensorineural loss, Convergence rate, SNR improvement.
ear may not be sufficient to accommodate the range of intensities in speech signals. So, the stronger components of speech are perceived at a level, which is uncomfortably loud, while the weaker components are not heard at all [10], [11], [16].

Several investigations on speech intelligibility have demonstrated that subjects with sensorineural loss patients need 5 to 15db higher SNR than the normal hearing subjects. While most of the defects in transmission chain up to cochlea can now-a-days be successfully rehabilitated by means of surgery. The great majority of the remaining inoperable cases are sensorineural hearing impaired patients [5], [16]. Digital technology has made an important contribution in the field of audio logy. Digital signal processing methods offer great potential for designing a hearing aid but, today’s Digital Hearing Aid are not up to the expectation for sensorineural loss patients. Hearing-impaired patients applying for hearing aid reveal that more than 50% are due to sensorineural loss. So for only Adaptive filtering methods are suggested in the literature for the minimization of noise from the speech signal for sensorineural loss patients [8].

Adaptive Filtering Method:
The least mean square algorithm was first introduced by Widrow and Hoff in 1959 is simple, robust and is one of the most widely used algorithm for adaptive filtering. LMS algorithm is very popular because of its simplicity and easy of computations. LMS algorithm is generally the best choice for many different applications [18], [19]. This method can be effectively applied to reduce the noise i.e. to improve the SNR for sensorineural loss patients [6], [12], [15]. Unfortunately, its convergence rate is highly dependent on the feedback coefficient µ and the input power to the adaptive filter [18], [19]. The mean square error of an adaptive filter trained with LMS decreases over time as a sum of exponentials whose time constants are inversely proportional to the eigenvalues of the autocorrelation matrix of the filter inputs. Therefore, small eigenvalues create slow convergence modes in the means square error function. Large on the other hand, put a limit on the maximum learning rate that can be chosen without encountering stability problems [1]-[3].

In this work we use Fast DCT-LMS algorithm to improve the SNR and to reduce the convergence rate of the LMS for sensorineural loss patients. DCT-LMS algorithm is suited for non-stationary inputs like speech signals and the convergence time is also less compare to direct LMS techniques and Transform domain LMS like DFT-LMS and DWT-LMS [17], [20]. DCT is orthonormal, separable, frequency basis much like a Fourier transform [18]. The DCT has a strong energy compaction property. Most of the signal information tends to be concentrated in a few low frequency components of the DCT. It is a close relative of DFT – a technique for converting a signal into elementary frequency components, and thus DCT can be computed with a Fast Fourier Transform. Unlike DFT, DCT is a real valued and provides a better approximation of a signal with fewer coefficients. The DCT is central to many kinds of signal processing but DCT is mainly used in image processing application. DCT is very popular and used extensively in current image compression algorithms [18]. For non-stationary signals like speech signals, the DCT provides good approximation of a signal with fewer coefficients [3], [4].

When the adaptive filter is implemented as a tapped delay line operating on the entire available input signal bandwidth, selection of a single value of µ is required. Then, the algorithm convergence time and stability depends upon the ratio of the largest to the smallest...
eigenvalues associated with the correlation matrix of the input sequence. As the eigenvalue spread of the input autocorrelation matrix increases, the convergence speed of LMS deteriorates. The best convergence properties are obtained when all the eigenvalues are equal. That is, when the input auto correlation matrix is proportional to the identity matrix. In that case, the inputs are perfectly un-correlated and have equal power. As the eigenvalue spread of the input auto correlation matrix increases, the convergence speed of LMS deteriorates.

In this work the flow of input samples is continuously transformed by a fixed data-independent transform. That is meant to de-correlate the input signals, this pre processing followed by a power normalization stage causes the eigenvalues of the LMS filter inputs to cluster around one and speeds up the convergence of the adaptive weights. So in this case, we derive the eigenvalue distribution of the auto correlation matrix after DCT and power normalization. This provides the good tracking capabilities in non-stationary environments. In section 1, we briefly discussed about the sensorineural loss patients and brief review about the convergence rate of the LMS adaptive algorithm. Section 2, considers LMS filtering in Discrete Cosine Transform domain. Simulated results are discussed in section 3 and section 4 concludes the paper.

The Fast DCT by Using FFT:

A block transform based on the DCT or DFT is equivalent to a filter bank consisting of multiple band pass filters. Such a filter bank is called as a Time Domain Aliasing Cancellation Filter bank or TDAC Filter bank. Additionally this method provides an added advantage in the block processing of discrete time signals such as audio and speech. The next step in designing a fast algorithm based on a TDAC filter bank is to select the transform that will be used. A number of transforms that have a meaningful interpretation in the transform domain are used in the digital processing of signals. More notably are the DFT, DCT, etc. Of these it was found that the DCT offers maximum advantages [1], [17] and [20]. To decide the best transform the performance of different transforms is generally compare with that of the Karhunen-Loeve Transform (KLT) which is known to be optimal with respect to the following performance measures: variance distribution, estimation using the mean-square error criterion, and the rate-distortion function. Although the KLT is optimal, it is a statistical transform whose basis vectors change with a change in the source distribution. Also there is no general algorithm that enables its fast computation. On comparison using the above parameters it was found that the DCT performance compares more closely with that of the KLT relative to the performances of the DFT, WHT, and HT.

The DCT of a data sequence \( x(m) \) is defined as:

\[
x(m), m = 0, 1, \ldots , (M - 1)
\]

\[
G_A(0) = \frac{\sqrt{2}}{M} \sum_{m=0}^{M-1} x(m)
\]

\[
G_A(k) = \frac{2}{M} \sum_{m=0}^{M-1} x(m) \cos \left( \frac{(2m+1)k\pi}{2M} \right)
\]

(1)

Where \( G_A(k) \) is the kth DCT coefficient. It can be seen that the set of basis vectors \( \left\{ \frac{1}{\sqrt{2}} \cos \left( \frac{(2m+1)k\pi}{2M} \right) \right\} \) is actually a class of discrete Chebyshev polynomials. Similarly the IDCT is:

\[
x(m) = \frac{1}{\sqrt{2}} G_A(0) + \sum_{k=1}^{M-1} G_A(k) \cos \left( \frac{(2m+1)k\pi}{2M} \right)
\]

(2)

The basis set actually used in TDAC systems is slightly modified and takes on the form:

\[
X(k) = 2 \sum_{n=0}^{\frac{N-1}{2}} x(n) \cos \left( \frac{2\pi}{N} (n + N_0)(k + \frac{1}{2}) \right), 0 \leq k \leq N/2
\]

(3)
Where \( z(n) \) is the windowed input sequence.

\[
x(n) = \sum_{k=0}^{N-1} X(k) \cos \left( \frac{2\pi (n + n_0) k}{N} \right), 0 \leq n \leq N
\]

(4)

Where, \( X(k) \) is the spectral co-efficient. In both the expressions \( n_0 = (N/2 + 1)/2 \). Such TDAC with the DCT as the transform is called as the Modified Discrete Cosine Transform or MDCT. The entire transform requires \( \frac{N^2}{2} \) real multiplications and \( \frac{N(N-1)}{2} \) real additions. Thus the complexity of computations is of the order \( O(N^2) \). Similarly for the IMDCT, we require \( \frac{N^2}{2} \) real multiplications and \( N(N - 1)/2 \) real additions, again a complexity of \( O(N^2) \). Ideally we would like computation times logarithmic or at least linear in the size of the input block length to make the use of these transforms feasible in real time signals. This in turn motivates us to look for algorithm, which computes the MDCT and the IMDCT expressions as fast as efficiently as possible. Fast algorithm uses an FFT like decimation to compute the MDCT in an efficient way. We start with the IMDCT expression,

\[
x[n] = \frac{1}{2C_N(n)} \sum_{k=0}^{N-1} \left[ X[k-1] + X[k] \right] C_N[n,k]
\]

(5)

Where

\[
C_N[n,k] = \cos \left( \frac{\pi}{2N} (2n + 1 + N/2)(2k + 1) \right)
\]

Further decomposition requires an additional manipulation similar to that of the decimation in time FFT. The whole process is repeated as many times as necessary to completely decompose the IMDCT into several sets of two point transforms. This type of fast algorithm requires \( \frac{N}{4} (\log_2 N + 1) \) real multiplications and \( \frac{N}{4} (3\log_2 N - 3) \) Real additions.

In section 1, we briefly discussed about the Sensorineural loss patients and brief review about the convergence rate of the LMS adaptive algorithm. Section 2, considers LMS filtering in DCT domain. Simulated results are discussed in section 3 and section 4 concludes the paper.

2. Fast DCT-LMS:

DCT-LMS is composed of three stages as shown in Figure 1.

Stage 1: Transformation of Input Signal by DCT

The input to the filter is

\[
x_k = [x_k, x_{k-1}, \ldots, x_{k-n+1}]^T
\]

(1)

This vector is processed by a unitary discrete cosine transform. Once the filter order \( N \) is fixed, the transform is just an \( N \times N \) matrix \( T \) with orthogonal rows. We have
orthogonal transform matrix $T$ such that the transform matrix $T^T$ is selected to be a unitary matrix,
$$i.e. \quad T_n^T T_n = T_n^T T_n = I \quad \cdots \cdots (2)$$
We assume that the input signals of the filter are real-valued and the elements of $T$ are also real valued [3]

Transforming a input signal (1) by a matrix $T_n$ transforms its Toeplitz autocorrelation matrix
$$R_n = E[x_n^T x_n] \quad \cdots \cdots (3)$$
into a non Toeplitz matrix
$$B_n = E[T_n^T T_n x_n^T x_n] = T_n^T R_n T_n \quad \cdots \cdots (4)$$
The transformation operation is
$$u_k(n) = T_n[x_n] \quad \cdots \cdots (5)$$
The transform outputs then form a vector
$$u_k(n) = [u_k(0), u_k(1), \ldots, u_k(n-1)]^T$$

**Stage 2: Power Normalization**

The transformed signal $u_k(i)$ is then normalized by the square root of their power $p_k(i)$. Where $i = 0, 1, \ldots, n-1$. The powers can be estimated by the following methods

1. The powers $p_k(i)$ can be estimated by filtering the $u_k^2(i)$ with an exponentially decaying window of parameter $\beta \in (0, 1)$

2. The powers can also be estimated based on a sliding rectangular window or with the help of an arbitrary weighting filter.

In this work, power normalization is as follow:

Power normalizing $T_n x_n$ transforms its elements
$$\left( T_n x_n \right)(i) \quad \begin{array}{c} \text{into} \\ \sqrt{\text{Power of} \left( T_n x_n \right)(i)} \end{array} \quad \cdots \cdots (6)$$

Where the power of $\left( T_n x_n \right)(i)$ can be found on the main diagonal of $B_n$.

Then the power-normalized signal is
$$v_k(i) = \frac{u_k(i)}{\sqrt{p_k(i) + \epsilon}} \quad \cdots \cdots (7)$$

Where $p_k(i) = \beta p_{k-1}(i) + (1 - \beta) u_k^2(i) \quad \cdots \cdots (8)$

for $i = 0, 1, \ldots, n-1$. The small constant $\epsilon$ is introduced to avoid numerical instabilities when is close to zero. The signals $v_k(i)$ are equal to the discrete cosine transformed outputs $u_k(i)$, but the learning constant $\mu$ in LMS filtering is replaced by a diagonal matrix whose elements are proportional to the inverse of the powers. This type of LMS is referred to as **power-normalized LMS**. Discrete cosine transformation followed by a power normalization stage, causes the eigenvalues of the LMS filter inputs to cluster around one and speeds up the convergence of the adaptive weights. The autocorrelation matrix after transformation and power normalization is thus
$$S_k \approx E(\text{diag} B_n)^{-1/2} B_n (\text{diag} B_n)^{1/2} \quad \cdots \cdots (9)$$

If $T_n$ decorrelated $x_n$ exactly, $B_n$ would be diagonal, $S_k$ would be an identity matrix $I_n$, and all the eigenvalues of $S_k$ would be equal to one, but since practically the DCT is not a perfect decorrelator, this does not work out exactly [2]. But the power normalization makes the eigenvalues of the LMS filter inputs to cluster around one and speeds up the convergence of adaptive weights.

The output vector after power normalization is
$$v_k(n) = [v_k(0), v_k(1), \ldots, v_k(n-1)]^T \quad \cdots \cdots (10)$$

**Stage 3: LMS filtering**

The resulting equal power signals $v_k(i)$ are applied as an input to an adaptive linear combiner whose weights
\( w_i(i) \) are adjusted using LMS algorithm described below. The weight vector is defined as

\[
\begin{align*}
w(n) &= [w_0(n), w_1(n), \ldots, w_{n-1}(n)]^T \\
\end{align*}
\]

Then the filter output is given by

\[
y_k(n) = w_k^T(n)v_k(n) \\
\]

and the instantaneous output error is

\[
e_k = d_k - \sum_{i=0}^{n-1} y_k(i) \\
\]

Where is \( d_k \) the desired signal. This error is used to update the adaptive filter taps using a modified form of the LMS algorithm

\[
w_{k+1}(i) = w_k(i) + \mu e_k v_k(i) \\
\]

The parameters used in algorithm are:

- Samples=20000, \( \alpha=0.45 \), \( \mu=0.075 \), and filter order=32.

3. SIMULATED RESULTS

The algorithm works on the corrupted speech signals with different types of noise signals like cafeteria noise, low frequency noise; babble noise etc. in different SNR’s. The various parameters like \( \alpha \), \( \mu \), and filter order were changed and their influence has been checked. A more meaningful quantity is the eigenvalue Spread that is calculated to find out how well the algorithm convergence to the optimum Wiener solution. We have found that both the parameters SNR and convergence ratio are strongly depending on the number of samples in the input signal, \( \alpha \), and \( \mu \) and filter order. As the number of samples in the input signal increases SNR decreases and convergence ratio increases. Figure 2, 3, 4 and 5 shows the input signal, that is corrupted signal, desired signal and the filtered signal for different input SNR’s. Table 1 shows the computational complexity of the proposed method. Table 2 shows the SNR’s of the DCT adaptive filtered outputs for different SNR of the input signals.

4. CONCLUSION

We have seen that the DCT has good ortho-normal and energy compaction property. The SNR improvement of at least 10 dB is obtained for input SNR equal to zero dB, which is higher than the other transformation techniques like DFT-LMS [17] and DWT-LMS [20]. Even in the case of above adaptive methods, the eigenvalue distribution of the input autocorrelation matrix is calculated after the transformation and power normalization. But, it is unable to give good SNR improvement and the convergence ratio is also very high. Proposed algorithm is not comparable with direct least mean square algorithm in terms of convergence ratio, where the eigenvalue ratio is in terms of thousands [6], [18].

References

FFT based DCT-LMS Speech Enhancement for Sensorineural Loss Patients


Table 1. Output SNR for different input SNR

<table>
<thead>
<tr>
<th>SNR of the input signal in dB</th>
<th>SNR of the output signal in dB</th>
<th>Eigen value ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10.0</td>
<td>6.09</td>
</tr>
<tr>
<td>+5</td>
<td>11.24</td>
<td>5.44</td>
</tr>
</tbody>
</table>

Table 2. Computational complexity for N=8.

<table>
<thead>
<tr>
<th>Transform</th>
<th>No. of real multiplications</th>
<th>No. of real additions</th>
<th>Complexity of computation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDCT</td>
<td>32</td>
<td>28</td>
<td>$O(N^2)$</td>
</tr>
<tr>
<td>IMDCT</td>
<td>32</td>
<td>24</td>
<td>$O(N^2)$</td>
</tr>
<tr>
<td>Fast DCT using FFT</td>
<td>7</td>
<td>15</td>
<td>22</td>
</tr>
</tbody>
</table>
Author’s Biography


Currently he is a Professor with the department of Instrumentation Technology, at Sri Jayachamarajendra College of Engineering, Visveswaraiah Technological University, India. His research interests include Rehabilitation Engineering, Digital Signal Processing, Speech recognition, Speech enhancement and EEG analysis. He has authored more than 50 publications in Journals, National and International Conferences in these areas.

Sunitha S.L received the B.E. degree in Instrumentation Technology from Malnad College of Engineering, Hassan University of Mysore, India. MTech. degree in Biomedical Engineering from Sri Jayachamarajendra College of Engineering, Mysore from Visveswaraiah Technological University, Karnataka. India. Currently she is a research scholar with Sri Jayachamarajendra College of Engineering, India and is teaching at SJMIT, Chitradurga. Her research interests include Digital Signal Processing, Speech processing, Neural Networks and Wavelet transforms.
Simulation Studies On State And Bias Estimation Of Continuous Stirred Tank Reactor Using Augmented State Kalman Filter

S. Abraham Lincon¹  D. Sivakumar¹  J. Prakash²

1. Department of Electronics and Instrumentation Engg, Annamalai University, Annamalai Nagar 608002.
2. Department of Instrumentation Engg, MIT Campus, Anna University, Chennai – 600025.
E.Mail: dsk2k5 @ gmail.com

ABSTRACT

In order to control a process, we first need an accurate estimate of the state variables. But often the variables of interest, the quantities needed to describe the state of the system cannot be measured directly and some means of inferring these values from the available data must be generated. Kalman estimator is a tool for obtaining the reliable state estimate. In the application of Kalman filtering technique for state estimation, an accurate model of the process dynamics and measurements is required. However in many practical cases random bias affects the system dynamics and/or observations and may lead to degradation of the performance of the filter if the bias is not incorporated in the model. A novel approach for bias and state estimation in a continuous stirred tank reactor employing the Kalman estimator is proposed in this research paper. Implementation of Kalman Estimator is done using MATLAB 5.3 software.

Keywords: Augmented state Kalman Filter  Continuous stirred tank Reactor  Sensor fault, Actuator fault, fault Detection and Diagnosis.

1. INTRODUCTION

Fault Detection and Identification for dynamic systems has gained more attention in the recent years mainly due to an increasing demand for higher performance as well as higher safety and reliability standards (Isermann, 1984; Basseville, 1988; Frank, 1990; Gertler, 1993; Patton, 2005). Faults can occur either in the main processing equipments (leak in tanks, variation in process parameters like heat transfer coefficient etc.) or in the auxiliary equipments (bias/drift in sensors, actuators, controller outputs, stuck actuators and sensors). Faults such as stuck sensors and stuck actuators are classified as hard faults. In principle such faults occur less frequently and it is comparatively easy to detect using low and high alarm thresholds. On the other hand small biases in sensors, actuators and controller outputs are considered as soft faults and are not easy to detect. This paper is focused on detection and diagnosis of soft faults only. The main task of fault detection and diagnosis can be described as early determination (detection) and localization (diagnosis) of faulty elements in a dynamic system.

In fault diagnosis its application is to detect the onset of fault and estimate the parameter drifts and/or abrupt changes in system parameters. Parameter estimation problem can be formulated as state estimation problem by considering the fault parameter of interest as additional states.

Kalman filter has been widely used for state estimation of linear stochastic systems. In order to generate accurate state estimates in the presence of faults and plant-model mismatch, the state vector has been augmented with fault parameter as an additional state and solved as simultaneous state and bias estimation problem resulting which is the Augmented state Kalman filter.
The paper is organized as follows: The fault detection and diagnosis scheme is described in section 2, the system description in section 3, simulation results are presented in section 4 and conclusion in section 5.

2. FAULT DETECTION AND DIAGNOSIS SCHEME

Consider the process which is described by the following linear time invariant state space model,

\[
X(k+1) = \Phi X(k) + \Gamma_u U(k) + \Gamma_w W(k) \quad (1)
\]

\[
Y(k) = CX(k) + V(k) \quad (2)
\]

where \( X(k) \in \mathbb{R}^n \) state vector, \( U(k) \in \mathbb{R}^m \) the known deterministic inputs and \( y(k) \in \mathbb{R}^r \) the measured outputs.

\( W(k) \) and \( V(k) \) represents state and measurement noises. It is further assumed that \( W(k) \) and \( V(k) \) are zero mean normally distributed and mutually uncorrelated white noise sequences with covariances,

\[
E \{ W(k) W^T(k) \} = Q \delta_{ij} \quad E \{ V(k) V^T(k) \} = R \delta_{ij} \quad \text{with} \quad Q \geq 0; R \geq 0
\]

where \( \delta_{ij} \) is the Kronecker delta and \( E \) denotes the expectation operator. The linear time-invariant matrices \((\Phi, \Gamma_u, \Gamma_w \text{ and } C)\) can be obtained from the first principle model of the process after performing linearization and discretization operations.

A fault \( f(k) \) due to step change in process parameter, or actuator bias can be modeled using the following state evolution equation,

\[
X(k+1) = \Phi X(k) + \Gamma_u U(k) + \Gamma_w W(k) + \Gamma_f f(k) \quad (3)
\]

where \( \Gamma_f \) represents coupling matrix and \( e_{ij} \) is a unit vector indicating the index of the variable. The matrix \( \Gamma_f \) for different faults can be obtained from first principle models.

A fault due to sensor bias(\( \Gamma_s \)) can be modeled by modifying the measurement equation (2) as,

\[
Y(k) = CX(k) + V(k) + \Gamma_s e_{ij} f(k) \quad (4)
\]

In the presence of fault the process evolves according to equations (3) and (4). Let us assume the process evolves according to an augmented state space model after combining equations (3) and (4)

\[
X(k+1) = \Phi X(k) + \Gamma_u U(k) + \Gamma_w W(k) + \Gamma_f f(k) \quad (5)
\]

\[
f(k+1) = f(k) \quad (6)
\]

\[
Y(k) = CX(k) + V(k) + \Gamma_s f(k) \quad (7)
\]

Defining an augmented state vector \( X_a = \begin{bmatrix} X(k) \\ f(k) \end{bmatrix} \)

Equations (5),(7),(8) can be rewritten as

\[
X_a(k+1) = \Phi_a X_a(k) + \Gamma_{ua} U(k) + \Gamma_{wa} W(k) \quad (8)
\]

\[
Y(k) = C_a X_a(k) + V(k) \quad (9)
\]

Where

\[
\Phi_a = \begin{bmatrix} \Phi & \Gamma_f \\ 0 & I \end{bmatrix}, \quad \Gamma_{ua} = \begin{bmatrix} \Gamma_u \\ 0 \end{bmatrix}, \quad C_a = [C \ \Gamma_s]
\]

The initial state \( X(0) \) and \( f(0) \) are Gaussian random variables with

\[
E \{ X(0) \} = \bar{X}(0), E\{f(0)\} = \bar{f}(0), \quad E \{ (X(0) - \bar{X}(0)) (X(0) - \bar{X}(0))^T \} = P_x(0)
\]

\[
E \{ (f(0) - \bar{f}(0)) (f(0) - \bar{f}(0))^T \} = P_f(0) > 0
\]

The specific values of \( \Gamma_f \) and \( \Gamma_s \) in the augmented state space model for each type of fault under consideration are as follows:

Sensor bias: \( \Gamma_f = 0 \) and \( \Gamma_s = I \)

Actuator bias: \( \Gamma_f = \Gamma_u e_{ui} \) and \( \Gamma_s = 0 \)

The maximum likelihood estimates of the augmented states can be generated using Augmented State Kalman Filter.
Augmented State Kalman Filter

\[
\hat{X}_a(k+1/k) = \Phi_a \hat{X}_a(k/k) + \Gamma_a U(k) \\
(10)
\]

\[
\hat{Y}(k+1/k) = C_a \hat{X}_a(k+1/k) \\
(11)
\]

\[
P_a(k+1/k) = \Phi_a P_a(k/k) \Phi_a^T + \Gamma_a \Sigma_a \Gamma_a^T \\
(12)
\]

\[
K_a(k+1) = (\Phi_a P_a(k/k) C_a^T + R)^{-1} \hat{Z} \\
(13)
\]

\[
\hat{X}_a(k+1/k) = \hat{X}_a(k+1/k) + K_a(k+1) (\hat{Y}(k+1/k) - \hat{Y}(k+1/k)) \\
(14)
\]

\[
P_a(k+1) = (I - K_a(k+1) C_a) P_a(k+1/k) \\
(15)
\]

where \( P_a(0/0) = Q_a \)

An Irreversible series parallel reaction known as Van de Vusse Reaction scheme is of the following form,

\[
A^* \xrightarrow{k_1} B^* \xrightarrow{k_2} C^* \\
2A^* \xrightarrow{k_3} D^* \\
\]

where, \( A^* \) is Cyclopentadiene, \( B^* \) is Cyclopentenol, \( C^* \) is Cyclopentanediol, \( D^* \) is Dicyclopentadiene and \( k_1, k_2, k_3 \) are Reaction rate constants. This scheme was presented by Van de Vusse (1964). Engell and Klatt (1998) it is absorbed that the production of Cyclopentenol from Cyclopentadiene is based on this reaction scheme.

### Process Modeling

The Component Balance Equations for the components \( A^*, B^*, C^*, D^* \) are as follows:

\[
\frac{dC_A}{dt} = \frac{F}{V} (C_B - C_A) - k_1 C_A - k_2 C_A^2 \\
\]

\[
\frac{dC_B}{dt} = \frac{F}{V} C_A + k_1 C_B - k_2 C_B^2 \\
\]

\[
\frac{dC_C}{dt} = \frac{F}{V} C_A + \left( \frac{C_B}{C_A} C_B \right) - k_3 C_B \\
\]

\[
\frac{dC_D}{dt} = \frac{F}{V} C_A + \left( \frac{C_B}{C_A} C_B \right) - k_3 C_D \\
\]

where, \( C_A \) is the Concentration of \( A^* \), \( C_B \) is the Concentration of \( B^* \), \( C_C \) is the Concentration of \( C^* \), \( C_D \) is the Concentration of \( D^* \), \( V \) is the Volume of CSTR and \( F/V \) is the Space velocity. The desired product is the component \( B^* \), the intermediate component in the series reaction. Notice that the first two equations (16) and (17) does not depend on the concentration of components \( C^* \) or \( D^* \). From the Open loop response of the CSTR process
Simulation Studies On State And Bias Estimation Of Continuous Stirred Tank Reactor Using Augmented State Kalman Filter

the steady state operating data of the process are obtained and are shown in the Table.1. From the open loop response the time constant is found to be 0.1985 seconds. 1/10 of time constant is 0.01. The discretized model is determined with sampling time $t_s$ set at 0.01 sec is as follows

\[
\Phi = \begin{bmatrix}
0.94425 & 0 \\
0.0079156 & 0.9556
\end{bmatrix},
\Gamma = \begin{bmatrix}
0.038029 \\
-0.010763
\end{bmatrix}, \quad C = \begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix}
\]

Table.1 Steady state Operating Data of CSTR

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{AS}$</td>
<td>6.087 mol/lit</td>
</tr>
<tr>
<td>$C_{BS}$</td>
<td>1.117 mol/lit</td>
</tr>
<tr>
<td>$C_{AFS}$</td>
<td>10 mol/lit</td>
</tr>
<tr>
<td>$F_s/V$</td>
<td>2.8744 min$^{-1}$</td>
</tr>
<tr>
<td>$k_1$</td>
<td>0.8333</td>
</tr>
<tr>
<td>$k_2$</td>
<td>1.6667</td>
</tr>
<tr>
<td>$k_3$</td>
<td>0.1666</td>
</tr>
</tbody>
</table>

For the Augmented Single stage Kalman Estimator,

\[
Q = \begin{bmatrix}
6e^{-10} & 0 & 0 \\
0 & 1.1e^{-10} & 0 \\
0 & 0 & 1e^{-8}
\end{bmatrix},
R = \begin{bmatrix}
6e^{-10} & 0 \\
0 & 1.1e^{-10}
\end{bmatrix}
\]

\[
P(0/0) = Q_0 = \begin{bmatrix}
0 & 0 \\
0 & 0
\end{bmatrix}, \quad \hat{X}(0) = \begin{bmatrix}
0 \\
0
\end{bmatrix}
\]

4. RESULTS AND DISCUSSION

Simulation runs have been performed with abrupt fault in actuator and sensor of magnitudes 1.4372e-2 and 1.5217e-2 respectively.

Figure 2 shows the True and Estimated Concentrations (states) of CSTR process for Augmented State Kalman Estimator in the presence of abrupt fault (bias) of magnitude 1.4372e-2 introduced at 100th sampling instant. The Estimate of abrupt fault (bias) in the actuator of magnitude 1.4372e-2 introduced at 100th sampling instant is shown in Figure 3.

In Figure 4 the Estimate of abrupt fault (bias) in the sensor of magnitude 1.5217e-2 introduced at 100th sampling instant is shown.
Figure. 4 Estimate of abrupt fault (bias) in the sensor of magnitude 1.5217e-2 introduced at 100th sampling instant.

5. Conclusion

For stable estimation of linear stochastic systems, Kalman Filter. The fault in sensor is considered as an additional state and accordingly the system matrices are augmented which results in an Augmented State Kalman Filter. The Performance of Augmented State Kalman Filter for simultaneous State and Fault parameter estimation have been studied on the model of CSTR for abrupt fault in actuator and sensor. From this simulation studies it is evident that the State and Fault parameters are estimated accurately with minimum variance by the Kalman Estimator. Further it is observed that the probability occurrence of false and missed alarm are very less. The proposed FDD scheme can be integrated with any control scheme to develop fault tolerant control scheme.

References


Author’s Biography

S. Abraham Lincon obtained his B.E degree in Electronics and Instrumentation Engineering in the year 1984 and received M.E. degree in Power System Engineering in 1987 and another M.E degree in Process Control and Instrumentation Engineering in 2000 from the Annamalai university, Chidambaram. Presently he is working as a Reader in the Department of Instrumentation Engineering in Annamalai University. His areas of research are Process Control, Fault Detection and Diagnosis and Multivariable Control.

Dr. D. Sivakumar obtained his B.E degree in Electronics and Instrumentation Engineering in the year 1984 and further the M.E. degree in Power Systems in the year 1990 from the Annamalai university, Chidambaram. Presently he is working as a Professor in the Department of Instrumentation Engineering in Annamalai University. He is presently guiding many Ph.D research scholars in areas like Fault Detection and Diagnosis, Neural Networks and Fuzzy Logic applied to Process Control, Signal and Image Processing.

Dr. J. Prakash obtained his B.E degree in Electronics and Instrumentation Engineering in the year 1993 from Annamalai university, Chidambaram and he got his M.E degree in Control System Engineering in the year 1995 from PSG college of Technology, Coimbatore. Subsequently he received his Ph.D degree in Chemical Process Control Engineering from the prestigious institute IIT Madras, Chennai. Presently he is working as an Assistant Professor in the Department of Instrumentation Engineering, Anna University, MIT Campus, Chennai. Currently he is guiding many Ph.D and M.S scholars in the areas like Instrumentation, Process Control, Fault Talent Control, Fault Detection Diagnosis, Multivariable Control, Digital Signal Processing.
PRISM : A software tool for the prediction of secondary structures in proteins

K. Saravanan¹  T. Sivaraman²

ABSTRACT

The three-dimensional (3D) structures of proteins are being determined by using NMR or/and X-ray crystallography techniques in conjunction with computational methods at atomic level resolutions. These high resolutions structures of proteins are essential for understanding their structure-function relationships. However, there is a divergent correlation between these experimental outcomes and requirements of current research in structural biology. In this background, PRISM a software tool was created to predict percentage of secondary structures in proteins based on molecular mass of their deuterated-forms. By taking into consideration of many different structural and environmental factors, the PRISM also validates its outputs and gives suggestion to improve experimental conditions for better predictions. The applications of PRISM for the data analysis in proteomics and genomics have also been brought into fore in detail.

Keywords: Proteins, Secondary structures, Exchange, Mass spectrometry, PRISM algorithm.

1. INTRODUCTION

Proteins, one of the major governing bodies of bio-systems, play significant roles on structural and functional aspects of all living systems [11]. In general, proteins are made-up of a-L-amino acids that are linked by peptide bonds. Each protein has unique secondary structures, which regulate its function(s). Understanding the features of these secondary structures is essential for denovo protein/peptide designing [6,11,15]. At present, the secondary structures of proteins can be elucidated at atomic level resolution by using techniques such as NMR Spectroscopy and X-ray crystallography [14]. However, requirement of extensive analysis on the data obtained from these techniques consumes a long period of time [5]. Moreover, the benefit of these techniques may not fulfill timely the need of projects like Human Genome Project in which structures are expected for thousands of proteins. In order to help and speed-up the processes of structural analysis on proteins, we have developed a novel program, PRISM. The structure and applications of the PRISM has been discussed in this article.

2. PREREQUISITES FOR PRISM ANALYSIS

The molecular mass (MM) of deuterated conformations of proteins under interest should be estimated by mass spectrometry (MS) and the MM should be given as one of two inputs to PRISM. Usually, proteins can be deuterated by Hydrogen-Deuterium (H/D) exchange process [9]. H/D exchange is a process in which labile protons in proteins are substituted by solvent deuterons. By dissolving a test protein in deuterium oxide (D₂O) for a limited time at defined pH, temperature and ionic strength, a deuterated protein can be prepared [2,9]. The H/D exchange process is represented as shown below.
PRISM : A software tool for the prediction of secondary structures in proteins

In this model, exchange competent (Open NH) and exchange incompetent (Closed NH) conformations interconvert with rate constants $k_{op}$ and $k_{cp}$, respectively. Exchange happens only from the open conformation with a rate constant of $k_{ch}$. The observed rate constant of exchange is described as,

$$k_{obs} = \frac{(k_{op} + k_{ch})}{(k_{op} + k_{cl})}$$

Under conditions, where $k_{ch}$ is greater than $k_{cl}$, the observed rate constant of exchange $k_{obs}$ becomes $k_{op}$. This is known as EX1 exchange [1]. However, if $k_{cl}$ is greater than $k_{ch}$, then,

$$k_{obs} = \frac{(k_{op} * k_{ch})}{k_{cl}} = K_{eq} * k_{ch}$$

Where $K_{eq}$ is the equilibrium constant. This is known as EX2 exchange [13]. The deuterated proteins should be prepared by EX2 exchange mechanism for determining its secondary structural content by PRISM. This follows from the fact that EX2 exchange probes the equilibrium of a protein between its folded and unfolded states [4,9,13].

The MM of a deuterated protein can be calculated using MS [1]. The advantage of this method is many folds: i) the MS requires protein sample only in microgram level ii) MM can be calculated at higher resolution of 0.001 Dalton (Da) iii) the time required to collect the data is in the time span of minutes (<5 minutes) iv) experiments are simple and inexpensive. The MM of deuterated protein as determined by MS and amino acid sequence of the protein are only the two inputs to PRISM. The combined MS and PRISM analyses will reveal the percentage of secondary structural content of given test protein in fifteen minutes or so. The implications of the outcomes have been discussed in the later section of this article.

The ‘C’, POP language, is used to write the PRISM. The language facilitates the use of selection and repetition controls as the PRISM requires them at enormous level. In near future, the PRISM will be allowed to be freely accessible for all through the website of PRIST (Ponnaiyah Ramajayam Institute of Science and Technology). The website of PRIST is www.prist.ac.in.

3. THE PRISM ALGORITHM

The PRISM algorithm has been developed to predict percentage of secondary structures in proteins based on MM of their deuterated conformations as determined by MS. In outline, the program first counts number of amino acids in the input protein sequence and then their molecular masses are summed. From this value, the MM of the input protein sequence is determined from the following equation.

$$\text{MMIS} = \sum_{i=1}^{n} \text{MMA} - [(n-1) \times \text{MMH}_{2}\text{O}]$$

Where, MMIS – Molecular Mass of input sequence
n – Number of amino acid
MMA – Molecular Mass of amino acid
MMH$_2$O – Molecular Mass of H$_2$O

The respective atomic and molecular mass corresponding to various atoms and molecules used in PRISM were referred using the OPUSTM and IsoPro software [1,13]. The deuterated form of the protein is calculated assuming all labile protons (H) of the protein are exchanged to solvent deuterons (D).

Lastly, the percentage of secondary structures of the input protein is predicted using the following relationship.
PSS = [(CDMM – MSMM) / NBLP] * 100
Where, PSS = Percentage of secondary structures
CDMM = Molecular mass of completely deuterated
form of input sequence
MSMM = Molecular mass as determined by MS
NBLP = Total number of backbone labile protons.

The algorithm will be illustrated in more detail by a
schematic diagram for a hypothetical protein molecule
having eight amino acids (Fig. 1). The primary sequence
is given in Fig. 1A (P - Proline, R - Arginine, I - Isoleucine,
S - Serine, M - Methionine and C - Cysteine). Molecular
formula and MM of the sequence are shown in Fig. 1B
based on the number and types of amino acids that are
building the sequence. Fig. 1C depicts total number of
labile protons in each amino acid of the sequence. It is
important to mention that all α-L-amino acids have only
one backbone labile proton except proline, which has no
backbone labile proton [6, 11]. The MM of completely
deuterated form of given sequence has also been shown
in Fig. 1C. Fig. 1D reveals the percentage of secondary
structures (PSS) as predicted for the sequence and its
reliability.

4. Special Features of PRISM
The PRISM requires two inputs: i) the primary sequence
of a protein for which secondary structures need to be
predicted ii) MM of the deuterated protein as determined
by MS. The PRISM delivers seven outputs for each
complete run: i) Molecular formula ii) Molecular mass
iii) Total number of labile protons (TLP) iv) Side chain
labile protons (SLP) v) Backbone labile protons (BLP)
vi) MM of completely deuterated protein vii) Percentage
of secondary structures (PSS). The outputs from the
PRISM for a hypothetical protein (PRISMPRCCSKSTSNEWPRISMAA) are depicted in
Table 1.

The unique feature of PRISM is that it gives seven outputs
for oxidized as well as reduced conformations of proteins
input. In reduced state, proteins are bereft of disulfide
bonds whereas in oxidized state, cysteine residues of
proteins become cystine. The PRISM accounts both
forms of proteins and this particular aspect pave a way to
derive structural parameters pertaining to disulfide bonds
in proteins provided their secondary structures are known.
Moreover, information about various types of labile
protons in proteins is useful in designing kinetics of H/D
exchange in proteins.

In some cases, PSS as predicated by PRISM for given
input might be misdirected. For instance, if the time
required for H/D exchange process is not optimized, the
protein molecules will be partly deuterated wherein some
side chain labile protons may not be completely
exchanged. In this circumstance, the PRISM
overestimates the PSS and gives warning message as “PSS
is not reliable, because some SLPs are protected from
exchange”. In case where PSS ≥ 80%, the warning
message is that “A few number of SLPs may be protected
from exchange”. This is based on the observations that
proteins may assume maximum about 60% secondary
structures in their 3D architectures [6, 11, 14]. Apart from
the above possible conditions, the number of amino acids
and BLPs accounting the PSS is also determined by
PRISM. If the number were less than 4, independent of
PSS, the warning message would be “The PSS is not
reliable. More experimental evidences are required to
support/improve the prediction”. This is based on the
fact that at least four amino acids are essential for
constituting defined secondary structures such as α-helix,
3ₑ-helix, parallel β-sheets and anti parallel β-sheets in
proteins [14]. Only under conditions where PSS < 80%
and NBLP accounting secondary structures is ≥ 4, the
PRISM shows ‘NIL’ in its warning message. Hence, the
warning messages are obviously helpful to revisit/refine the experimental conditions. To our best knowledge, though there are many algorithms [3,7,10,12] for prediction of secondary structures in proteins, PRISM is the first software tool for predicting secondary structural contents of proteins based on their deuterated forms.

5. CONCLUSION
We have clearly demonstrated that PRISM predicts intact secondary structures of proteins and rationalize its results by taking into consideration of factors from many facets. In the near future, the PRISM will be improved for defining various types of secondary structures in proteins along with PSS. This particular goal can be achieved by coupling a program predicting types of secondary structures in proteins with PRISM. Foreseeing the potential applications of PRISM in structural biology, we do anticipate a great scope to improve the software tool at many different angles.

REFERENCES
Figure 1: The schematic diagram outlines the major steps involved in PRISM. Fig.1D depicts outcomes of PRISM for oxidized form of given polypeptide sequence in Fig.1A. The molecular mass (MM) of 972.3325 was taken as if the MM determined by MS for the input sequence. The Fig.1B and 1C illustrate how the PRISM counts the number of amino acids and labile protons of the given sequence, respectively. As the PRISM finds that NBLP accounting the PSS is less than 4, the warning message alerts “PSS is not reliable. More experimental evidences are required to support/improve the prediction” (Refer text for details).

Table 1: Inputs and Outputs of PRISM for a hypothetical protein.

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Inputs</th>
<th>Outputs</th>
<th>Conformations of proteins</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Protein Sequence: PRISMPRCCSKSTSNWPRISMAA</td>
<td>Oxidized form</td>
<td>Reduced form</td>
</tr>
<tr>
<td>1.</td>
<td>MM by MS: 2746.5706 Da</td>
<td>C_{106}H_{180}O_{35}N_{36}S_{4}</td>
<td>C_{106}H_{182}O_{35}N_{36}S_{4}</td>
</tr>
<tr>
<td>2.</td>
<td>Molecular Mass (MM)</td>
<td>2707.1785 Da</td>
<td>2709.1946 Da</td>
</tr>
<tr>
<td>3.</td>
<td>MM of completely deuterated protein</td>
<td>2752.5385 Da</td>
<td>2756.5706 Da</td>
</tr>
<tr>
<td>4.</td>
<td>Number of BLPs</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>5.</td>
<td>Number of SLPs</td>
<td>24</td>
<td>26</td>
</tr>
<tr>
<td>6.</td>
<td>Number of TLPs</td>
<td>45</td>
<td>47</td>
</tr>
<tr>
<td>7.</td>
<td>PSS in the given protein</td>
<td>28.19</td>
<td>47.24</td>
</tr>
</tbody>
</table>

Warning: NIL
Figure 1A.

Input sequence: PRISMPRC

MM by MS: 972.3325 Da

Figure 1B

Molecular formula: C_39H_70O_{10}N_{14}S_{2}

MM: 959.2045 Da

Figure 1C

MM of completely deuterated protein: 975.3325 Da

Figure 1D

Percentage of Secondary Structures (PSS): 49.60

Warning: PSS is not reliable. More experimental evidences are required to support/improve the prediction.
Author’s Biography

Dr. T.Sivaraman received his M.Sc. in Chemistry from Bharathidasan University, Trichy in 1994 and his Ph.D. in Biochemistry from National Tsing Hua University, Taiwan in 1999. After post-doctoral career at University of Iowa, USA and National University of Singapore, Singapore, he joined the faculty in the department of Biochemistry at Ponnaiyah Ramajayam College, Thanjavur in 2003. His research interest is focused on understanding the relationships between structures and functions of proteins using experimental techniques and computational methods.

Mr. K.Saravanan received his M.Sc. in Computer Science from A.V.C.College (Autonomous), Mayiladuthurai in 1992, M.S. in Software Systems from B.I.T.S., Pilani in 1998 and M.Phil. in Computer Science from M.S. University, Tirunelveli in 2003. Right now, he is doing his Ph.D. in Computer Science under the guidance of Dr.T.Sivaraman. He has been working as HOD of computer science, Ponnaiyah Ramajayam College, Thanjavur from 1994.