International Journal of Computer Science and Application
IJCSA Volume 1, Issue 1, August 2012 PP. 1-5
ISSN(online)
ISSN(print)
www.ij-csa.org

Features of Adaptive Test Suites

Narendra Kumar Rao. B[1], RamaMohan Reddy. A[2]
[1]Sree Vidyanikethan Engineering College,Tirupati, [2]Sri Venkateswara University, Tirupati.
Email:narendrakumarrao@yahoo.com

(Abstract) Case-based reasoning is an approach to problem solving and learning that has got a lot of attention over the last few years. This paper provides an overview of the case-based reasoning, brief outline on applicability of case-based reasoning to test suite, describing the various features of adaptive test suite system based on CBR framework will possess, focuses on benefits of adaptive approach over existing test case selection/retrieval approaches. The current work demonstrates the benefit of case base in reasoning approach for selecting test case or for testing the product during system testing and regression testing phases of the product.

Keywords: Retrieve; Reuse; Revise; Retain; Test Suite Reduction; Test cCase Prioritization; Test Case Selection; Test suite Optimization.

INTRODUCTION

In Case Based Reasoning, problems are solved by adapting the solutions of similar previous cases stored in a case memory. CBR mainly consist of four phases they are Retrieve, Reuse, Revise, and Retain. [1] Retrieving the cases from the case base whose problem is most similar to the new problem. Reusing is the solutions from the retrieved cases to create a proposed solution for the new problem. Proposed solutions are revised to take into account of the problem differences between the new problem and the problems in the retrieved case. Retaining the new problem and its revised solution as new case for new case for case-base if appropriate. CBR is having assumptions, first and main assumption is that "Similar problems have similar solutions". And other assumptions are "The world is a regular place"- what holds true today will probably hold true tomorrow. Next is "Situation repeat"- if they do not there is no point in remembering them.

Test suite is a collection of test cases that intended to be used to test a software program to show that it has some specific set of behaviors. A Test case is representation of test scenarios. A test suite comprises of detailed instructions or goals for each collection of test cases and information on the system configuration to be used during testing. The main contribution of the work is Classification of techniques for R4 phases in CBR, Test suite optimization, Case Based reasoning and Test suite optimization.

Markovitch & Scott proposed a unifying framework for the systematic discussion of various strategies for coping with harmful knowledge in general, and the case of utility problem in particular. The framework is based on various types of filters for eliminating harmful knowledge at various stages in the problem solving procedure. An approach that is especially relevant in CBR is to simply delete harmful cases from the case base so that they cannot actively contribute to ongoing problem solving costs deletion policies in CBR correspond to selective retention filters in the Markovitch & Scott framework. Surprisingly, in many speed-up learners the apparently naive random deletion of knowledge items (to maintain the knowledge base to some predefined size) works quite well for optimizing efficiency. Even though random deletion eliminates both useful and redundant items it can equal the success of more sophisticated methods.

Overview of Case-Based Reasoning:

The processes involved in CBR can be represented by (Fig-1). Aamodt and Plaza described a CBR typically as a cyclical process comprising *the four steps*:

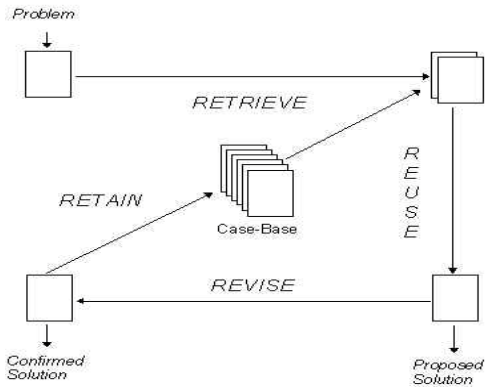

Fig-1. The CBR Cycle

- RETRIEVE the most similar case(s);
- REUSE the case(s) to attempt to solve the Problem
- REVISE the proposed solution if necessary, and
- RETAIN the new solution as a part of a new case.

A new problem is matched against cases in the case base and one or more similar cases are retrieved. A solution suggested by the matching cases is then reused and tested for success. Unless the retrieved case is a close match the solution will probably have to be revised producing a new case that will be retained.

Few Important terminologies related to CBR are as follows:
Case Representation:

A case is a contextualized piece of knowledge representing concerned experience or knowledge. It stores the past lesson that is the content of the case and the context in which the lesson is to be used. Typically a case comprises of following:

- problem describes the state of the world when a scenario case occurs,
- solution which selects the derived solution to a problem, and/or
- outcome describes the state of the world after the case occurrence.
- Indexing

Case indexing involves assigning indices to cases for alleviating case retrieval. Indices should follow few properties:

- predictive,
- address the purposes of case usage,
- allows for widening the future use of the case-base, and
- be sufficient enough for usage in future

Both manual and automated methods are used in index selection. Selection of indices manually involves deciding a case's purpose with respect to the aims of the reasoner and deciding under what circumstances the whether the case will be used.

Despite the success of many automated methods, it is believed that people tend to do better at choosing indices than algorithms.

Induction

Induction algorithms (e.g. ID3) determine which features does the best job in differentiating cases, and generate a decision tree type structure for organizing the cases in memory. This approach is better when a single case feature is required as a solution, and where that particular case feature is depends on others.

Adaptation

Once a matching case is retrieved a CBR system should adapt the solution stored in the retrieved case to the current case needs. Adaptation identifies prominent differences between the retrieved case and the current case and then applies formulae or rules that are taken into account when suggesting a solution. In general, there are two kinds of adaptation in CBR:

- *Structural adaptation*, in which adaptation rules are to be applied directly to the solution stored in cases. This kind of adaptation is followed with JUDGE and CHEF.
- *Derivational adaptation*, that reuses the algorithms, methods or rules that generated the original solution to produce a new solution to the current problem. In this method the planning sequence that constructed that original solution must be stored in memory along with the solution as in MEDIATOR.

An efficient CBR system requires both structural adaptation rules to adapt poorly understood solutions and derivational mechanisms for adapting solutions of cases that are clearly understood.

Few case based reasoning adaptation techniques are as follows:

- Null adaptation
- Parameter adjustment
- Abstraction and respecialisation
- Critic-based adaptation
- Re-instantiation
- Derivational replay
- Model-guided repair
- Case-based substitution

TEST SUITE:

In software development, a **test suite**, less commonly known as a *validation suite*, is a collection of test cases that are intended to be used to test a software program to show that it has some specified set of behaviors. A test suite often contains detailed instructions or goals for each collection of test cases and information on the system configuration to be used during testing. A group of test cases may also contain prerequisite states or steps.[8]

Test suite reduction problem:

The first formal definition of test suite reduction problem introduced in 1993 by Harrold et al. as follows: Given. {*t1*, *t2*,…, *tm*} is test suite *T* from *m* test cases and {*r1*, *r2*,…, *rn*} is set of test requirements that must be satisfied in order to provide desirable coverage of the program entities and each subsets {*T1*, *T2*,…, *Tn*} from *T* are related to one of *ri*s such that each test case *tj* belonging to *Ti* satisfies *ri*.

Problem. Find minimal test suite *T'* from *T* which satisfies all *ri*s covered by original suite *T*.

Generally the problem of finding the minimal subset *T'*, *T'* is subset of *T* which satisfies all requirements of *T*, is NP-complete, because we can reduce the *minimum set-cover* problem to the problem of test suite minimization in polynomial time. Thus, researches use heuristic approaches to solve this problem.

Related work in the context of test suite reduction can be classified into two main categories: The works in which a new technique is presented and empirical studies on the previous techniques. The works proposed earlier focused includes heuristic algorithms, genetic algorithm-based techniques and approaches based on integer linear programming. In a recent study, four typical test suite reduction techniques have been evaluated and compared on 11 subject programs. Based on the results, this study suggests that the heuristic may be the first choice when selecting from the test case reduction techniques.

Testing criteria are defined to help the selection of subsets of the input domain to be covered during testing. For example, a code coverage criterion provides test suite adequacy with respect to coverage of the program entities and also provides a check on its quality. Assuming testing criterion *C* which satisfies by the test suite *T*, a test case, *t*, is *redundant* if the suite *T*-{*t*} also satisfies *C*. Therefore, removing those test cases which are redundant with respect to some specific criteria preserves test suite's adequacy with respect to that criteria. In previous empirical studies researchers commonly applied various code coverage criteria in their reduction techniques. The results of empirical studies demonstrate that the percentage of the test suite size , reduced the percentage of faults. Usually a test suite reduction technique attempts at removing *redundancy* among test cases and retain the most *effective* test cases into the test suite. Effective test cases are those that satisfy most of the requirements as well as exposing the most of the existing faults. Note that the more requirements are satisfied, the more execution paths within the program would be exercised which yields the all kinds of the faults to be exposed.

TEST CASE CLASSIFICATION:

System test cases cover the application from end to end. It is not based on the requirements given by the client. We need to cover the whole application. Test cases were classified into different types:

- Graphical user interface testing
- Usability testing
- Performance testing
- Compatibility testing
- Error handling testing
- Load testing
- Volume testing
- Stress testing
- Security testing
- Scalability testing
- Sanity testing
- Smoke testing
- Exploratory testing
- Ad hoc testing
- Regression testing
- Reliability testing
- Installation testing
- Maintenance testing
- Recovery testing and failover testing.

i. Applicability of case-based reasoning to test suite:

The Case Based Reasoning as such can be tailored to meet the necessary condition for test suite reduction. The four Phases involved can be designed as below:

Retrieve: The retrieval case can only proceed once there is a proper structure storing the desired case in required format suitable for retrieval. In this direction a tree based approach is designed for the case. A partial DC tree is given in Fig-2, a Decision and Classification tree is constructed. A DC tree comprises of system test cases arranged based system artifacts of the following (typical but not restricted to):

- Type of system testing
- Modules of the system
- Dependency between modules
- Interfaces involved
- Classes involved
- Dependency between classes, etc..

In Case based reasoning the cases are constructed based on the above defined artifacts as attributes. The system test cases obtained from requirements are arranged as the separate case in low order of the tree. In fact a tree is constructed based on the above attribute and values. A system test case can fall under a particular system testing, module of a system, dependency between modules, interface, classes involved, dependency among classes and so forth which can be sub case for the major case.

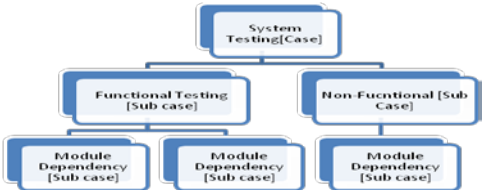

Fig-2: Partial DC Tree

Specifically the test cases can be decided by the tester of the system test cases when he is performing it manually. Anyways this CBR approach will help in identifying the test cases based on the specified details like the following. A particular objective of system testing may be selected by the tester of system test cases and this will list out all the system test cases required for fulfilling the particular objective. On further clarity/querying in information provided by the tester like the modules of the system to be tested, Dependency between modules, interface, classes involved, dependency among classes and so forth the retrieve can result in identifying a proper test case of choice which the tester may be looking for. In detail, further approach can be extended to index system test cases based on a system test case Identifier, which comprise of all the attributes forming the case. In this case the identification can be very easy for classification. Also any new case can be added further. Fig-3 signifies the same.

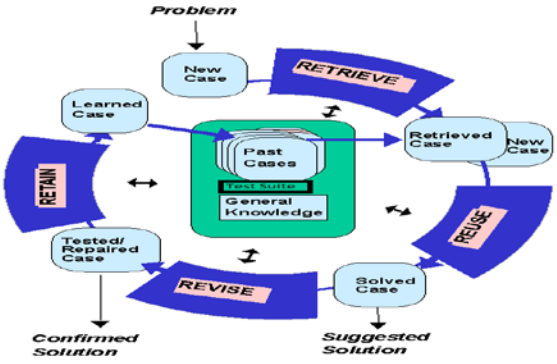

Fig-3: phases of the Test Suite Optimization based on Case Based Reasoning

Reuse The retrieved set of test case forms a basis for the current scenario can also be taken up as a new case or sub case as defined previously. Any set of system test cases that are previously tested can also be used during regression testing. The result of system test can be any one of the following passed, failed, not yet tested. The system test cases which were previously distinguished as a test case can be reused again, thereby reusing test cases.

Revise: The revision involves the testing of system test cases manually or automatic testing and thereby changing the status to one of the following states ie. Tested or not tested again tested can have two more sub states like tested but passed, tested but failed. The test cases can be revised among them so that not only test case states can be changed but tne cases can be added, duplicate system test cases can be eliminated, new cases can be added and so on.

Retain: The test cases which were revised earlier can now form a part of newly created case/sub case or a old case/sub case being created. This can further be updated in the DC tree mentioned in earlier stage of DC tree.

ii. Features of adaptive test suite system based on CBR framework*:*

The Intended adaptive test suite system acts as a repository where System test cases are arranged in form of DC tree. The designed system along with DC tree is accessible to Tester(s).Tester who requires to test a system under development has to specify details regarding the following:

- Type of system testing
- Modules of the system
- Dependency between modules
- Interfaces involved
- Classes involved
- Dependency between classes, etc

Based on more and more details a tester specifies out of the above (because test cases are arranged in above format only in the DC tree).The system is capable of doing the following:

Test case Selection and Reduction

The adaptive system can guide a tester in selection of test cases, when he is specifying the type of testing type of system testing, modules of the system, dependency between modules, interfaces involved, classes involved, dependency between classes, etc..The selected test cases are reduced by complexity of number of branches of tree at any given level in DC tree.

Store the case history of Tester(s).

- A Tester arriving at the untrained system can select all the available test cases that the CBR system suggests, later on the tester starts specifying the type of testing type of system testing, modules of the system, dependency between modules, interfaces involved, classes involved, dependency between classes, etc..
- More details the tester specifies he narrows down on the number of test cases in hand for the tester for selection.
- The entire history of the test case selection is stored in case base under a specified case name specified by Tester (name being a non-significant identifier).
- On further testing scenarios which are not relevant to existing cases in case base, new cases are stored in case base.
- If a relevant scenario of testing happens to be repeated by any current tester, which is already in case base, the adaptive system comes into being suggesting few other matching cases by previous testers. Now the current tester can opt to select a new set of system test cases or a previously used set of case, which is suggested by the adaptive system.

Most Frequently Tested cases/Scenarios/Modules/bugs

Based on number of times a test case a particular stored test case in case base is being accessed, frequency can be assigned to test cases as such the help in understanding the importance of test case for testing of module/interface. Hence frequency of execution of case(of case-base)/test case/Scenario/module/bug verification is notified based on frequency count being assigned to test case ,whenever it is accessed/deemed more frequently by the tester.

Test case Prioritization

Test case prioritization techniques schedule test cases for execution in an order that attempts to increase their effectiveness to meeting some performance goal. Such a case-wide sequence of test case can be notified from weight age methods, sequencing method etc..which are few of prioritization techniques that are supported. Also they can represent severity of defects/test cases. Such test cases can really help during sanity/release testing of modules to the client. Lesser critical tester (resource) involvement is required even critical release of product to client, which may reduce project costs. Any of familiar priority based technique may be used for this case.

Test case based Metric Evaluation

Following are few test case metrics that can be evaluated using the above specified module in adaptive test case Conceptual System:

- Functional or Test Coverage Metrics.
- Software Release Metrics.
- Software Maturity Metrics.
- Reliability Metrics.

These metrics are helpful in deciding the type of testing required, type of bugs found during test, severity of bugs, stability of module etc.

Defect Prediction

The test cases and the corresponding results help in identifying the defects in classes, modules, interfaces, system testing etc..A Naïve bayes or equivalent approach may suggest or help in concluding the defect prediction.

Regression testing scenarios

The basic feature of adaptivity to test cases is effective during regression testing, because they help in identifying the particular type of testing/test case required to test the module/class etc..A tester can get significant help during regression testing while it suggests about results of test cases previously tested, any modular changes affecting the test case results may easily be notified to the tester, which is significant

help to tester and improves productivity. A detailed discussion can correlate to a separate project on itself.

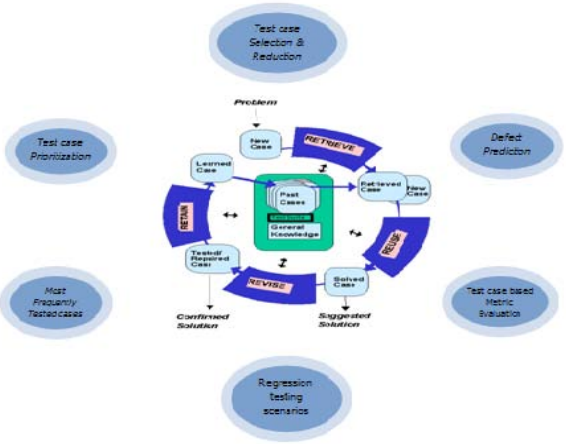

Fig-4 Conceptual System for Adaptive Test suite with few features

Above specified features form a Conceptual System for adaptive test suite optimization.

iii. Benefit of case base in reasoning approach for selecting test cases

Case based reasoning approach is based on the fact on how human beings think. Case base in adaptive test suite has the following features:

- Storage of test cases used by tester.
- Storage of testers experience in form of case in case base.
- Storage of test case results for the already tested set of test cases for a module.
- Deciding the mechanism for test case selection.
- Assigning priority of test cases.
- Deciding the modules to be tested.
- Making Defect predictions in projects.

CONCLUSION AND FUTURE WORK:

In this paper, we described about the Case Base Reasoning and the phases involved in it, features that can be used with adaptive test suite optimization, benefits of test case base etc.. Further with respect to any decision tree based System test case selection algorithm can be used. The same adaptive technique can further be used to suggest the type of test case selection algorithm to be used for a particular type of system test case selection. A detailed discussion on this can further formulate a approach of case based reasoning applied to choosing a test case selection, prioritization method selection for a particular domain of product under development viz..embedded, Web based, SAP based, Cloud based etc.. Each feature specified in attachment with adaptive test suites can further promulgate to individual research items in themselves for various domains of Projects under development. Regression is vast feature out of the above that can be exploited to maximum extent.

A Work of this nature is essential in wake of tools present for generation, execution of tools already available, but this work focuses on transfer of human knowledge, expertise, testing based on results of previous version of test cases executed, effective resource utilization(tester) in wake of reduced and effective execution of reduced set of test cases. This will further reduce testing costs to a greater extent in combination with other approaches already suggested by many other researchers.

REFERENCES

[1] RAMON LOPEZ DE MANTARAS ”Retrieval, reuse, revision and retention in case-based reasoning” The Knowledge Engineering Review, Vol. 20:3, 215–240,2006.

[2] Selma Limam Mansar “Case-Based Reasoning as a Technique for Knowledge Management in Business Process Redesign”, Academic Conferences Limited, 2003.

[3] Pádraig Cunningham “Case-Based Reasoning in Scheduling: Reusing Solution Components” International Journal for Production Research,1997.

[4] LORIA “Application of the Revision Theory to Adaptation in Case-Based Reasoning: the Conservative Adaptation”,ICCBR,2007.

[5] A. Aamodt, E. Plaza “Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches” AI Communications. IOS Press, Vol. 7: 1, pp. 39-59,1994.

[6] Dennis Jeffrey “Test Suite Reduction with Selective Redundancy”, ICSM'05.

[7] James A. Jones “Test-Suite Reduction and Prioritization for Modified Condition/Decision Coverage”IEEE transactions, March 2003.

[8] Mats P.E. Heimdahl “Test-Suite Reduction for Model Based Tests: Effects on Test Quality and Implications for Testing”, Proceedings 19th International Conference on Automated Software Engineering,2004.

[9] Kartheek Muthyala “A Novel Approach To Test Suite Reduction Using Data Mining”,IJCSE, vol-2, june 2011.

[10] Lilly Raamesh “An Efficient Reduction Method for Test Cases”,IJEST,vol-2,2010.

[11] Hadi Hemmati “ Empirical Investigation of the Effects of Test Suite Properties on Similarity-Based Test Case Selection”, ICST '11

[12] G. Rothermel, M.J. Harrold, J. von Ronne, C. Hong, “Empirical Studies of Test-Suite Reduction”,Journal of Software Testing, Verification, and Reliability, 12(4), 2002, pp. 219-249. On Software Maintenance, IEEE Computer Society, 1998.

[13] M.J. Harrold, R. Gupta, M.L. Soffa, “A Methodology for Controlling the Size of a Test Suite”, ACM Transactions on Software Engineering Methodologies 2**,** 1993, pp. 270-285.

[14] T.Y. Chen, M. F. Lau, “Heuristics toward the Optimization of the Size of a Test Suite” Proc. 3rd Int’l Conf. on Software. Quality Management. Vol. 2, Seville, Spain, April 1995, pp. 415-424.

International Journal of Computer Science and Application
IJCSA Volume 1, Issue 1, August 2012 PP. 6-11
ISSN(online)
ISSN(print)
www.ij-csa.org

Design a New Architecure of Audio Amplifiers Class-D for a Hardware Mobile Systems

Krit Salahddine[1], Laassiri Jalal[2] and El Hajji Said[3]
[1]Polydisciplinary Faculty of Ouarzazate, University Ibn Zohr, BP/638, Morocco
[2]Department of informatics, Faculty of Sciences, University Ibn Tofail, BP 33, Morocco
[3]Department of Mathematic Informatics, University Mohamed V-Agdal, BP 1040, Morocco
Email: Krit_salah@yahoo.fr, laassiri.jalal@gmail.com,elhajji@fsr.ac.ma

(Abstract) This paper presents the advantages of using Audio amplifiers-class-D which has an important role in every mobile system involving an audible sound. General power amplifiers till recently have been very inefficient, bulky and unreliable. Though Class AB amplifiers have major market share in the audio industry because of their efficiency compared to previous classes of amplifiers such as Class A and Class B, recent demand for smaller devices with longer battery life has resulted in replacement of class AB amplifiers (linear amplifiers) with Class D (switching amplifiers). Class D amplifiers provide the balance between efficiency and distortion required by portable devices, hi-fi audio systems, as they utilize the switching operation where the transistors are either fully on or fully off resulting in amplification with zero power dissipation ideally. The main focus of this thesis is to analyze various design issues involved in the implementation of class D amplifiers. As many designers in the future will be switching to Class D amplifiers because of the recent advances in switching amplifiers, an effort was made to develop the thesis so as to be able to serve as a basic reference guide which gives them a good understanding of existing architectures, challenges in efficient power amplifier design, modulation methods, power stage topologies and implementation of class D amplifiers. A detailed study of parameters and parasitic that affect the performance of class D amplifiers has been carried out with design, implementation, and simulation of various stages. Various component selection decisions and layout issues have been discussed for an efficient, low distortion class D amplifier.
Keywords: Audio Amplifiers; Efficiency; Class D; Distortion; Modulation.

1. INTRODUCTION

The audio amplifier Class D is a switching amplifier that consists in a pulse width modulator (with switching frequency in order of several hundred kHz), a power bridge circuit and a low pass filter. This type of amplifier has demonstrated to have a very good performance. These include power efficiencies over 90%, THD under 0.01%, and low EMI noise levels that can be achieved with a good amplifier design.

The old version of the audio amplifiers is the class AB which has been used for driving a speaker load in portable devices including cell phones. However, their power efficiency is typically less than 20%, which reduces the overall systems battery lifetime and increases the heat dissipation.

This paper presents a the advantage of the use of class D amplifier for portable devices and a comparison with the other type of the audio amplifiers like class A, class B and class AB. The proposed work was experimentally verified using the technology 0.35-µm CMOS process; The amplifier achieves a power efficiency of 90% delivering up to 1.5W output power to an 8 ohm load, and operates with a 2.5V-5.5V supply.

2. THE ADVANTAGE AND THE INCONVENIENT OF THE DEFERENT AUDIO AMPLIFIERS

Class A – In a Class A amplifier, the output devices are continuously conducting for the entire cycle, or in other words there is always bias current flowing in the output devices Fig.1. This topology has the least distortion and is the most linear [2], but at the same time is the least efficient at about 20%. The design is typically not complementary with high and low side output devices.

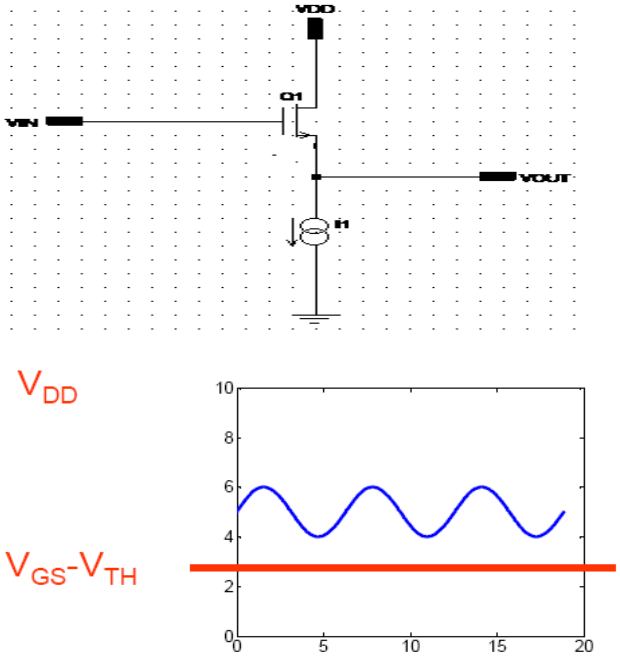

Fig.1. Amplifier Class A

Class B – This type of amplifier operates in the opposite way to Class A amplifiers. The output devices only conduct for half the sinusoidal cycle (one conducts in the positive region, and one conducts in the negative region), or in other words, if there is no input signal then there is no current flow in the output devices. This class of amplifier is obviously more efficient than Class A, at about 50%, but has some issue with linearity at the crossover point, due to the time it takes to turn one device off and turn the other device on.

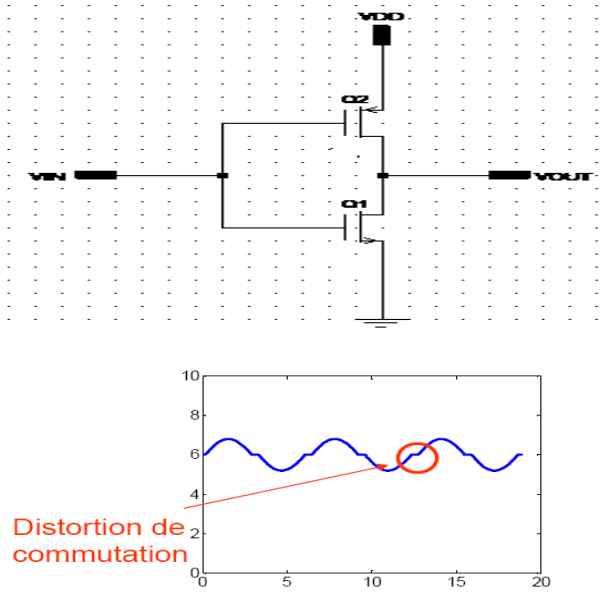

Fig.2. Amplifier Class B

Class AB – This type of amplifier is a combination of the above two types, and is currently one of the most common types of power amplifier in existence Fig.3. Here both devices are allowed to conduct at the same time, but just a small amount near the crossover point. Hence each device is conducting for more than half a cycle but less than the whole cycle, so the inherent non-linearity of Class B designs is overcome, without the inefficiencies of a Class A design. The efficiencies for Class AB amplifiers is about 50%.

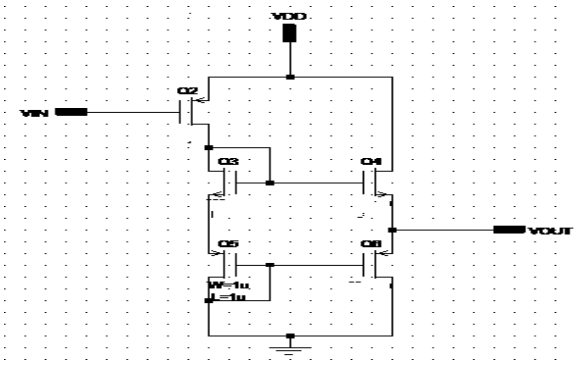

Fig.3.Amplifier Class AB

Class D – This class of amplifier is a switching or PWM (Pulse wide Modulation) amplifier as mentioned above[3], [4], [5]. This class of amplifier is the main focus of this application note. In this type of amplifier, the switches are either fully on or fully off, significantly reducing the power losses in the output devices [8].

Efficiencies of 90-95% are possible. The audio signal is used to modulate a PWM carrier signal, which drives the output devices, with the last stage being a low pass filter to remove the high frequency PWM carrier frequency. Class D amplifiers take on many different forms, some can have digital inputs and some can have analog inputs. Here, we will focus on the type which has analog inputs.

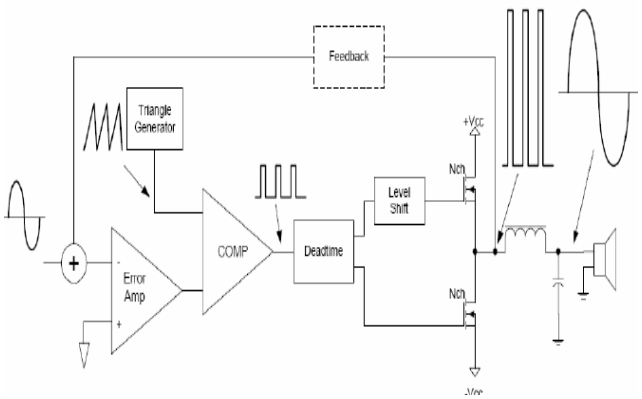

Fig.4. Block Diagram of a class D Amplifier.

Fig.4 above shows the basic block diagram for a Half Bridge Class D amplifier, with the waveforms at each stage. This circuit uses feedback from the output of the half-bridge to help compensate for variations in the bus voltages. So how does a Class D amplifier work?

A Class D amplifier works in very much the same way as a PWM power supply (we will show the analogy later). Let's start with an assumption that the input signal is a standard audio line level signal. This audio line level signal is sinusoidal with a frequency ranging from 20 Hz to 20 kHz typically. This signal is compared with a high frequency triangle or saw tooth waveform to create the PWM signal as seen in fig 5 below. This PWM signal is then used to drive the power stage, creating the amplified digital signal, and finally a low pass filter is applied to the signal to filter out the PWM carrier frequency and retrieve the sinusoidal audio signal (also seen in fig 4).

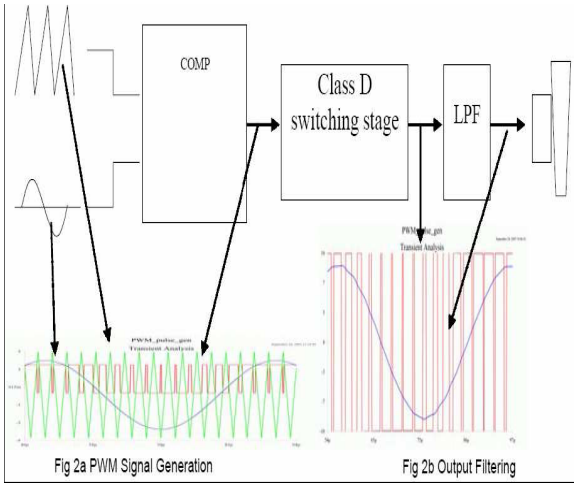

Fig.5. Class D Amplifier Waveforms

2.1 Design Challenges

A. Top view

The top schematic view of Class-D amplifier Fig. 6 is composed by a main part (inverting input integrator, triangle generator, PWM modulator, gate driver and MOSFET H-bridge) and secondary parts like common mode voltage and bias current generator, startup and protection circuit, input gain selector and shifter.

Two analog power supplies (tailed together with decoupling capacitor on the circuit board) are used with an ESD (Electro Static Dielectric) protection and bonding and are not shown is this schematic but have been taken into account for simulation.

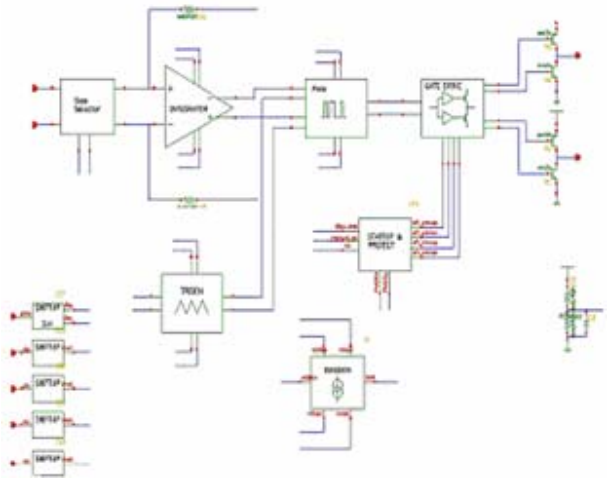

Fig.6. Class D schematic

2.2 Simulation Results And Comparison

The class D audio amplifier has been integrated in a 0.25um, double-poly, triple-metal, BiCMOS process, occupying an area of 1.5 x 1.2 mm2 fig.9 show the Layout of the proposed Amplifier Class D. The measured output peak current was measured to be 0.35A into an 8 ohm load, using a 3.6V supply voltage. The IC consumed 2.5mA of quiescent current under those conditions with an output power of 0.5W and a power efficiency of 79%. A power efficiency of 90%, Fig.8 shows that the comparison output efficiency between class-D and class-AB was achieved at a supply voltage of 6.5V. Fig. 10 shows the efficiency versus power curve. The total harmonic distortion plus noise (THD+N) was measured with an Audio Precision (AP) measurement system [6], [7], [8].

A low pass filter had to be used at the output since the AP measurement system cannot handle pure class D audio amplifiers because of the high frequency PWM waveform. The THD+N measured under the above-mentioned conditions vary from under 0.04% at low frequencies to a maximum of 0.4% at 7 kHz. Fig. 7 shows a plot of the THD+N versus the power under different supply voltage levels [9], [10], [10], [12]. Fig 9 shows layout of Amplifier Class D with process 0.18 um. Table 1 summarizes the measured results. The output offset is higher than predicted in the simulations because of mismatch in the feedback and in the input resistors.

B. Why use a low-pass RC output filter with Class-D amplifiers?

Class-D audio power amplifier (APA) families use a modulation scheme that does not always require an output filter for operation, but they do require some sort of low-pass filtering when making an output power measurement or a THD+N measurement. This is because the 250-kHz switching signal is seen as a common-mode voltage across

the inputs of the audio measurement instrument. Typically, audio analyzing equipment has low common-mode rejection at 250 kHz, because the equipment is designed to work in the audio band. Although most audio analyzing instruments have internal filtering, they still have input amplifiers that cannot respond to the fast rising edges of the PWM signal. The purpose of the RC filter is to remove the 250-kHz switching component from the PWM output of a Class-D amplifier.

C. Total Harmonic Distortion Plus Noise (THD+N)

The typical THD+N measurement combines the effects of noise, distortion, and other undesired signals into one measurement and relates it (usually as a percentage) to the fundamental frequency. Ideally, only the fundamental test frequency of the sine-wave input is present at the output of the APA, which in practice is never the case. The THD+N measurement requires notching out the fundamental test frequency and measuring the RMS voltage (which includes unwanted harmonics and noise) across the audio band (which the AP does automatically) and then dividing that measured value by the fundamental test frequency value and expressing it as a percentage.

D. Output Power

Audio power amplifier design typically uses speakers with impedances from 3 Ω to 32 Ω. When calculating the power, the output voltage (Vout) is specified as an RMS value and the following equationis used:

$$P_O = \frac{[V_{O(RMS)}]^2}{R_L} \quad (1)$$

Where,

PO = Output power

RL = Load impedance

And,

$$V_{O(RMS)} = \frac{V_{O(P)}}{\sqrt{2}} \text{ and } V_{O(RMS)} = \frac{V_{O(PP)}}{2\sqrt{2}} \quad (2)$$

Where,

VO(P) = Peak voltage

VO(PP) = Peak-to-peak voltage

It is important to understand the relationships among peak output voltage (VO(P)), peak-to-peak output voltage (VO(PP)), and the RMS output voltage (VO(RMS)) because these specific values are used when calculating the output power delivered to the load.

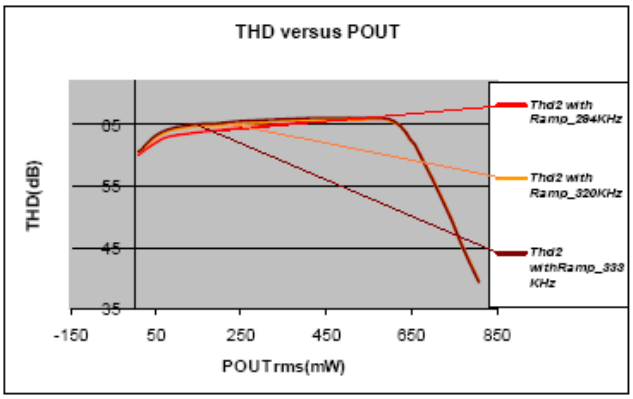

Fig. 7. Shows the THD+N versus the power.

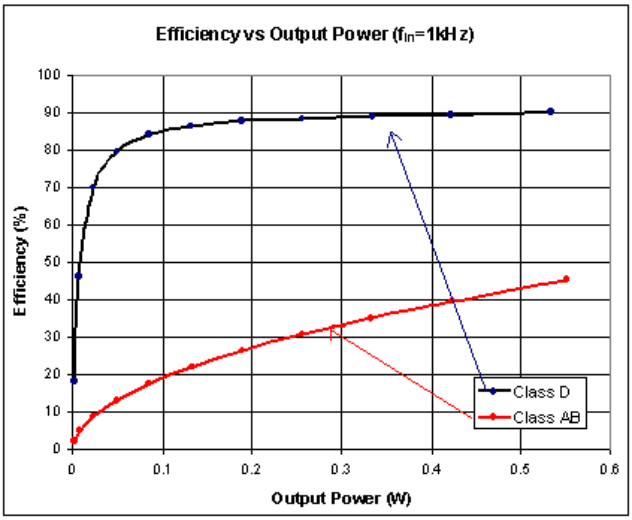

Fig.8. Efficiency of Class-D and Class-AB versus output power

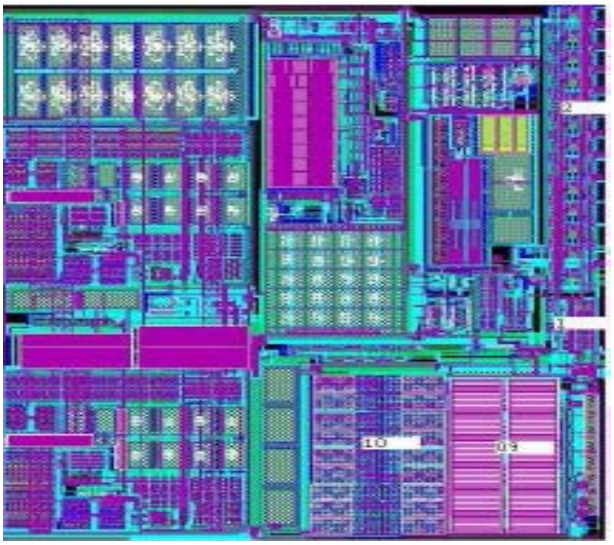

Fig.9. Layout of the proposed Amplifier Class D.

TABLE1 : COPARISON BETWEEN CLASS AB AND CLASS D.

	Class-AB	**Class-D**
Effeciency	50	90
Power Consuption	500mA	200 mA
Distorsio	0,5	0,1
Area	2,5 mm2	1,5mm2

3. CONCLUSIONS

We have presented a new class D audio amplifier for low voltage applications with high efficiency and minimum system solution size. The amplifier is a great improvement over its class AB counterparts where it comes to battery powered application since it dissipates less power in the amplifier itself.

REFERENCES

[1] Score, Michael, and Donald Dapkus. "Audio Power Amplifier Solutions for New Wireless Phones," Wireless Symposium, February 2000.

[2] Van der Zee, Ronan A. R., and Ed van Tuijl "A Power-Efficient Audio Amplifier Combining Switching and Linear Techniques," IEEE Journal of Solid-State Circuits, vol. 34, pp. 985-987, July 1999.

[3] Choi, Soo-chang, Jun-woo Lee, Woo-kang Jin, Jae-hwan So, and Suki Kim, "A Design of a 10- W Single Chip Class D Audio Amplifier with Very High Efficiency using CMOS Technology," IEEE Transactions on Consumer Electronics, vol. 45, pp. 465 – 473, August 1999.

[4] Jeong, Jay H., Hawn H. Seong, Jeong H. Yi, Gyu H. Cho, "A class D switching power amplifier with high efficiency and wide bandwidth by dual feedback loops," Proc. of International Conference on consumer Electronics, 1995, pp. 428 –429.

[5] Score, Michael. "Optimized Modulation Scheme Eliminates Output Filter," AES Preprint #5196, 109th Convention, pp. 22-25, September 2000.

[6] Chen, Wayne T. and R. Clif Jones, "Concepts and Design of Filterless Class-D Audio Amplifiers," Texas Instruments Technical Journal, vol. 18, No. 2, April 2001.

[7] Leigh, S. P, P. H. Mellor, and B. M. G. Cheetham, "Distortion Analysis and Reduction in a Completely

[8] Krit Salahddine and Hassan Qjidaa "Audio Power Amplifier Solutions for New Wireless Phones" IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.1, January 2009

[9] Krit Salahddine and Hassan Qjidaa "Ultra-low quiescent regulator Including precise reference and the Power-On-Reset circuit" Conference IEEE Sciences Electroniques, Technologies de l'Information et des Telecommunications (SETIT 05) Tunisia.

[10] Krit Salahddine and Hassan Qjidaa "Standard CMOS Low drop-out regulator with new dynamic Compensation" International Conf. on Modeling and Simulation, ICMS-2005 Marrakech.

[11] Krit Salahddine, Hassan Qjidaa "A novel CMOS charge-pump circuit with current mode control 110 mA at 2.7 V for telecommunication systems" JOS Journal of Semiconductors, VOL. 31 No. 4, April 2010

[12] Krit Salah-ddine , Zared Kamal, Qjidaa Hassan and Zouak Mohcine, "A 100 mA Low Voltage Linear Regulators for Systems on Chip Applications Using 0.18 µm CMOS Technology" IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 3, January 2012.

Author Introduction

Salah-ddine Krit received the B.S. and Ph.D degrees in Microectronics Engineering from Sidi Mohammed Ben Abdellah university, Fez, Morroco. Institute in 2004 and 2009, respectively. During 2002-2008.

He is also an engineer Team leader in audio and power management Integrated Circuits (ICs) Research. Design, simulation and layout of analog and digital blocks dedicated for mobile phone and satellite communication systems using CMOS technology. He is currently a professor of informatics with Polydisciplinary Faculty of Ouarzazate, Ibn Zohr university, Agadir, Morroco. His research interests include wireless sensor Networks (Software and Hardware), computer engineering and wireless communications. http://kritsalahddine.blogspot.com/

Jalal Laassiri received his Bachelor's degree (License es Sciences) in Mathematics and Informatics in 2001 and his Master's degree (DESA) in computer sciences and engineering from the faculty of sciences, university Mohammed V, Rabat, Morocco, in 2005, and he developed He received his Ph.D. degree in computer sciences and engineering from University of Mohammed V, Rabat, Morocco, in Juin, 2010. He was a visiting scientific with the Imperial College London, in London, U.K. He is Member of the International Association of Engineers (IAENG), He joined the Faculty of Sciences of Kénitra, Department of Computer Science , Ibn Tofail University, Morocco, as an Professor in October 2010, His current research interests include Software and Systems Engineering,

UML-OCL, B-Method, ..http://sites.google.com/site/laassirijalal/

Said El Hajji, Professor of Higher Education at Mohammed V - Agdal University, chief of Laboratory MIA, Faculty of Sciences, Rabat, Morocco.
http://www.fsr.ac.ma/mia/elhajji.htm

International Journal of Computer Science and Application
IJCSA Volume 1, Issue 1, August 2012 PP. 12-18
ISSN(online)
ISSN(print)
www.ij-csa.org

Solution of One-dimensional Hyperbolic Problems Using Cubic B-Splines Collocation

Christopher G. Provatidis[1], Stelios K. Isidorou[2]
[1,2] Department of Mechanical Engineering, National Technical University of Athens, Athens, Greece
Email: [1] cprovat@central.ntua.gr, [2] isidorou_st@yahoo.gr

(Abstract) In this paper we extend the cubic B-splines collocation method to enable it to solve one-dimensional hyperbolic (eigenvalue and wave propagation) problems under arbitrary boundary conditions. This is achieved by analogy with the finite element method, introducing a 'collocation mass matrix' that cooperates with the previously known system matrix, now called 'collocation stiffness matrix'. In agreement with earlier findings on elliptic problems, we found that in time-dependent problems it is again sufficient to use double internal knots (C^1-continuity) in conjunction with two collocation points between successive breakpoints. In this way, the number of unknowns becomes equal to the number of equations, which is twice the number of breakpoints. We paid particular attention to the handling of Neumann-type boundary conditions, where we found it necessary to properly eliminate a column in both mass and stiffness matrices. For the first time, we found that the cubic B-splines collocation procedure (with C^1-continuity) leads to identical results with those obtained using piecewise Hermite collocation. The numerical examples show an excellent quality of the numerical solution, which is far superior to that of the conventional finite element method, for the same number of nodal points.

Keywords: B-splines; Collocation; Eigenvalues; Hermite Polynomials; MATLAB™; Transient Analysis.

1. INTRODUCTION

In the framework of global interpolation methods based on Lagrange polynomials, it has been recently shown that the computation effort may be substantially reduced when applying the collocation instead of the Galerkin-Ritz formulation [1]. However, using spline curves, or piecewise polynomials, is more effective in representing the solution to the differential equation than pure polynomials [2]. The volume of Ascher et al. [3] provides a treatise on spline bases, collocation theory, and spline collocation for application to the numerical solution of boundary-value-problem (BVP) for ordinary differential equations (ODE). Fairweather and Meade [4] give an extensive review (273 papers covering the period 1934–1989) of collocation methods and various implementations. They describe the most common forms of collocation, including nodal, orthogonal, and collocation/Galerkin. An early work towards the solution of eigenvalue problems, however based on Schoenberg's formulation [5], is [6].

In spite of the abovementioned work, so far most part of the relevant research focuses on mathematical topics and reduces mainly to elliptic problems. For example, the FORTRAN codes of [2] have been implemented also in the MATLAB software (formerly under the name *splines tool*) [7], but they operate 'as is' for the solution of linear and nonlinear elliptic problems only. This happens because (i) the eigenvalue and transient analysis require the construction of mass and stiffness collocation matrices, and (ii) although the implementation of Dirichlet type boundary conditions is a trivial task, the same does not hold for Neumann-type ones that require a special treatment. As for the state-of-the-art, the implementation of the function 'spcol' in time-dependent analysis has been made in very few biometrics [8] and chemical engineering applications [9].

In this context, the primary aim of this paper is to investigate the applicability and performance of deBoor's methodology [2] in the numerical solution of hyperbolic problems (eigenvalue and transient). The main novel feature of this work is the development of *mass* and *stiffness* collocation matrices analogous to the finite element method. Consequently, standard eigenvalue analysis based on the QR algorithm, and standard time-integration techniques such as the central difference method will be applied.

In addition to the abovementioned *global* cubic B-splines collocation, *piecewise*-Hermite polynomials (without upwind features) that act between adjacent breakpoints [10] will be compared for the first time. The finding of coincidence is the secondary novel feature of this work.

The theory is sustained by four one-dimensional numerical examples from the field of applied mechanics.

2. FORMULATION

Below is the formulation of typical static (thermal) and dynamic problems and then follows the global B-spline and piecewise Hermite interpolation.

2.1 Thermal Analysis

The steady-state thermal behavior of structures follows Laplace equation. In the particular case of a radially symmetric structure the PDE degenerates to a simple ODE in terms of the radius *r* as follows:

$$\frac{\partial^2 u}{\partial r^2}+\frac{1}{r}\frac{\partial u}{\partial r}=0 \qquad (1)$$

2.2 Elastodynamics – Wave equation

The vibration of an elastic rod is described through the usual hyperbolic differential equation:

$$\frac{\partial}{\partial x}\left(EA\frac{\partial u}{\partial x}\right)+f_x=\rho A\frac{\partial^2 u}{\partial t^2} \qquad (2)$$

where E and ρ denote the material properties, i.e. the Young's modulus and the mass density, respectively, and A denotes the cross sectional area of the rod. Finally, f_x denotes the distributed load along the longitudinal *x*-axis, while *t* denotes the time. In dynamic analysis, the axial displacement, *u*, is a function of both space and time (i.e. $u=u(x,t)$), while in static analysis it is only a function of space (i.e. $u=u(x)$).

Introducing the velocity of the longitudinal elastic wave:

$$c=\sqrt{E/\rho}\,, \qquad (3)$$

in the absence of body forces ($f_x=0$), **Eq.2** receives the well known form of an acoustic wave:

$$\left(1/c^2\right)\cdot\partial^2 u/\partial t^2-\partial^2 u/\partial x^2=0\,. \qquad (4)$$

It is well known that, besides elastic waves, **Eq.4** can also describe wave propagation within acoustic pipes of constant cross-sectional area; in such a case the variable *u* corresponds to the acoustic pressure.

2.3 B-splines Interpolation

B-splines interpolation can be found in many textbooks [2,11], in which efficient procedures to determine the involved basis functions, $N_{i,p}(x)$, can be found. In general, the interpolation within an interval $0\le x\le L$ is given by:

$$u(x)=\sum_{i=1}^{n}N_{i,p}(x)\cdot a_i,\quad i=1,\ldots,n \qquad (5)$$

Concerning **Eq.5**, it should become clear that:

The starting point for the construction of the functional set, $N_{i,p}$, is the selection of the breakpoints and the piecewise polynomial degree, *p*. Let us assume *m* breakpoints, which obviously define (*m*-1) segments. In the sequence, let the polynomial degree be *p* = 3 (cubic B-splines).

(i) The variables a_i involved in **Eq.5** are not directly associated to nodal values (those at the breakpoints), except of those two corresponding to the end points.

(ii) The value of '*n*' depends on the multiplicity chosen for the internal nodes. In this paper, we require C^1*-continuity*, thus the multiplicity must be equal to *two*; therefore, '*n*' is twice the number of breakpoints ($n=2m$). In general, the spline curve is C^{p-1}-continuous everywhere except at the location of the repeated knots where it is C^{p-2}-continuous.

(iii) Therefore, if we choose two collocation points within any of the aforementioned (*m*-1) segments defined by the breakpoints, we can obtain so many equations as the number of the unknown coefficients. In case of elliptic problems, the best choice was found to be two Gauss points at the well-known relative positions ($\xi=\pm 1/\sqrt{3}$), as also was found by deBoor and Swartz [12] (in elliptic problems).

Typical plots of basis-functions are shown in **Figure 1**, in which double knots have been considered. It is noted that both the first and the last basis functions correspond to the left-end and the right-end axial displacement, respectively; these are also a mirror of each other with respect to the center of the interval [0,*L*], and equal to unity at the two ends.

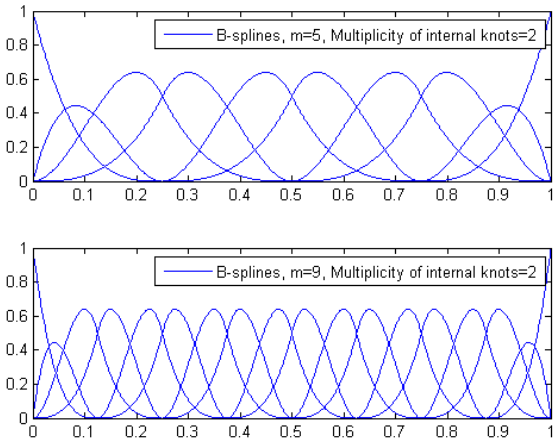

Figure 1. Basis functions in the domain [0,1]. In the top and the bottom there are four subintervals (*m* = 5) and eight subintervals (*m* = 9), respectively; in both cases the internal breakpoints are considered with multiplicity of internal knots equal to *two*.

2.4 Piecewise-Hermite Interpolation

The domain is again uniformly divided into (*m*-1) segments using *m* nodal points (the ends are included). Starting from left to the right, the nodes are numbered by ascending order, whereas the 2*m* degrees of freedom are as follows: $(u_1,q_1),(u_2,q_2),\ldots,(u_m,q_m)$ with $q_i=\left(\partial u/\partial x\right)_{x=x_i}$. In this way, having considered the flux as an independent DOF, C^1-continuity is ensured.

Between any of the (*m*-1) segments, in local numbering the variable *u* is approximated by:

$$u(x,t) = N_1(x)u_1(t) + N_2(x)q_1(t) + N_3(x)u_2(t) + N_4(x)q_2(t) \quad (6)$$

where the shape functions are the well-known Hermite polynomials (**Figure 2**), which are expressed in terms of the normalized coordinate $w = (x/L)$:

$$\begin{aligned} N_1(w) &= 1 - 3w^2 + 2w^3 = 1 - 3\left(\frac{x}{L}\right)^2 + 2\left(\frac{x}{L}\right)^3 \\ N_2(w) &= w - 2w^2 + w^3 = \left[\left(\frac{x}{L}\right) - 2\left(\frac{x}{L}\right)^2 + \left(\frac{x}{L}\right)^3\right]L \\ N_3(w) &= 3w^2 - 2w^3 = 3\left(\frac{x}{L}\right)^2 - 2\left(\frac{x}{L}\right)^3 \\ N_4(w) &= -w^2 + w^3 = \left[-\left(\frac{x}{L}\right)^2 + \left(\frac{x}{L}\right)^3\right]L \end{aligned} \quad (7)$$

It is noted that the above notation presumes that the fluxes q_1 and q_2 are both positive when directed to the positive *x*-axis. Then q_1 causes 'compression' while q_2 causes 'tension'; also the corresponding shape functions in **Figure 2** become anti-symmetric with respect to the middle of the interval [0,*L*] and obtain positive and negative values, respectively. If the convention for q_2 changes, both shape functions may become positive and symmetric with respect to the middle of the domain. Then, the 'mirror' property is obtained for both translational (u_1, u_2) and rotational (q_1, q_2) degrees of freedom.

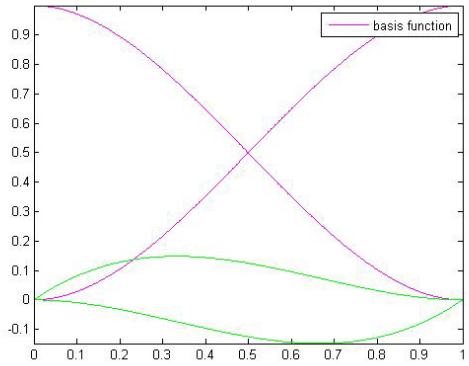

Figure 2. Hermite polynomials defined in [0,1] (blue: translational DOF (N_1,N_3), green: rotational DOF (N_2,N_4)).

3. COMPUTATIONAL PROCEDURE

In all cases below, the ODE is collocated at the $n = 2m$ Gauss points in the interior of the subintervals, i.e. at the locations:

$$x_{col} = \frac{(x_i + x_{i+1})}{2} \pm \frac{1}{\sqrt{3}} \cdot \frac{(x_i - x_{i+1})}{2}, \quad i = 1,\ldots,(m-1) \quad (8)$$

Below, the computational procedure for dealing with eigenvalue and wave propagation problems is presented.

3.1 Eigenvalue Problem

Collocating **Eq.4** at the abovementioned $n = 2m$ Gauss points one obtains the well known matrix formulation [1]:

$$[\mathbf{M}]\{\ddot{\mathbf{a}}(t)\} + [\mathbf{K}]\{\mathbf{a}(t)\} = \{\mathbf{f}(t)\} \quad (9)$$

where $[\mathbf{M}]$ and $[\mathbf{K}]$, both of dimensions $(n-2)\times n \equiv 2(m-1)\times 2m$, are the nonsymmetric mass and stiffness collocation matrices, respectively, which are given by:

$$\begin{aligned} m_{ij} &= \left(\frac{1}{c^2}\right)N_j(x_i), \\ k_{ij} &= -N_j''(x_i), \quad i = 1,\ldots,2(m-1), \quad j = 1,\ldots,2m \end{aligned}, \quad (10)$$

with $N_i(x)$ denoting the shape functions (in global numbering) at the collocation point x_i, in ascending order from left to the right.

It is remarkable that **Eq.9** covers both cases, i.e. the global B-splines interpolation (cf. **Eq.5**) as well as the piecewise Hermite interpolation (cf. **Eq.6** and **Eq.7**).

Concerning the eigenvalue (free vibration) problem, the force vector $\{\mathbf{f}(t)\}$ is taken equal to zero and, therefore, the eigenvalues are extracted by requiring:

$$\det\left\|[\mathbf{K}] - \omega^2[\mathbf{M}]\right\| = 0 \quad (11)$$

In this study we used the standard MATLAB function 'eig'.

3.2 Wave Propagation Problem

The wave propagation problem can be generally solved using either modal analysis or one of the well known time-integration methods (explicit, implicit). In this paper we choose the *central difference* method [13]:

$$\ddot{u}_n = \frac{1}{\Delta t^2}(u_{n-1} - 2u_n + u_{n+1}) \text{ and } \dot{u}_n = \frac{1}{2\Delta t}(u_{n+1} - u_{n-1}). \quad (12)$$

Substituting **Eq.12** into **Eq.9**, the last written for $t_{n+1} = t_n + \Delta t$, one obtains:

$$\left(\frac{1}{\Delta t^2}\mathbf{M}\right)\mathbf{u}_{n+1} = \mathbf{f}_n - \left(\mathbf{K} - \frac{2}{\Delta t^2}\mathbf{M}\right)\mathbf{u}_n - \left(\frac{1}{\Delta t^2}\mathbf{M}\right)\mathbf{u}_{n-1}, \quad (13)$$

from which we can solve for $\mathbf{u}_{n+1}$.

3.3 Boundary Conditions

Despite the apparent simplicity of the procedure, some technical difficulties appear and require special treatment and explanation. First of all, it is reminded that two DOF exist at every nodal point (or breakpoint) for the piecewise cubic Hermite (or B-splines) collocation formulation. Concerning the left and right ends, the DOF correspond to the quantities *u*

and $q=\left(\partial u/\partial x\right)_{x=0 \text{ or } L}$. In every formulation, the vector of the coefficients is as follows (it is reminded that the domain is uniformly divided into (*m*-1) segments using *m* nodal points, where the ends are included):

Piecewise cubic Hermite:

$$\left\{u_1, q_1, u_2, q_2, \ldots, u_{m-1}, q_{m-1}, u_m, q_m\right\}^T$$

Cubic B-splines:

$$\left\{a_1 \equiv u_1, a_2, a_3, a_4, \ldots, a_{2m-3}, a_{2m-2}, a_{2m-1}, a_{2m} \equiv u_m\right\}^T$$

Therefore, for a bar fixed at both ends (or an acoustic pipe open at its both ends), in the piecewise Hermite formulation it is sufficient to eliminate the first ($u_1=0$) and the last minus one ($u_m=0$) columns of the matrices [M] and [K], while in B-splines formulation we must eliminate the first and the last column (it is reminded that B-splines have the "mirror" property).

The situation becomes more complex when dealing with a free end (Neumann-type), for example at *x*=*L*. In this case the strain (or flux, or velocity) is zero or has a given value, but it is *not* always allowed to eliminate the corresponding column. In more details:

(i) In piecewise Hermite formulation the DOF that corresponds to the vanishing flux at the end is quite independent of the rest DOFs, and therefore it can be directly eliminated. In contrast to the FEM analysis, none row is eliminated but only the corresponding columns. Therefore, starting from a matrix system of $\left(n-2\right)\times n \equiv 2\left(m-1\right)\times 2m$ dimensions, the elimination of two DOF leads to a system of $\left(n-2\right)\times\left(n-2\right) \equiv 2\left(m-1\right)\times 2\left(m-1\right)$ dimensions.

(ii) In B-splines formulation the elastic strain (or the acoustic flux/velocity) at the right free end (*x*=*L*) is not an independent variable but it can be derived by taking the first derivative of **Eq.5**:

$$q\left(L\right) \equiv u'\left(L\right) = \sum_{i=1}^{n} N'_{i,p}\left(L\right)\cdot a_i \tag{14a}$$

Due to the compact support of the basis functions, for *p*=3 only the last two out of the *n*=2*m* basis functions have a non-vanishing derivative at *x*=*L*. Therefore, it holds:

$$\begin{aligned} & q\left(L\right) = N'_{n-1}\left(L\right)\cdot a_{n-1} + N'_n\left(L\right)\cdot a_n \\ & \Rightarrow a_{n-1} = \frac{\left[q\left(L\right) - N'_n\left(L\right)\cdot a_n\right]}{N'_{n-1}\left(L\right)} \end{aligned} \tag{14b}$$

For the solution of the eigenvalue problem we can assume that $q\left(L\right)=0$, and therefore **Eq.14b** is somehow simplified. Concerning the matrix term $\left(\left[\mathbf{K}\right]-\omega^2\left[\mathbf{M}\right]\right)\cdot\left\{\mathbf{a}\right\}$, every row can be transformed from its initial form:

$$\begin{aligned} & \left(k_{i1}a_1 + \ldots + k_{i,n-2}a_{n-2} + k_{i,n-1}a_{n-1} + k_{i,n}a_n\right) \\ & -\omega^2\left(m_{i1}a_1 + \ldots + m_{i,n-2}a_{n-2} + m_{i,n-1}a_{n-1} + m_{i,n}a_n\right) \end{aligned} \tag{15}$$

to the final expression:

$$\begin{aligned} & \left[k_{i1}a_1 + \ldots + k_{i,n-2}a_{n-2} + \left(k_{i,n} - N'_n\left(L\right)/N'_{n-1}\left(L\right)\cdot k_{i,n-1}\right)a_n\right] \\ & -\omega^2\left[m_{i1}a_1 + \ldots + m_{i,n-2}a_{n-2} + \left(m_{i,n} - N'_n\left(L\right)/N'_{n-1}\left(L\right)\cdot m_{i,n-1}\right)a_n\right] \end{aligned} \tag{16}$$

Therefore, concerning again B-splines collocation, in case of one Dirichlet boundary condition at *x*=0 and one Neumann condition at *x*=*L*, we proceed as follows. First we eliminate the first column ($a_1=u_1=0$). Then we condense the last two columns in one by subtracting the (*n*-1)-th column (multiplied by the coefficient $N'_n\left(L\right)/N'_{n-1}\left(L\right)$) from the last one. In this way we finally derive a system of $\left(n-2\right)\times\left(n-2\right) \equiv 2\left(m-1\right)\times 2\left(m-1\right)$ dimensions.

Note that the abovementioned condensation is not applicable to the piecewise Hermite formulation, obviously because $N'_3\left(L\right)=0,\ N'_4\left(w\right)=1$.

4. NUMERICAL RESULTS

The performance of the proposed global collocation method will be evaluated in four characteristic test cases: one static (to show the convergence quality) and three dynamic ones. As previously mentioned, from an engineering point of view the dynamic problems may concern either an elastic rod of constant cross-sectional area or, equivalently, an acoustic straight pipe. Since previous studies [12,14] concerning the elliptic problem have shown that the best choice is to use *cubic* B-splines in conjunction with collocation points taken at the two Gauss points between the uniformly distributed breakpoints, results will be presented for these conditions only. For comparison purposes, the collocation method is compared with the conventional finite element (FEM) solution, using the same nodes as the breakpoints.

4.1. Problem 1: Steady-state Conduction in a Cylindrical Wall

This example was chosen because it possesses a non-polynomial (logarithmic) solution. It refers to the potential problem (Laplace equation) with a "steep" temperature distribution, occurring along the radius of a thick long hollow cylinder. The latter is subject to given uniform inner surface temperature (u_1 = 1000 ^{o}C) and a given uniform outer surface temperature (u_2 = 0 ^{o}C) while the inner and outer radii are R_1 = 1m and R_2 = 32m, respectively. The analytical temperature distribution depends on the radial direction only:

$$u\left(r\right) = u_1 + \left(u_2 - u_1\right)/\ln\left(R_2/R_1\right)\cdot\ln\left(r/R_1\right) \tag{17}$$

The radius is divided into a variable number of *n* = 4, 8, 16, 32 and 64 uniform segments. Based on the normalized L_2 error norm ($L_u = L_2\times 100$ in %):

$$L_2 = \sqrt{\int_{R_1}^{R_2} \left(\tilde{u} - u_{exact}\right)^2 dx} \Big/ \sqrt{\int_{R_1}^{R_2} \left(u_{exact}\right)^2 dx}\ , \qquad (18)$$

The excellent quality of the convergence is shown in **Table 1**.

Table 1. Convergence quality of the error norm.

Number of subdivisions (n)	Energy norm L_u (in %)
4	1.19E00
8	0.73E-01
16	0.20E-02
32	2.54E-05
64	1.74E-07

4.2 Problem 2: Elastic Rod Fixed at Both Ends

A rod of length L is fixed at $x = 0$ and $x = L$ and is subject to axial free vibration (equivalently, we could refer to an acoustic pipe with both ends open). The exact eigenvalues are given by:

$$\omega_i^2 = \left(i\pi/L\right)^2 \cdot \left(E/\rho\right),\ i = 1, 2, \ldots \qquad (19)$$

The rod is uniformly divided into 1, 2, 4, 8, 16, 32 and 64 segments. The numerical results are shown in **Figure 3** where one can notice that the convergence is monotonic, of high quality and far superior to the conventional FEM solution.

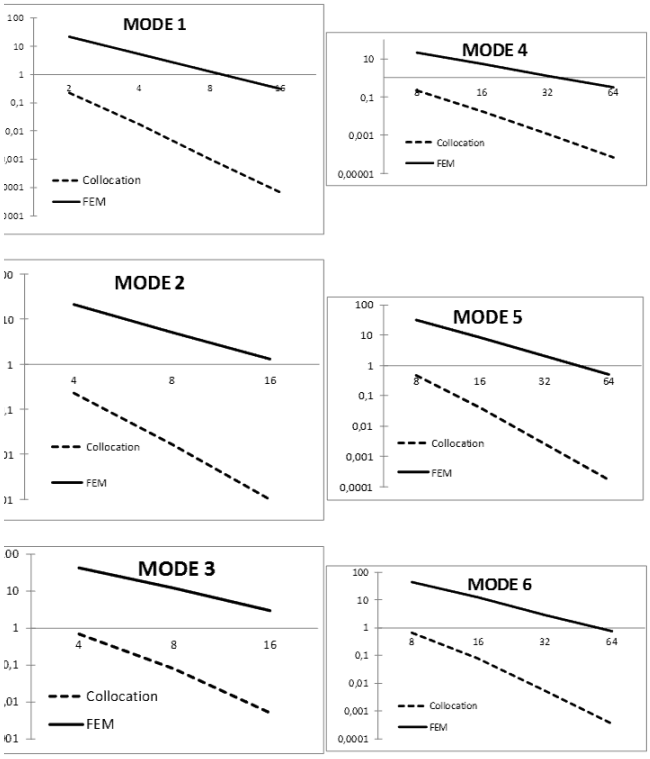

Figure 3. Errors (in %) of the calculated eigenvalues for an elastic rod with fixed ends (the horizontal axis refers to the number n of elements or subintervals of breakpoints).

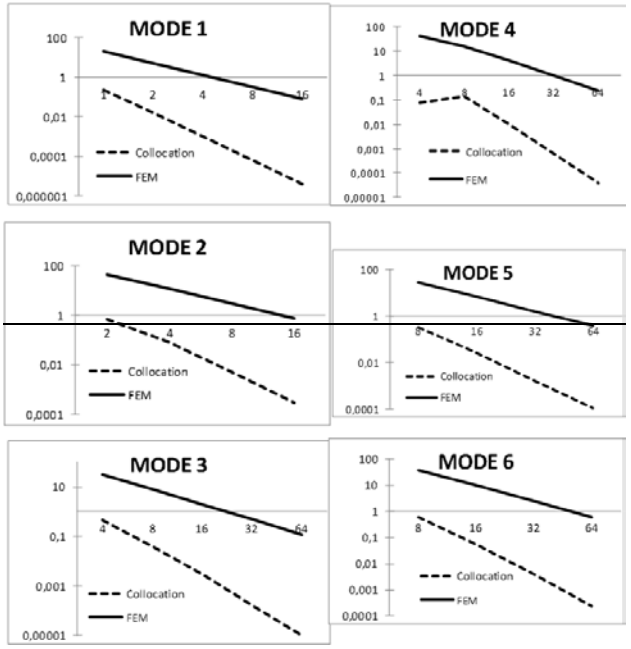

Figure 4. Errors (in %) of the calculated eigenvalues for an elastic rod with fixed left and free right end (the horizontal axis refers to the number n of elements or subintervals of breakpoints).

4.3 Problem 3: Elastic Rod Fixed at Left and Free at Right End

The same rod is now analyzed for different boundary conditions (the equivalent acoustic pipe is open at left and close at right end). The quality of the calculated eigenvalues and the superiority to the FEM is shown in **Figure 4**, where it is expressed as an error (per cent) with respect to the exact solution:

$$\omega_i^2 = \left[\left(2i-1\right)\pi/\left(2L\right)\right]^2 \cdot E/\rho,\ i = 1, 2, \ldots \qquad (20)$$

4.4 Problem 4: Elastic Rod Subjected to a Heaviside Load

An elastic rod of length L is fixed at one of its extremities (x=0) and it is subjected to an axial Heaviside type loading $\sigma_L = Eq_0 H\left(t-0\right)$ [N/m^2] at the other one (x=L). The analytical solution can be found in [15,16]. For simplicity, all geometric and material data were assigned the unitary value. The (explicit) central difference scheme was applied.

For the case of n=8 uniform intervals ($c = \sqrt{E/\rho}$ =1m/s), where the distance between two successive nodes equals to Δx = 0.125m, a very upper limit for the time step could be $\Delta \hat{t}_{cr} = 0.125$ (CFL-criterion). However, for the sake of conservatism, the time step has been conservatively chosen equal to $\Delta t = 0.02\,\Delta x/c$ =0.0025 s. The time history (6000 steps) of the normalized axial displacement at the free end

(x=L) and the middle point (x=L/2) is shown in **Figure 5** and is of excellent quality.

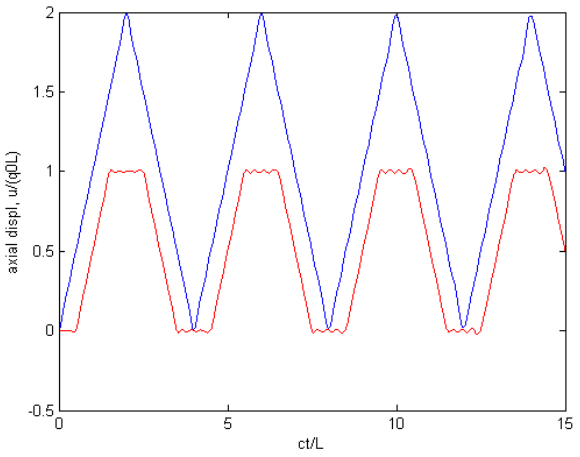

Figure 5. Axial displacement at the free end (x=L) [blue line] and the middle [red line], using the proposed collocation method n = 8 uniform intervals and time step $\Delta t = 0.2\,\Delta x/c = 0.0025$ s

5. DISCUSSION

Similar to previous findings that were obtained for polynomials [1], the superiority of the proposed collocation method to that of the finite element method is due to its global character and the relatively high degree of the piecewise polynomials (p = 3). Clearly, the cubic B-splines interpolation is very smooth and can accurately approximate the involved eigenvectors, while the conventional finite element method uses piecewise linear (hat) shape functions that appear sharp edges at their interfaces (they posses only C^0-continuity).

In all elliptic and hyperbolic numerical examples of this study we found that the cubic B-splines collocation method leads to identical results with those obtained using the piecewise cubic Hermite collocation method. This numerical finding can be theoretically explained as follows. Since two knots are associated to each breakpoint (multiplicity equal to two) it implies that the splines curve is generally C^{p-2}-continuous, where p is the degree of the piecewise polynomial. But as in this study the polynomial degree was chosen to be cubic ($p=3$), the splines curve is C^1-continuous, a fact that is consistent with the C^1-continuity involved in piecewise Hermite approximation (also of third degree).

Although in this type of problems the proposed B-splines interpolation is identical (as for the quality of the obtained results) with the piecewise cubic Hermite collocation method, the first is the vehicle to choose any multiplicity for the internal knots. For example, although no relevant results were reported in this study, we could alternatively choose the multiplicity be equal to *one*, instead of two. In such a case the numerical solution would be C^2–continuous, as it would include a full cubic polynomial plus some Schoenberg's truncated cubic monomials [5]. For example, if we divide the domain [0,L] into four segments, cubic B-splines approximation would lead to seven control points and seven associated coefficients, a_i, $i=1,\ldots,7$. However, since we deal with a two-point BVP, we have to determine only five out of the seven aforementioned coefficients. To this purpose one must choose at least five collocation points within the domain of length L. Obviously, one could use even more than five collocation points but then, since the number of equations would be larger than the number of variables, a least-squares scheme had to be applied. In the same example, although a choice of five (or more, in combination with the least-squares method) Gauss points or Chebyshev roots generally works, the quality of results becomes questionable while the DeBoor's approach (double knots, and two collocation points between successive breakpoints) is always mathematically robust [12].

6. CONCLUSIONS

In this study we achieved to extend the well-known cubic B-splines collocation method, previously used in the solution of elliptic problems only, to one-dimensional hyperbolic ones under arbitrary (Dirichlet of Neumann) boundary conditions. The key point was that, in addition to the system matrix, we achieved to create a mass collocation matrix; thus the computational problem was formulated very similarly to the well known finite element method. Consequently, the eigenvalues were calculated in an algebraic way using standard QR algorithms, while wave propagation (time response) was predicted using a simple central-difference scheme for time-integration. The findings of this study show that the proposed technique has an excellent convergence and its overall performance is far superior to the conventional finite element method for the same number of nodal points. Another interesting finding is that, B-splines collocation based on two knots per breakpoint leads to *identical* eigenvalues and time response with those obtained using again the collocation method but in conjunction with piecewise cubic Hermitian polynomials. Although the results of this study reduced to 1D problems only, on-going research has revealed that extension to 2D (quadrilaterals) and 3D (boxlike) time-dependent problems is a straightforward procedure on the basis of tensor products of the involved control points per direction.

REFERENCES

[1] C. G. Provatidis, Free vibration analysis of elastic rods using global collocation, Archive of Applied Mechanics 78, 241–250 (2008)

[2] C. deBoor, The numerical solution of an ordinary differential equation by collocation, in A Practical Guide to Splines, Edited C. deBoor, Revised Edition, Springer-Verlag, New York, (2001) pp.243-262.

[3] U. M. Ascher, R. M. M. Mattheij and R. D. Russell, Numerical Solutions of Boundary Value Problems for Ordinary Differential Equations (2nd edn), SIAM, Philadelphia (1995)

[4] G. Fairweather and D. Meade, Survey of spline collocation methods for the numerical solution of differential equations, in Mathematics for Large Scale Computing, Edited J. C. Diaz, Lecture Notes in Pure Applied Mathematics, Marcel Dekker, New York (1989), vol. 120, pp. 297–341.

[5] I. J. Schoenberg, Contributions to the problem of approximation of equidistant data by analytic functions, Quart. Appl. Math. 4, 45–99 and 112–141 (1946)

[6] J. W. Jerome and R. S. Varga, Generalizations of spline functions and applications to nonlinear boundary value and eigenvalue problems, in Theory and Applications of Spline Functions, Edited T. N. E. Greville, Academic Press, New York (1969), pp.103-155.

[7] C. deBoor, Private e-mail communications, June-August 2007.

[8] M. Hauptmann, J. Wellmann, J. H. Lubin, P. S. Rosenberg and L. Kreienbrock, Analysis of expose-time-response relationships using a spline weight function, Biometrics 56, 1105-1108 (2000).

[9] H. Pedersen and M. Tanoff, Spline collocation method for solving parabolic PDE's with initial discontinuities: Application to mixing with chemical reaction, Computers & Chemical Engineering 6, 197-207 (1982)

[10] W. Sun, Hermite cubic spline collocation methods with upwind features, ANZIAM J. 42 (E), C1379-C1379 (2000).

[11] L. Piegl and W. Tiller, The NURBS Book, 2nd ed., Springer, Berlin (1996)

[12] C. DeBoor and B. Swartz, Collocation at Gaussian points, SIAM J. Numer. Anal. 10, 582-606 (1973)

[13] K.J. Bathe, Finite Element Procedures. Prentice-Hall/Upper Saddle River, New Jersey (1996), pp. 770-774.

[14] C. G. Provatidis and S. K. Isidorou, Comparison of advanced collocation methods for the solution of ordinary differential equations. In: D. Tsahalis (ed.), CD Proceedings 3rd International Conference on Experiments / Process / System modeling / Simulation & Optimization, Athens, 8-11 July, 2009.

[15] L. A. Pipes and L. R. Harvill, Applied Mathematics for Engineers and Physicists, 3rd ed. McGraw-Hill International Book Company, International Student Edition, (1981) pp.494-496.

[16] C. G. Provatidis, Coons-patch macroelements in two-dimensional eigenvalue and scalar wave propagation problems, Computers & Structures 82, 383-395 (2004).

International Journal of Computer Science and Application
IJCSA Volume 1, Issue 1, August 2012 PP. 19-25
ISSN(online)
ISSN(print)
www.ij-csa.org

Flexible Construction of High-Girth Qc-Ldpc Codes

Gabofetswe Malema
Department of Computer Science, University of Botswana, Gaborone, Botswana
Email: malemag@mopipi.ub.bw

(Abstract) This article presents a highly flexible method for constructing high-girth quasi-cyclic low-density parity-check (QC-LDPC) codes. The proposed algorithm constructs a Tanner graph formed by connecting groups of rows (check nodes) and columns (variable nodes) of the constructed code. To obtain a sub-matrix structure, rows and columns are divided into groups of equal sizes. Rows and columns in a group are connected in their numerical (positional) order to obtain a cyclic structure. Connected rows and columns must satisfy the desired minimum cycle length. We present conditions that guarantee desired girths. The proposed algorithm is by far more flexible in constructing a wide range (rates and lengths) of regular and irregular QC-LDPC codes compared to existing methods. The algorithm, which has linear complexity with respect to the number of rows or columns, provides an easy and fast way to construct QC-LDPC codes. Constructed codes show good bit error rate performances.

Keywords: QC-LDPC Codes; Tanner Graph; Girth; Code Rate and Length.

1. INTRODUCTION

Quasi-Cyclic Low-Density Parity-Check (QC-LDPC) codes have been shown to be easily implementable in both encoder and decoder because of their block and cyclic properties [1]. They have also been shown to perform close to Shannon's capacity limit [2]. Performance of LDPC codes could be further improved by constructing codes with larger girths [3]. There are several methods for constructing QC-LDPC codes including algebraic, finite geometry, graphical and combinatorial techniques, examples of which are found in [4]--[8]. These construction methods avoid four-cycles by employing the row-column constraint. That is, no two rows share the same column more than once and vice versa. Although these methods can be used to construct a wide range of codes, they are generally limited in producing codes with arbitrary rates and lengths. They are also limited in their structure in that the number of sub-matrices and their configuration is fixed or limited in variability. The lack of flexibility constrains the variety of codes that could be constructed and may also constrain hardware implementation.

In [7] girth 12 QC-LPDC codes are obtain based on special graphs in which three connected vertices are used to form a column. The limitations of this method is that, it works with (*3,k*) QC-LDPC codes, the sub-matrices sizes are not arbitrary and row and column weights are not always regular. QC-LPDC codes from finite geometries such as Euclidean (EG) and projective geometries (PG) are also limited in girth and code dimensions [6]. The same is true with QC-LDPC codes from Balanced Incomplete Block Designs (BIBD)[8]. The constraints for satisfying girths also involve use of prime numbers which restricts the size of sub-matrices. Other QC-LDPC codes construction methods such as one in [9] are also restrictive. The method guarantees girth 8 but restricts dimensions of obtained codes to multiples of $3k^2$xk^3, where k is code row-weight. One highly flexible construction method is found in [5] in which an arbitrary base or seed matrix could be used. A wide variety of rates and column and row weights can be used. The method sets algebraic conditions under which cycles of a particular girth are avoided for a given seed matrix. However, the conditions do not allow for all sizes of sub-matrices, p. A similar algorithm is found in [10] in which combinatorial designs are used to derive protographs that can result in girths higher than 12. This construction method also does not result in flexible codes and most obtained codes are also very large to be practically useful.

In this article we propose a method for constructing high-girth QC-LDPC codes. The method is highly flexible in that regular and irregular codes could easily be obtained over a wide range of rates and lengths. The number of sub-matrices could also be varied unlike in most previous methods. The method was initially proposed in [11][12] as a general method for constructing QC-LPDC codes by random searching. In [11][12] a distance graph is constructed that is then converted into a parity-check matrix. However, girths higher than eight could not be guaranteed for codes with column-weight of two and girths higher than six for codes with column-weights higher than two. We add conditions to the initial algorithm in [11][12] to guarantee higher girths. We also construct a Tanner graph instead of a distance graph. Although distance graphs are more compact than a Tanner graph, the algorithm complexity is higher than when a Tanner graph is constructed. Our algorithm works in a similar way to the methods in [4][13][14] in that we set the conditions to satisfy a given girth with given row and

column weights then search for connections that satisfy these conditions. The advantage of our algorithm is that it works for any sub-matrix configuration or base matrix and could be used to obtain girths larger than 12.

This article is structured as follows. Section 2 presents the proposed algorithm for constructing flexible and high-girth QC-LDPC codes. Conditions for avoiding small cycles in a Tanner graph are introduced and illustrated. Bit-error rate performance analysis of obtained codes is presented in Section 3. Section 4 has concluding remarks.

2. PROPOSED ALGORITHM

Random or pseudo-random construction algorithms such as bit-filling (BF) [15] and progressive-edge growth (PEG) [16] have been developed to construct a wide range of codes. These algorithms construct a LDPC code by connecting rows and columns of a code one at a time provided a targeted girth and or rate is not violated. We take advantage of the flexibility found in random search methods such as BF and PEG to construct a wide range of QC-LDPC codes. To obtain QC-LDPC codes we add some modifications to these random search algorithms.

Our proposed algorithm has the following four main steps:

1) Divide rows (check nodes) and columns (variable nodes) of the constructed code into *j* (column weight) and *k* (row-weight) or more equal size groups respectively. The division of rows and columns into groups creates sub-matrices of the code.

2) The row-groups (RG) and column-groups (CG) are then paired (connections) such that each row-group appears *k* times and each column-group *j* times. The number of each row-group or column-group appearances (connections) determines the rate of the code. If the number of group appearances varies an irregular code is obtained. Row and column group connections form a base matrix or protograph.

3) For each row-column group (RCG) pair select a row, *i*, in the row-group, and search for a column, *x*, in the column-group that is at a desired distance (shortest path between nodes) from row *i*. Connect rows in the row-group to columns in the column-group according to the connection of row *i* and column *x*. That is, if row *i* is connected to column *x*, then row *i+a* is connected to column *x+a*. The connections are modulo of the size of row and column groups, p. These connections create a cyclic shift in the sub-matrices (shifted identity sub-matrices) of the constructed code.

4) Use the obtained Tanner graph to form a parity check matrix.

A (12,2,3) QC-LDPC code is constructed in **Figure 1** to illustrate the steps of the algorithm above. The number of row-groups is two (column-weight) and three (row-weight) for column-groups. Each group is of size four. Row-groups are denoted as RG and column-groups as CG. The groups are paired as[RG1,CG1],[RG1,CG2], [RG1,CG3],[RG2,CG1],[RG2,CG2] and [RG2,CG3] such that each row-group appears three (*k*) times and each column-group two (*j*) times. In each Row-Column group (RCG) connection a specified girth condition is observed. In

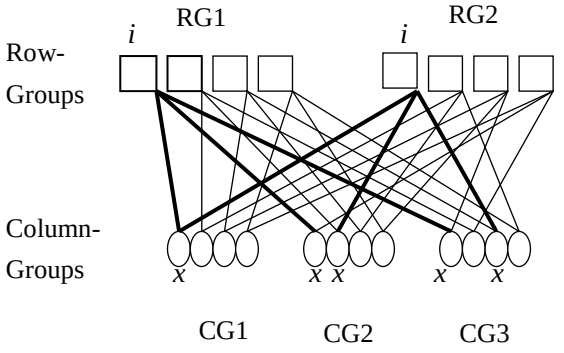

Figure 1. Construction of a QC-LDPC Code Tanner graph

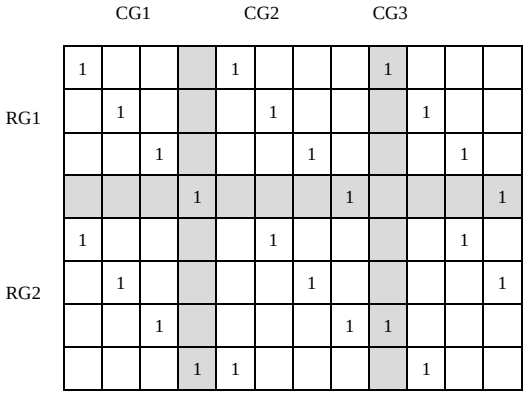

Figure 2. A (12,2,3) QC-LDPC code in matrix form

I_s	I_s	I_s	I_s
I_s	I_s	I_s	I_s
I_s		I_s	I_s

I_s	0	I_s	0	I_s	0	I_s	0
0	I_s	0	I_s	0	I_s	0	I_s
I_s	I_s	0	0	0	I_s	I_s	0
I_s	0	I_s	0	I_s	0	0	I_s
0	0	I_s	I_s	0	I_s	I_s	0
0	I_s	0	I_s	I_s	0	0	I_s

Figure 3. Structure of obtained quasi-cyclic LDPC codes

this case four-cycles are avoided. Reference row *i* in RG1 is connected to the first columns labelled *x* in all the column groups and the connections are in bold. In each case the rest of the connections in RG1 follow that of row *i* and column *x*. In RG2 the reference row *i* is also connected to the same column *x* in CG1 and to the second and third columns labelled *x* in CG2 and CG3 respectively. The reference row *i* connects to columns *x* that do not form 4-cycles. The connections of RG2 are shifted by 1 and 2 in CG2 and CG3 respectively. The search for column *x* could be done sequentially or randomly. In sequential searches, a column-group is searched from beginning to end in sequential order. In random searches, a column is chosen arbitrary in a column group.

Figure 2 shows the matrix representation of the row-column connections of the code in **Figure 1**. The top row sub-matrices represent RG1 connections and the bottom sub-matrices RG2 connections. Column sub-matrices represent the three column-groups connections. When the number of row-groups and column-groups is equal to *j* and *k* respectively a code structure such as that of **Figure 3 (a)** is obtained where *Is* is a shifted *p x p* identity sub-matrix. When more than *j* row-groups or *k* column-groups are used, codes with zero sub-matrices are obtained as the example in **Figure 3 (b)** illustrates, where *O* is a *p x p* zero sub-matrix.

2.1 Avoiding Four-Cycles

In the four steps of the proposed algorithm outlined above, smaller cycles are avoided by connecting the reference row to a column that satisfies a desired distance. This condition guarantees girth of six if a column satisfying the distance of at least four is found. If the reference row *i* does not form a 4-cycle then all other rows in the same group do not form a 4-cycle. **Figure 4 (a)** shows how four-cycles are avoided to obtain girth of six. Column *x* must be at least a distance of four from row *i*. Since the connections of row *b* follow those of *i*, row *b* also avoids formation of 4-cycles. For row $i+a$ (or row *b*) to form a 4-cycle, $x+a = z+a$ in **Figure 4(a)**. That is, $x=z$. Since *i* is not connected to *z*, $(z!=x)$, then $i+a$ also avoids 4-cycles. Also if *i* forms a six-cycle with *x*, *b* cannot not form a four-cycle with *c*. That would happen if $i+a = b$ and $x+a=c$. That is, *x* is at a distance of 3 from *i* which should not be the case since we are searching for *x* with a distance of at least 4.

To obtain girth of eight, six-cycles are avoided. The girth-condition in the proposed algorithm is also sufficient for girth of eight. That is, if reference row *i* does not form less than eight-cycles then the rest of the rows in the group also do not form cycles less than eight. **Figure 4 (b)** shows how six-cycles are avoided by connecting rows and columns that are apart by at least six. Row *i* avoids connections to columns that are within a distance of six from itself. However, there is a possibility that smaller cycles could be formed when the rest of rows in the group are connected according to the reference row *i*. In **Figure 4(b)** column *z* is not connected to row *i* because it does not meet the distance condition of at least six. There may be a column *c* at a distance of three from row *i* and row *b* in the same group with *i* that is at a distance of three from column *x* (*i*,*b* and *c*,*x* in the same groups) as in **Figure 4(c).** If the difference (as indicated by arrows in the Figure) from row *i* to row *b* is the same as that from column *c* to *x* then row *b* will be connected to *c*. With *i* already connected to *x* an eight-cycle is formed between *i*, *x*, *b* and *c* as shown in **Figure 4(c).** These kinds of connections may result in formation of smaller than desired cycles. In this case, the smallest possible cycle formed is eight. Therefore, girth-eight is guaranteed if row *i* and column *x* do not form a six-cycle. Smaller cycles are formed by nodes connection of nodes on both sides. The smallest path length between these nodes is 3 and when connected they form a path of 8.

2.2 Avoiding Eight-Cycles

To obtain girth-ten, eight cycles are avoided. The girth condition in the proposed algorithm does not guarantee girth of ten. That is, smaller cycles could be obtained even if the cycle between row *i* and column *x* is at least ten. Smaller cycles could be formed as in **Figure 4 (c).** Although, row *i* and column *x* do not form an eight-cycle, if they are connected, they may map row *b* onto column *c* which leads to an eight-cycle between rows *i*, *b* and columns *x*, *c* as described earlier. These connections are made when the difference from row *i* to row *b* is the same as the difference from column *x* to column *c*. The differences are modulo of row and column groups' size, *p*. To avoid, formation of eight-cycles when targeting ten-cycles, the difference between row *i* and *b* must not be the same as that of between columns *x* and *c* given that the distance from *i* to *c* and *b* to *c* is 3. This is in addition to the condition that column *x* must be at the distance of at least eight from row *i* before they are connected.

2.3 Avoiding Ten-Cycles

To obtain girth-twelve in the Tanner graph ten-cycles are avoided. The reference row *i* must be connected to column *x* that is at a distance of at least ten. As in the case of girth-ten, smaller cycles could be formed when the rest of rows and columns are connected according to the connection of reference row *i* and column *x*. **Figure 5** shows the conditions that lead to formation of ten-cycles even when row *i* and column *x* do not form less than twelve cycles. Rows *i* and *b* are in the same group and so are columns *x* and *c*. If row *b* is at a distance of 4 from row *i* and column *c* is also at a distance of 4 from column *x* then the connection *i* to *x* and *b* to *c* forms a ten-cycle as in part (a) of **Figure 5**. In part (b) of **Figure 5** row *b* is at a distance of 3 from *x* and column *c* at a distance of 5 from *i*. In part (c) *b* is at a distance of 5 from *x* and column *c* is at a distance of 3 from *i*. If connecting *i* to *x* forces *b* to be connected to *c* then a ten-cycle is formed in each case. Therefore to avoid formation of ten-cycles, the conditions

in **Figure 5** are avoided by ensuring that connecting *i* and *x* does not lead to a connection of *b* and *c* with the shown distances. The avoidance of ten-cycles is simply implemented by ensuring that the difference from row *i* to row *b* is not the same as that of from column *x* to column *c*. Eight cycles are avoided as described in subsection 2 above.

2.4 Avoiding Twelve-Cycles

To obtain girths higher than twelve, twelve-cycles are avoided. Twelve-cycles are avoided in a similar way to eight and ten-cycles. However, to obtain girths higher than twelve the base matrix must satisfy certain conditions as derived in [17]. Conditions for girths of 16, 18 and 20 are also derived for the base matrix.

Figure 6 shows conditions than lead to formation of twelve-cycles even when row *i* and column *x* do not form cycles less than fourteen. In part (a) of the Figure column *c* is at a distance of 5 from row *i* and *x* is also at a distance of 5 from *b*. If the difference from row *i* to *b* is the same as that from column *x* to *c* then *b* is connected to *c* which forms a twelve-cycle with *i* and *x*. A twelve-cycle is also formed when *b* is at a distance of 6 from *i* and *x* at a distance of 4 from *b* as in part (b) of the Figure. In part (c) row *b* is at a distance of 4 from *i* and column *c* at a distance of 6 from *x*. In part (d) row *c* is at a distance of 3 from row *i* and column *x* at a distance of 7 from *b*. Twelve-cycles are also formed when *c* is at a distance of 7 from *i* and *x* is at a distance of 3 from *b* as in part (e) of **Figure 6**. To obtain girth of fourteen these conditions are avoided.

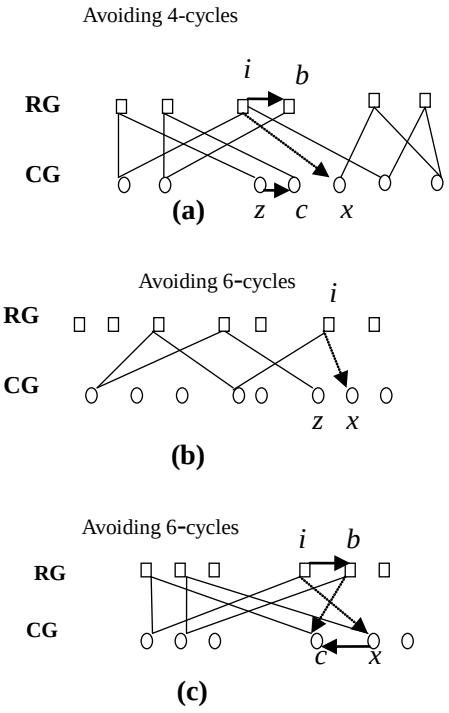

Figure 4. Avoiding 4- and 6-cycles in Tanner graphs

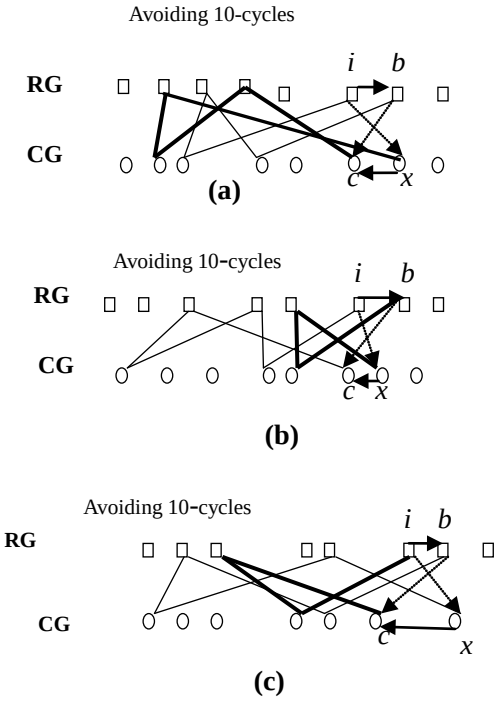

Figure 5. Avoiding ten-cycles in Tanner graphs

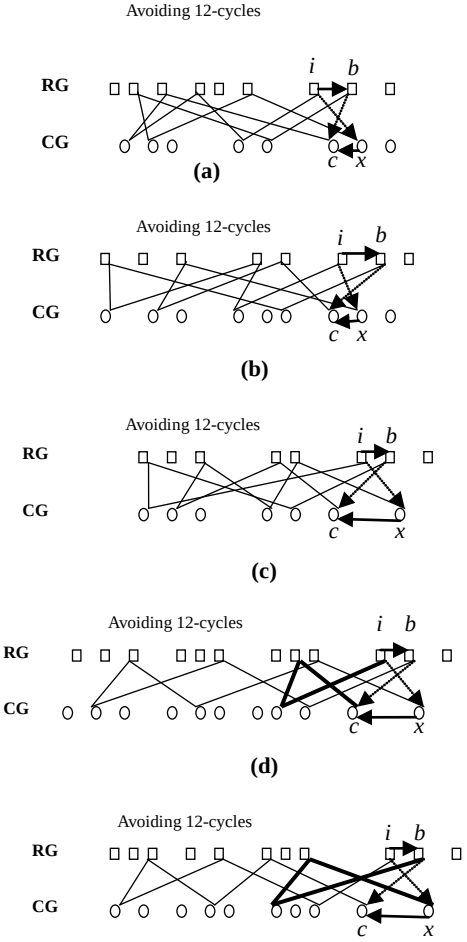

Figure 6. Avoiding twelve-cycles in a Tanner graph

2.5 Algorithm Analysis

The algorithm's conditions are that row *i* and *x* satisfy the desired girth and there are no connections of *b* and *c* that result in small cycles. The difference from *i* to *b* must not be equal to that of from *x* to *c* so that smaller cycles are not formed by *b* and *c*. Row-Column connections could be made either from the row or column side in the proposed algorithm. The algorithm is presented formally below with connections made from the row side.

QC-LDPC Code Search Algorithm

1) Divide code rows into *j'* equal groups of size *p*, (RG_1...$RG_{j'}$) and columns into *k'* groups of size *p*, (CG_1...CG_k'), where $j' \geq j$ and $k' \geq k$. k is code row-weight and *j* is column-weight. r_x is row *x*. U_{rx} is a set of rows and columns within a desired distance from row r_x. c_x is column *x*. U_{cx} is a set of rows and columns within a desired distance from column *cx*. Distance is the shortest path between any two nodes (rows or columns).

2) Make row-column group (RCG) pairings according to your specifications such that each row-group appears *k* times and each column-group *j* times for regular codes. The number of such group pairings is *j'k* or *k'j* for regular codes. The row-column groups are (RCG_1 ...$RCG_{kj'}$).

3) *For t = 1 to kj'* {
Select r_i from RG in RCG_t, where $1 \leq i \geq p$.
Sequentially or randomly search for c_x from CG in RCGt, where c_x notin Uri and avoid smaller than desired cycles by using the difference condition. Else the algorithm fails.
for z = 0 to p-1 {
r_{i+z} is connected to c_{x+z}. }}

4) Use the obtained Tanner graph to form a parity-check matrix.

One of the advantages of this algorithm compared to those in [11] [12][14] is that there is some pattern in the girth conditions. For example, avoiding eight-cycles the algorithm checks distances (path lengths) of 3 from row *i* and column *x* as in **Figure 4(c).** To avoid ten-cycles, rows and columns at distances of 3-5, 4-4, and 5-3 as in **Figure 5** are searched. For twelve-cycles the distances are 3-7, 4-6, 5-5, 6-4, and 7-3 as in **Figure 6**. The two distances add up to the avoided cycle length minus two and the minimum path length is three. Minus two because of the two connections from *i* to *x* and *a* to *b* are not yet counted. The algorithm works in a similar way regardless of the number and connections of row and column groups. The code rate is simply varied by changing the column and rows weights accordingly. The code length is varied by changing the size of sub-matrices, *p*.

The computational complexity of the algorithm is analysed and estimated in terms of the number of rows, *M*, and the number of row groups as follows.

- There are *j'k* or *k'j* row-column groups pairings for regular codes and each group is of size *M/j'*.
- Searching for a column satisfying the distance in each group takes *M/j'* operations. Testing for the difference conditions also takes about *M/j'* operations. Therefore each row-column group pairing takes about *2M/j'* operations.

o Since there are *kj'* row-column group pairings for a regular code it takes *2j'kM/j'* operations to complete all searches and connections. This is assuming the algorithm does not fail. The complexity of this algorithm is therefore O(M). However, this is a search algorithm and may fail many times even when it is possible to construct a code with some given parameters. Therefore, practically the algorithm may need many tries to obtain a code with given code parameters.

As shown by the complexity analysis above, the proposed algorithm computational complexity is linear with the number of rows, *M*. Although each execution of the algorithm does not guarantee that we will get a code (that is, algorithm may fail in step 3), our experiments show that the algorithm constructs a code most of the time even for small groups sizes with girths less than ten as shown by our MATLAB code in [18]. We easily obtained codes with minimum group sizes [4] for girth-six and eight codes. However, for minimum sizes of sub-matrices and higher girths (larger than eight) the algorithm may take a long time to find a code with sub-matrix sizes closer to minimum sizes.

2.6 Base Matrices with Zero Sub-Matrices

Codes with Zero Sub-Matrices: Matrices with zero sub-matrices are designed by using row and column groups larger than row and column weights. The number and interconnection of row-column groups may be dictated by decoding performance, hardware architecture or just be random. Codes with zero sub-matrices could be regular or irregular. In [19][20] protographs are used to design codes that match the structure of a semi-parallel architecture. A protograph is a small bipartite graph from which a larger graph can be obtained by a copy-and-permute procedure. The protograph is copied a number of times and then edges of individual replicas are permuted to obtain a single, large graph.

In [19] the protograph is designed to resemble decoder architecture. The decoder architecture may first be chosen based on a protograph decoding performance. Random or simulated annealing techniques were used to find the best performing protograph. Performance of the code also depends on the size of sub-matrices and their shift values. The matrix is expanded with *p x p* shifted identity sub-matrices. We could use the proposed algorithm to improve performance of constructed codes by improving their girths. The advantage of the proposed algorithm is that it can be used to construct codes with any sub-matrix or protograph configuration.

Irregular Codes: Carefully constructed irregular codes could have better performance compared to regular codes [21]. It was also shown in [22] that the degree of a variable node plays an important role in determining its error correcting performance. Variable nodes with a high degree tend to be

decoded correctly compared to others. By targeting message bits to have higher degrees compared to parity bits we can improve the performance of a parity-check code. In [23] irregular quasi-cyclic codes from difference sets were found to have slightly better performance compared to regular ones. As stated in our algorithm, if the number of row or column groups is not uniform we obtain irregular codes. We can therefore construct irregular codes by having different numbers of appearances for row or column groups or both. A group weight is equal to the number of times it appears in connections. The proposed algorithm could be used to further improve performance of irregular codes by increasing their girths.

3. Performances Simulations

Bit error rate (BER) performances of constructed codes were simulated on an AWGN channel with BPSK modulation. Performance curves are shown in **Figures 7** and 8. **Figure 7** shows performance curves for (N,3,6) codes. Two new girth-ten (1998,3,6) QC-LDPC codes are compared against each other and a combinatorial QC-LDPC code [6]. One of the new QC-LDPC code is obtained with random selections while in the other code the searches for nodes satisfying the desired girth were sequential. Obtained QC-LDPC code using random selection performs better than the sequential QC-LPDC code. The results confirm what was found in [11][12] that sequential QC-LDPC codes perform worse that randomly constructed codes at girths of 6 and 8. However, in the case of girth-ten codes the performance gap is less than in girth 6 and 8 codes. This may be due to less regularity (pattern) in connections in higher girths compared to small-girths codes. The random QC-LDPC code also outperforms the QC-LDPC code from [6] and a girth-six random code of the same size at the shown simulation range. Also shown in the Figure is a (1008,3,6) code that slightly outperforms a code of the same size and girth obtained using the method in [9]. The new code performance better by about 0.05db at BER of 10-6.

Figure 8 shows performance of a new girth-14 (5776,3,4) code compared to a girth-12 code of the same size and rate from [7]. Our girth-14 code is outperformed by the girth-12 code. Also shown are high-rate QC-LDPC code performance curves. Our (3976,4,28) QC-LDPC code is slightly outperformed by a BIBD code[8] of the same rate and length in the shown range. At a rate 0.75 and code length of 4008, our code performs as well as a QC-LDPC code [4] of the same rate and length.

Obtained codes using our proposed algorithm perform as well as codes from other construction methods in shown simulation ranges. They do not show any performance superiority over other codes of the same size, length and girth. In some cases they are outperformed by codes with lower girths. It has to be noted that the structure of a LDPC code also plays an important role in its performance. In [4] the author proposed one way of searching for codes with better minimum distance. Other techniques such as avoidance of stopping sets [24] may be used to further improve performance of obtained codes. These techniques could be used in our proposed algorithm in future to search for better performing codes.

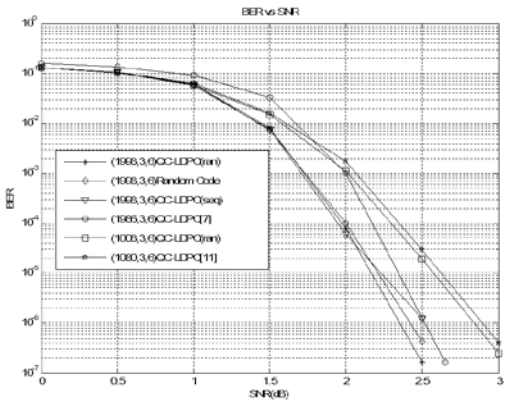

Figure 7. BER performance of regular (N,3,6) QC-LDPC Codes

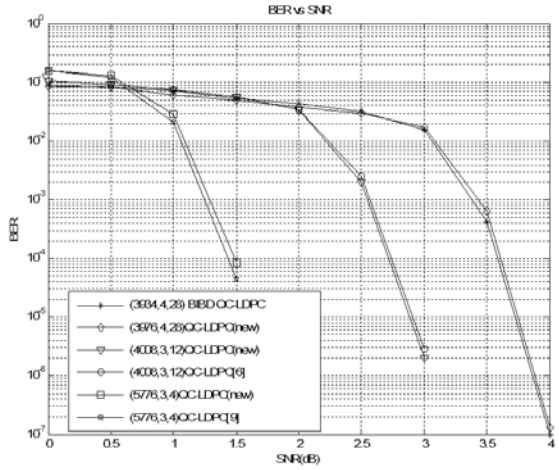

Figure 8. BER performance of high rate and girth QC-LDPC codes

4. Conclusions

A non-algebraic search algorithm for constructing high-girth quasi-cyclic LDPC codes has been presented. The algorithm is flexible in that, a wide range of rates and lengths can easily be obtained for both regular and irregular codes. The number and configuration of sub-matrices can also be changed to include zero sub-matrices. The algorithm is also simple to modify from one code design to another. For a given girth one needs to only change the group connections and size to obtain different code parameters. The algorithm offers more flexibility compared to previous developed QC-LDPC construction algorithms. Obtained codes show good BER performances comparable to random codes. Although obtained codes do not show BER performance superiority over other codes, they offer better flexibility in design.

REFERENCES

[1] S. Olcer, "Decoding Architecture for Array-code-Based LDPC Codes," Proc. IEEE GLOBECOM, pp. 2046- 2050, December 2003.

[2] L. Chen, J. Xu, I. Djurdjevic, and S. Lin, "Near Shannon-Limit Quasi-Cyclic Low-Density Parity-Check Codes," IEEE Transactions on Communications, Vol. 52, pp. 1038-1042, July 2004.

[3] M. O' Sullivan,J. Brevik and R. Wolski, "The Performance of LDPC Codes with Large Girth," Proc. of the 43rd Annual Allerton Conference; Communication, Control and Computing, September 2005.

[4] M. P. Fossorier, "Quasi-Cyclic Low-Density Parity-Check Codes From Circulant Permutation Matrices," IEEE Transactions on Information Theory, Vol. 50, pp.1788-1793, August 2004.

[5] M. O' Sullivan, "Algebraic Construction of Sparse Matrices with Large Girth," IEEE Transactions on Information Theory, Vol.52, pp. 718-727, February 2006.

[6] Y. Kou, S. Lin and M.P.C Fossorier ," Low Density Parity Check Codes Based on Finite Geometries: A Rediscovery and New Results," IEEE Transactions on Information Theory, Vol. 47, No. 7, pp. 2711-2736, November 2001.

[7] L. Jing, J. Lin and W. Zhu, "Design of Quasi-Cyclic Low-Density Parity-Check Codes with Large Girth" ETRI Journal, Vol. 29, No. 3,pp.381-389 June 2007.

[8] B. Ammar, B. Honary, Y. Kou, J. Xu and S.Lin,"Construction of Low-Density Parity-Check Codes Based on Balanced Incomplete Block Designs", IEEE Transactions on Information Theory, Vol. 50, No. 8, pp.1257-1268, June 2004.

[9] J. Fan, Y. Xiao and K. Kim, "Design LDPC Codes without Cycles of Length 4 and 6", Research Letters in Communications, Vol. 2008, Article ID 354137.

[10] S. Kim,J. No, H. Chung and D. Shin, "Construction of Protographs for QC-LDPC Codes With Girth Larger Than 12," IEEE Transactions on Information Theory, Vol. 53, No. 8, pp.2885-2891.

[11] G. Malema, "Constructing Quasi-Cyclic LDPC Codes Using a Search Algorithm," International Symposium on Signal Processing and Information Technology, Cairo Egypt, pp.969-973, December 2007.

[12] G. Malema, "Low-Density Parity-Check Codes: Construction and Implementation," PhD Thesis, Electrical and Electronic Engineering, The University of Adelaide, Adelaide, Australia, 2007.

[13] Y. Wang, J. Yedidia and S. Draper, "Construction of High-Girth QC-LDPC Codes,"5th International Symposium on Turbo Codes and Related Topics, pp.180-185,Lausanne,Switzerland,September 2008.

[14] Z. Li , and BV.K. Vijaya Kumar, "A class of Good Quasi-Cyclic Low-Density Parity-Check Codes Based on Progressive Edge Growth Graph," Proceedings of 38th Asilomar Conference Conference on signal, systems computing, pp. 1990 –1994, 2004.

[15] J. Campello, D.S. Dodha, and S. Rajagopalan, "Designing LDPC codes using Bit-Filling," Proceedings of the International Conference on Communications, Vol.1, pp. 55-59, Helsinki, Finland, June 2001.

[16] X. Hu, E. Eleftheriou, and D. Arnold, "Progressive edge-growth Tanner Graphs," Proc. IEEE GLOBECOM, Vol. 2, pp. 995-1001, San Antonio, TX, November 2004.

[17] S.Kim, J. No, H. Chung, and D. Shin, "Quasi-Cyclic Low-Density Parity-Check Codes with girths larger than 12," IEEE Transactions on Information Theory, vol. 53, no. 8, pp. 2885- 2891,August 2007.

[18] G. Malema, "Construction of QC-LDPC Codes -qc_ldpc.m",Mathworks http://www.mathworks.com/matlabcentral/fileexchange/34176.

[19] J.K. S Lee, B. Lee, J. Hamkins, J. Thorpe, K. Andrews, and S. Dolinar, "A Scalable Architecture for Structured LDPC Decoder," IEEE International Symposium on Information Theory, pp.292-295, Chicago, IL, July 2004.

[20] K. Andrews, S. Dolinar, D. Divsalar, and J. Thorpe, "Design of Low-Density Parity-Check Codes for Deep Space Applications," IPN Progress Report, pp.42-159, November 2004.

[21] M. Luby, M. Mitzenmachen, M. Shokrollahi, and D. Spielman, "Efficient Improved Low-Density Parity-Check Codes using Irregular Graphs," IEEE Transactions on Information Theory, Vol. 47, pp.585- 598, 2001.

[22] D. MacKay, S. Wilson, and M. Davey, "Comparison of Constructions of Gallager Codes," IEEE Transactions on Communications, Vol. 47, pp.1449-1454, October 1999.

[23] S. Johnson and S. Weller, "A Family of Irregular LDPC Codes with Low Encoding Complexity,"IEEE Communications Letters, Vol. 7, No. 2, pp.79-81, February 2003.

[24] G. Ritcher and A. Hof, "On a construction method of irregular LDPC codes without small stopping sets," IEEE International Conference on Communications, Vol. 3, pp. 1119-1124, Istanbul, Turkey, June 2006.

International Journal of Computer Science and Application
IJCSA Volume 1, Issue 1, August 2012 PP. 26-30
ISSN(online)
ISSN(print)
www.ij-csa.org

Effect of Tcp Reno Without Maximum Congestion Window in Ring Noc

Mohammad Reza Nouri Rad[1], Teamour Esmaeili[2]
[1 2]Dep.of Computer Engineering DareShahr Branch, Islamic Azad University, Iran
Email: [1]nouri_rad@yahoo.com, [2]esmaeiliteamour@yahoo.com

(Abstract) This paper shows the behavior of TCP Reno with two packet drops without maximum congestion window for increase the reliability of network on chip (NoC).We simulate Ring NoC architecture with Network Simulator 2 (NS2). The simulation results reveal the applicability of TCP Reno protocol in Congestion Control in proposed architecture.
Keywords: Network-on-Chip; TCP Reno; Congestion Window; Ns-2.

1. INTRODUCTION

Packet-based interconnection networks, known as Network-on-Chip (NoC) architectures, are increasingly adopted in System-on-Chip (SoC) designs, which support numerous homogeneous and heterogeneous functional modules. Systems-on-chip (SoCs) for multimedia or telecommunication applications will contain a large number of process-ing elements (PEs) such as a DSP processor, RISC CPU, embedded RAM, graphics engine, etc. As a result, there is a need for high-throughput communications links between these blocks. There exist many bus based SoCs which are widely used in industry such as AMBA [1].

Transmission Control Protocol (TCP) is one of the core protocols of the TCP/IP Protocol Suite. TCP is used to provide reliable data between two nodes and works at the transport layer of the TCP/IP model. TCP operates at a higher level, concerned only with the two end systems, for example, a Web browser and a Web server. In particular, TCP provides reliable, ordered delivery of a stream of bytes from a program on one computer to another program on another computer. Besides the Web, other common applications of TCP include e-mail and file transfer. Among its other management tasks, TCP controls message size, the rate at which messages are exchanged, and network traffic congestion [2].

Commonly used TCP variant is TCP Reno and uses basic AIMD mechanism only to adjust their congestion window size. TCP Reno was the modified version of TCP Tahoe. These protocols are not scalable as the delay-bandwidth product of the network becomes larger [3] because additive increase is too slow and multiple decrease is too fast. Basic TCP uses packet loss only to adjust the congestion window size. So, TCP Vegas and FAST TCP are proposed to cope up the same problem. FAST TCP uses packet loss as well as queuing delay as the congestion control parameter and to adjust window after every RTT (Round Trip Time) [4-7].

TCP/IP model is made up of 4 layers i.e. Application layer, Transport layer, Internet layer, Network layer. Transmission Control Protocol (TCP) is one of the main protocols used at the transport layer. The TCP/IP header is shown as below:

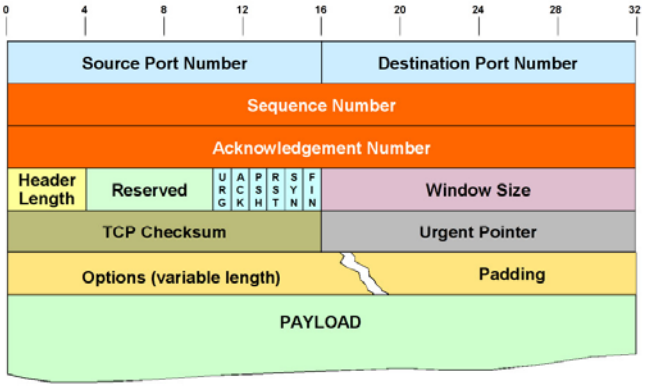

Fig. 1. the TCP header.

Congestion is the phenomenon that occurs at a router when incoming packets arrive at a rate faster than the router can switch (or forward) them to an outgoing link. However, it is important to distinguish contention and congestion. “Contention occurs when multiple packets have to be queued at a switch (or a router) because they are competing for the same output link, whereas congestion means that the switch has so many packets queued that it runs out of buffer space and has to start dropping packets”.

Congestive collapse (or congestion collapse) is a condition which a packet switched computer network can reach, when little or no useful communication is happening due to congestion. Congestion collapse generally occurs at choke points in the network, where the total incoming traffic to a node exceeds the outgoing bandwidth.

2. CLASSIFICATION OF CONGESTION CONTROL ALGORITHMS

The classification of congestion control algorithms in TCP was demonstrate as below:

2.1 Slow Start Algorithm

The Slow Start Algorithm tries to avoid congestion by sending data packets defensively. Therefore, two special variables named congestion window (cwnd) and Slow Start threshold (ssthresh) are stored on sender's side.

Initially, cwnd is sized to one packet when the sender injects a new packet into the network and waits for the Acknowledgment (ACK) [5]. from the receiver. Normally, this packet gets through the network and reaches the recipient in time, so it will be replied by an ACK. If this acknowledgment is received by the sender, cwnd is incremented; if network capacity is reached and packets get lost, the sender does not increment the number of packets any further. That means, by each sending cycle the number of injected data packets is doubled until network's capacity is reached and the required ACK cannot get through. the Slow-Start and Congestion Avoidance was shown s below:

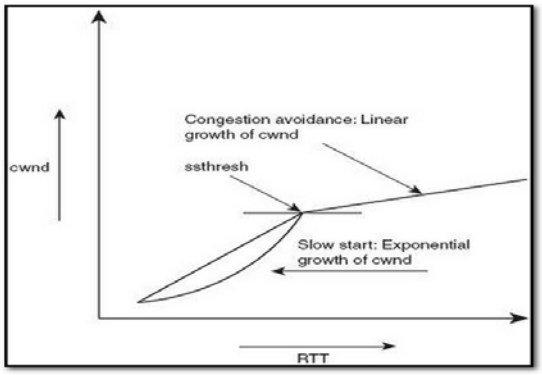

Fig. 2. Slow-Start and Congestion Avoidance.

2.2 Fast Retransmit Algorithm

Fast Retransmit Algorithm uses explicit feedback methods to avoid long timeout periods waiting for packet retransmitting in case of packet loss. Such problems are inherent in packet-switched data networks because every data packet can travel individually through the rest of the network and can use special routes from the sender to the recipient [8]. Consequently, the transmitted data packets will neither reach the recipient in accurate order nor complete continually. Therefore, after detecting a missing packet the recipient sends duplicated ACK packets for the last correct received packet until the missing packet receives. Unfortunately, TCP may use duplicate ACK packets to indicate out-of-order-packets, thus two ACK packets do not necessarily indicate a lost packet. Therefore, if a sender receives multiple ACK packets with the same sequence number, normally at least three of them, these packets indicate the last successfully transmitted packet. the Retransmit Algorithm was shown as below:

2.3 Fast Recovery Algorithm

A special Congestion Avoidance Algorithm often combined with Fast Retransmit to restart transmission at a higher throughput. rate than Slow Start is the FAST Recovery Algorithm. Fast Recovery starts when Fast Retransmit fails to work. If no further duplicate ACK packets are received for Fast Retransmit Algorithm the sender tries to return to normal sending state.

2.4 Congestion Avoidance

Congestion can occur when data arrives on a big pipe (a fast LAN) and gets sent out a smaller pipe (a slower WAN). Congestion can also occur when multiple input streams arrive at a router whose output capacity is less than the sum of the inputs. Congestion avoidance is a way to deal with lost packets [9]. The assumption of the algorithm is that packet loss caused by damage is very small (much less than 1%), therefore the loss of a packet signals congestion somewhere in the network between the source and destination. There are two indications of packet loss:

A timeout occurring.

Receipt of duplicate ACKs.

Congestion avoidance and slow start are independent algorithms with different objectives. But when congestion occurs TCP must slow down its transmission rate of packets into the network, and then invoke slow start to get things going again. In practice they are implemented together. Congestion avoidance and slow start require that two variables be maintained for each connection: a congestion window(cwnd), and a slow start threshold size(ssthresh).

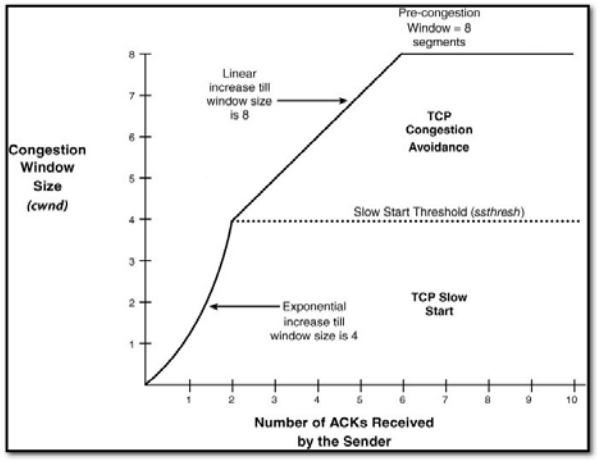

Fig. 3. Fast Retransmit Algorithm.

3. TYPES OF TCP PROTOCOLS

3.1 Packet Loss-based Protocols

These are the protocols which uses packet drop probability as the main factor for adjusting the window size. These variants of TCP use congestion control algorithms. There were developed initially and are still used. Loss based TCP protocols are more aggressive than the delay based TCP protocols [6]. These are classified as below:

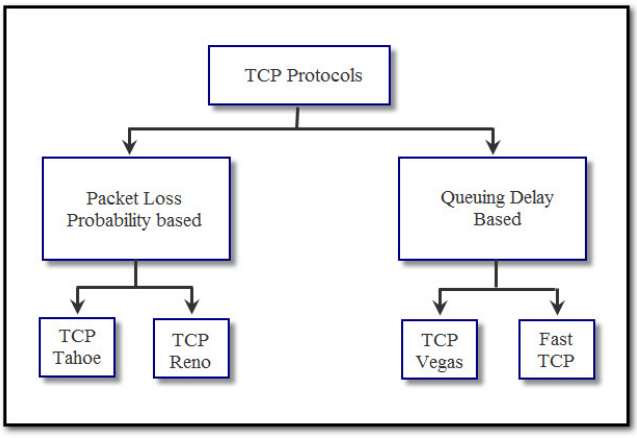

Fig. 4. The Types of TCP Protocols.

1) TCP Tahoe

TCP Tahoe is the TCP variant developed by Jacobson in 1988 [10]. It uses Additive Increase Multiplicative Decrease (AIMD) algorithm to adjust window size. It means that increases the congestion window by one for successful packet delivery and reduces the window to half of its actual size in case of data loss or any delay only when it receives the first negative acknowledge. In case of timeout event, it reduces congestion window to 1 maximum segment size (MSS) [11]. TCP Tahoe has following specification:

• TCP Tahoe uses packet loss probability to adjust the congestion window size.

• During Slow Start stage, TCP Tahoe increases window size exponentially i.e. for every acknowledgement received, it sends two packets.

• During Congestion Avoidance, it increases the window size by one packet per Round Trip Time (RTT) so as to avoid congestion.

• In case of packet loss, it reduces the window size to one and enters in Slow Start stage.

2) TCP Reno

This Reno retains the basic principle of Tahoe, such as slow starts and the coarse grain re-transmit timer. However it adds some intelligence over it so that lost packets are detected earlier and the pipeline is not emptied every time a packet is lost. Reno requires that we receive immediate acknowledgement whenever a segment is received. The logic behind this is that whenever we receive a duplicate acknowledgment, then his duplicate acknowledgment could have been received if the next segment in sequence expected, has been delayed in the network and the segments reached there out of order or else that the packet is lost. If we receive a number of duplicate acknowledgements then that means that sufficient time has passed and even if the segment had taken a longer path, it should have gotten to the receiver by now. There is a very high probability that it was lost so Reno suggests an algorithm called 'Fast Re-Transmit'. Whenever we receive 3 duplicate ACK's we take it as a sign that the segment was lost, so we re-transmit the segment without waiting for timeout. Thus we manage to re-transmit the segment with the pipe almost full [12].

3.2 Delay-Based TCP Protocols

Delay-based algorithms were developed so as to provide stable throughput at the receiver end. These TCP variants use congestion avoidance algorithms to avoid the packet loss and are less aggressive than packet loss based TCP protocols. Delay-based algorithms can maintain a constant window size avoiding the oscillations inherent in loss-based algorithms [13].These are classified as below:

3) TCP Vegas

TCP Vegas is a congestion control or network congestion avoidance algorithm that emphasizes packet delay, rather than packet loss, to determine the rate at which to send packets. TCP Vegas detects congestion during every stage based on increasing Round Trip Time (RTT) values of the packets in the connection unlike Reno, Tahoe etc. which detect congestion only after it has actually happened via packet drops [14].

• TCP Vegas adjusts the source rate before actually packet is dropped.

• Queuing delay is the difference between base RTT and avg RTT.

• TCP Vegas decreases the source rate in case of increase in queuing delay value and increases in case of decrease in queuing delay.

4) FAST TCP

FAST TCP is a new TCP congestion control algorithm for high-speed long-distance networks; it aims to rapidly stabilize networks into steady, efficient and fair operating points. It uses queuing delay, in addition to packet loss, as a congestion signal. Queuing delay provides a finer measure of congestion and scales more naturally with network capacity than packet loss probability does [15]. Using the queuing delay as a congestion measure in its window updating equation [4], allows FAST TCP to overcome difficulties [16] encountered by currently used algorithms (such as TCP Reno [12]) in networks with large bandwidth-delay products.

4. SYSTEM ARCHITECTURE

4.1 Hardware Architectures

The common characteristic of NoC architectures is that the constituent IP cores communicate with each other through switches [17]. In network-on-chip the bandwidth between resource (IPs) and switches is very higher than the bandwidth between switch to switch. The ring NoC has shown as below:

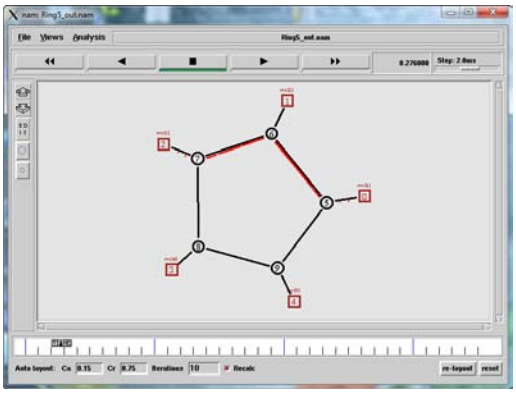

Fig. 5. The Ring Network on Chip.

As shown in fig. 5, The Square was processor elements and the circle elements were switches that connect to each other.

5. EVALUATIONS

5.1 Simulation Framework

In this paper, we have modeled our NoC architecture concepts with the widely used network simulator ns-2 [18]. NS-2 has been widely applied in research related to the design and evaluation of computer networks and to evaluate various design options for NoC architectures [19], including the design of routers, communication protocols, etc.

We consider that two Res11 and Res44 communicate with each other through network. the bandwidth between resources and switches was equals to 8 megabits/sec and the bandwidth between switches was equals to 800 kilobit/sec. the link delay between resources and switches was equals to 10 milliseconds and the link delay between switches was equals to 15 milliseconds. The ssthresh is set two packets.

5.2 Fast Retransmission

As shown in fig. 6, TCP Reno cannot recover from multiple losses in one window size without a timeout. The first packet loss can be recovered by fast retransmission. The second packet loss cannot get enough duplicate acknowledgments for fast retransmission and has to wait for retransmission timeout to recover from loosing.

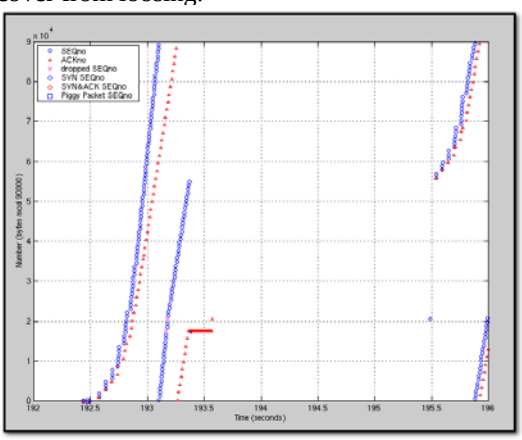

Fig. 6. Sequence Number of TCP Reno in switch.

5.3 Congestion Window

As shown in fig. 7, Congestion window is the flow control set by the traffic source. Advertised window is the flow control performed by the TCP sink. Thus maximum congestion window has the same effect as the TCP sink advertised window size.

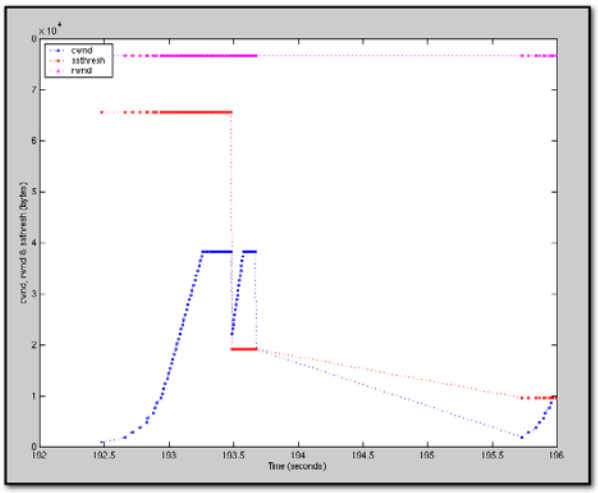

Fig. 7. Congestion window (cwnd), slow start threshold (ssthresh) and receiver window (rwnd).

6. CONCLUSION AND FUTUREWORK

This paper shows that TCP Tahoe protocol have the most influence on congestion control. In the future, we will investigate the problem of optimal buffer allocation for loss-based and delay-based protocols in more detail. Also, we will explore mechanisms for detecting loss-based flows sharing the same link queue with delay-based flows.

References

[1] J. Park, I. Kim, S. Kim, S. Park, B. Koo, K. Shin, K. Seo, and J. Cha,
"MPEG-4 video codec on an ARM core and AMBA," in Proc. of Workshop and Exhibition on MPEG-4, Jun. 2001, pp. 95–98.

[2] http://en.wikipedia.org/wiki/Transmission_Control_Protocol.

[3] Cheng Peng Fu, Bin Zhou, Jian Ling Zhang, "Modeling TCP Veno Throughput over Wired/Wireless Networks," IEEE COMMUNICATIONS LETTERS, VOL. 11 NO. 9, SEPTEMBER 2007.

[4] C. Jin et al., "FAST TCP: From Theory to Experiments," IEEE Network, vol. 19, no. 1, pp. 4– 11, Jan./Feb. 2005.

[5] J. Wang, D. X. Wei, and S. H. Low, "Modelling and Stability of FAST TCP," in Proc. IEEE INFOCOM 2005, Miami, FL, Mar. 2005.

[6] C. Jin, D. Wei, and S. H. Low, "FAST TCP for high-speed long-distance networks," Internet draft draft-jwl-tcp-fast-01.txt. [Online]. http://netlab.caltech.edu/pub/papers/draft-jwl-tcp-fast-01.txt.

[7] David X., Wei Cheng Jin, Steven H. Low Sanjay Hegde, “FAST TCP: Motivation, Architecture, Algorithms, Performance,” IEEE/ACM Transactions on Networking, 14(6):1246-1259, Dec 2006.

[8] J. Wang, A. Tang, S. H. Low,“Local stability of Fast TCP”, proc. IEEE conf. decision and control, December 2004.

[9] Jacobson, V., "Congestion Avoidance and Control Computer Communication Review, vol. 18, no. 4, pp. 314-329, ftp://ftp.ee.lbl.gov/papers/congavoid.ps.Z", August 1988.

[10] T. V. Lakshman, Member, IEEE, and Upamanyu Madhow, Senior Member, IEEE, “The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss,” IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 5, NO. 3, JUNE 1997.

[11] By Steven H. Low, Fernando Paganini, and John C. Doyle, “Internet Congestion Control,” IEEE Control Systems Magazine, 0272-1708/02/$17.00©2002 IEEE.

[12] Mandakini Tayade et al, " REVIEW OF DIFFERENT TCP VARIANTS IN AD-HOC NETWORKS", International Journal of Engineering Scince and Technology(IJEST), Vol. 3, No. 3, March 2011.

[13] T.V. Lakshman, Upamanyu Madhow, Bernhard Suter, “TCP/IP Performance with Random Loss and Bidirectional Congestion”, IEEE/ACM Transactions on networking, Vol. 8, NO. 5, 2000.

[14] Cheng Peng Fu, Bin Zhou, Jian Ling Zhang,“Modeling TCP Veno Throughput over Wired/Wireless Networks”, IEEE communications letters, Vol. 11, No. 9, 2007.

[15] C. Jin, D. Wei, and S. H. Low, “FAST TCP: Motivation, architecture, algorithms, performance,” in Proc. IEEE INFOCOM 2004, vol. 4, Mar. 2004, pp. 2490-2501.

[16] F. Paganini, Z. Wang, J. C. Doyle, and S. H. Low, “Congestion control for high performance, stability, and fairness in general networks,” IEEE/ACM Trans. Networking, vol. 13, pp. 43-56, Feb. 2005.

[17] Nostrum, http://www.imit.kth.se/info/FOFU/Nostrum.

[18] www.isi.edu/nsnam/ns.

[19] R. Lemaire, F. Clermidy, Y. Durand, D. Lattard, and A. Jerraya, “Performance Evaluation of a NoC-Based Design for MC-CDMA Telecommunications Using NS-2,” in The 16th IEEE International Workshop on Rapid System Prototyping, Jun. 2005, pp. 24–30.