RESEARCH ON COMPUTING SCIENCE

ISSN: 1665-9899

Advances on Digital System Design

Luis A. Villa
César A. Osuna
Marco A. Ramírez
(Eds.)

Advances on Digital System Design

Avances en el Diseño de Sistemas Digitales

Research on Computing Science

Research on Computing Science es una publicación trimestral, de circulación internacional, editada por Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica desarrollo tecnológico de la comunidad científica internacional. **Volumen 15**, septiembre, 2005. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. 04-2004-062613250000-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor Responsable: *Juan Humberto Sossa Azuela, SOAJ 560723*

***Research on Computing Science** is published by the Centre for Computing Research at IPN. **Volume 15**, September, 2005. Printing 500 issues. The authors are responsible of the contents of their articles. All rights reserved. Not part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of Centre for Computing Research. Printed in Mexico City, September, 2005, in the IPN Graphic Workshop-Publication Office.*

Volume 15

Volumen 15

Advances on Digital System Design

vances en el Diseño de Sistemas Digitales

olume Editors:

Editores del Volumen

Luis Villa
César A. Osuna
Marco A. Ramírez

stituto Politécnico Nacional
entro de Investigación en Computación
éxico, 2005

ISSN: 1665-9899

Editores

Luis Villa
César A. Osuna
Marco A. Ramírez

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

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

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

Printing: 500
Tiraje: 500

Printed in Mexico
Impreso en México

Preface

Technology trends on 21st Century have shown to us both: the miniaturization and the enormous interdependency between digital and analogical worlds. Micro-Electro-Mechanical Systems (MEMS) are actually the most solid alternatives which allow the integration of digital and analogical systems in the same nano-space.
This current volume on *Research on Computing Science* presents the last advances on the wide field of Digital and Analog System Design, considering in a near future aboard the state of the art of the nano-technological integration of both systems.
With this goal in mind, the editors have divided the book on five thematic tracks, in order to facilitate the reading of the contributions, instead of establishing a disciplinary dependency among topics.

First track of **Computer Architecture** presents several comparisons in terms of execution time of some algorithms, also some software methodologies for the designing of Multiprocessor Systems, reconfigurable hardware by concurrent replication of active resources and digital design modules for vital signals processing. Second track of **Digital Signal and Image Processing** presents several techniques for image compression and image filtering with the goal of image de-noising. Third track of **Virtual Instrumentation**, presents some works related with PC-based supervision-control for an automated greenhouse and a university nanosatellite design. Next track of **Real Time Systems,** describes techniques for digital filter design and some applications of periodic tasks in concurrent systems. Last track of **Control,** finally presents some studies concerning the diagnosis of non-linear systems, fault diagnosis methods for AC induction motors and an application for supervision and estimation using Neural Networks.

The eighteen papers integrating this volume were selected among the thirty-eight sent papers for review by the Volume Editorial Board after a rigorous reviewing process.

We would like to thank all participants, authors and reviewers for their enthusiastic and invaluable participation in this multidisciplinary effort. Special thanks to the Center for Computing Research from National Polytechnic Institute, Mexico for the economical and logistical support to end with this project.

Mexico, D.F. September 2005

Luis A. Villa
Cesar A. Osuna
Marco A. Ramírez

Table of Contents

Índice

Virtual Instrumentation and Digital System Design

Real Time Systems

Control

Computer Architecture

Improving the dependability of dynamically reconfigurable hardware by concurrent replication of active resources

José M. M. Ferreira, Manuel G. Gericota

FEUP / DEEC, R. Roberto Frias, 4200-465 Porto PORTUGAL jmf@fe.up.pt
ISEP / DEE, R. Ant. Bern. Almeida, 4200-072 Porto PORTUGAL mgg@dee.isep.ipp.pt

Abstract - This presentation describes a low-level technique to replicate active resources (i.e. resources that are being used by functions that are currently running) in dynamically reconfigurable FPGAs, with the main objective of releasing them to be tested in a non-intrusive way. This technique may be used to support i) Online concurrent testing to detect any faults that emerge during system operation, ii) Enhanced fault tolerance[1] (restoring the reliability index by replacing a defective resource), and iii) Reallocation of the FPGA logic space to prevent excessive delays or wasting resources due to fragmentation. All solutions proposed reuse the IEEE 1149.1 (JTAG) test access port and boundary-scan architecture to ensure a low-cost / low overhead implementation.

1 Introduction

SRAM-based field-programmable gate arrays (FPGAs) comprise an array of uncommitted configurable logic blocks (CLBs) and input / output blocks (IOBs), which are interconnectable via configurable routing resources. A large number of SRAM cells define the operation of all such blocks and interconnections. Dynamically reconfigurable FPGAs (DR-FPGAs) that support partial reconfiguration enable the device logic space to be reconfigured selectively, i.e. the redefinition of logic functions will only address the required subset of the device logic space. Any function implemented on the remaining logic space will continue to operate undisturbed while the reconfiguration process takes place. Due to some discrepancies in terminology among various authors, it is useful to state that the expression *dynamically reconfigurable FPGA* will be used throughout this work to refer to those devices that support partial reconfiguration. DR-FPGAs enable the implementation of virtual hardware by appropriate scheduling of applications. Efficient time and space management enable the implementation of applications which in total may exceed 100% of the logic space available.
Technological improvements enabled the recent introduction of self-reconfigurable FPGAs (SR-FPGAs), where an internal function may control the reconfiguration of the device logic space. SR-FPGAs are able to further reduce the cost and size of adaptive systems, by implementing online management tasks within the FPGA itself.

[1] The extension of this technique to enhance fault-tolerant architectures has just started in May 2005 and is being financed by the Fundação para a Ciência e a Tecnologia (FCT contract number POSC/EEA-ESE/55680/2004).

The increasing amount of logic available in FPGAs and the reduction of the reconfiguration time, partly due to the possibility of partial reconfiguration, extended the concept of virtual hardware to the implementation of multiple applications sharing the same logic resources in the spatial and temporal domains. However, higher complexity comes hand-in-hand with higher vulnerability. Transient phenomena e.g. single-event upsets (SEUs) or single-event transients (SETs), may lead to modifications in the configuration memory or to state modifications, particularly for larger die sizes [1, 2]. This problem is non-negligible at ground level, and it is further aggravated when these devices are used in space applications, where the cosmic radiation causing SEUs and SETs is far more important. On the other hand, the threat of electromigration also increases with smaller technological scales, and may lead to permanent physical damage. Even when the cause of the problem is not permanent (i.e. modifications in the configuration memory due to an SEU), the circuit may fail if corrective action is not taken in due time. Altogether, these factors indicate that good production tests are no longer enough to guarantee fault-free operation. Error conditions or physical defects may (and will) emerge in the field, and the only way to ensure reliability is to implement online concurrent fault detection and mitigation solutions. The concurrent replication of active resources herein presented enables an effective framework to ensure dependable system design, comprising the following components:

- **Online concurrent testing**: Active replication is used as the basis of a non-intrusive concurrent testing strategy, whereby each resource is replicated (functional and state information) and released for testing. Fault detection latency is related to the size of the FPGA — an emerging defect will only be detected when the affected resource is again under test. Error conditions may eventually cause system malfunction, if the fault latency is higher than the system inertia.
- **Enhanced fault tolerance**: Spatial redundancy architectures may be a solution when fault detection latency is not acceptable. However, if more than one module fails, the system may also fail (a triple redundancy system can only tolerate single module failures). Concurrent testing will identify the defective resource, which will be replaced to reestablish the reliability index. A simpler form of replication suffices in this case, since state information does not have to be transferred.

Concurrent testing and fault tolerance are necessary, but may not be sufficient to guarantee sustainable performance after many reconfiguration sessions. Increasing propagation delays due to poor rerouting and excessive fragmentation of the FPGA logic space are two major reasons of concern in this context. A dependable framework for dependable system design must therefore also support concurrent defragmentation, to enable the activation / deactivation of functions as needed at any given moment. The research work done so far is not divided equally among concurrent testing, enhancement of fault-tolerance, and concurrent defragmentation. Online concurrent testing concentrated most of our efforts, and has been validated via practical experimentation using a DR-FPGA from Xilinx. Extensive experimental data is available in this case, and an extract will be presented to validate the solutions presented. Experimental data is not yet available for the two other areas. Fault tolerance strategies are the main focus of current research, particularly in what concerns the implementation of self-repair methods using SR-FPGAs. Logic space defragmentation is an area that waits for the opportunity to develop appropriate higher

level management solutions that are able to exploit the replication technique proposed. Those areas will be therefore presented in less detail, but may lead to stimulating discussions and also to possible cooperation efforts in the near future.

2 Concurrent replication of active resources

Several off-line and online strategies have been proposed to test and diagnose FPGA faults [3-16]. The concurrent test approach herein presented reuses some of the ideas described in the literature, but eliminates their drawbacks by using *active replication*, which enables the relocation of the functionality allocated to each CLB, without halting the system. This approach is feasible even when the CLB is active, i.e. when it is part of an implemented function that is actually being executed [16]. A dynamic rotation mechanism ensures that all FPGA CLBs are released and tested within a given latency. The exclusive (re)use of the Boundary Scan (BS) test infrastructure to release and test the CLBs brings the additional benefit of reduced overhead at board level, since no other resources (other than those of the FPGA itself) are used.

Releasing active CLBs for testing requires their replication into CLBs already tested and available, in a way that is completely transparent to the application(s) that are currently running. This task is not trivial due to two major issues: i) configuration memory organization, and ii) internal state information.

The configuration memory may be visualized as a rectangular array of bits, which are grouped into one-bit wide vertical frames, extending from the top to the bottom of the array. The atomic unit of configuration is one frame — it is the smallest portion of the configuration memory that can be written to or read from. These frames are grouped together into larger units called columns. Each CLB column has an associated configuration column, with multiple frames, which mixes internal CLB configuration and state information, and column routing and interconnect information. The organization of the entire configuration memory into frames enables the online concurrent partial reconfiguration of the FPGA.

The configuration process is a sequential mechanism that spans through some (or eventually all) CLB configuration columns. More than one column may be affected during the replication of an active CLB, since its input and output signals (as well as those in its replica) may cross several columns before reaching its source or destination. Any partial reconfiguration procedure must ensure that the signals from the replicated CLB are not broken before being totally re-established from its replica. It is also important to ensure that the functionality of the CLB replica must be perfectly stable before its outputs are connected to the system, so as to avoid output glitches. The replication of CLBs is divided into two phases, as illustrated in figure 1.

In the first phase, the internal configuration of the CLB is copied and the inputs of both CLBs are placed in parallel. Due to the low-speed characteristics of the configuration interface used (the BS interface), the reconfiguration time is relatively long when compared with the system speed of operation. Therefore, the outputs of the CLB replica will be perfectly stable before being connected to the circuit, in the second phase. Both CLBs must remain in parallel for at least one system clock cycle,

to avoid output glitches. Notice that rewriting the same configuration data does not generate any transient signals, so the remaining resources addressed by the configuration frames are not affected. Another major requirement for the success of the replication process is the correct transferal of state information. If the current CLB function is purely combinational, a simple read-modify-write procedure will suffice to accomplish the replication process. However, in the case of a sequential function, the internal state information must be preserved and no write-operations may be lost while this process goes on.

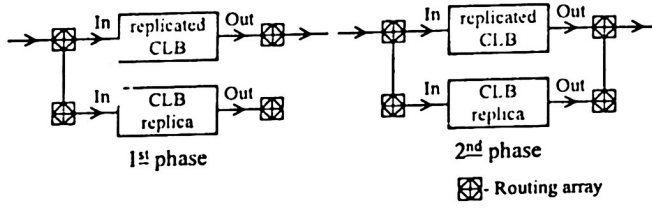


Fig. 1. Two-phase CLB replication process

When dealing with synchronous free-running clock circuits, the two-phase replication process that was previously described solves this problem. Between the first and the second phase, the CLB replica has the same inputs as the replicated CLB and all its four flip-flops acquire the state information, even if the system clock frequency is an order of magnitude lower than the clock frequency of the BS infrastructure, which is used for reconfiguration purposes. Several experiments made using this class of circuits have shown the effectiveness of this method in the replication of active CLBs. No loss of state information or the presence of output glitches was observed.

A different situation is present in the case of synchronous gated-clock circuits, where input acquisition by the flip-flop is controlled by a clock enable signal. In such cases, it is not possible to ensure that this signal will be active during the replication process, and that the value at the input of the replica flip-flops will be captured. On the other hand, it is not feasible to set this signal as part of the replication process, because the value present at the input of the replica flip-flops might differ from the one captured by the replicated flip-flops, in which case a coherency problem will occur. Furthermore, the state of the flip-flops could be updated during the replication process. A replication aid block is used to solve this problem. This block manages the transferal of state information from the replicated flip-flops to the replica flip-flops, while enabling its update by the circuit, at any moment, without losing new state information or delaying the replication process.

Practical experiments performed using a Virtex XCV200 over the ITC'99 Benchmark Circuits from the Politecnico di Torino [17], demonstrated the effectiveness of the proposed approach. These circuits are purely synchronous with only one single-phase clock signal present. However, the procedures presented are also applicable to multiple clock / multiple phase circuits, since only one clock signal is involved in the replication process at each time, provided that the slowest "clock" period is higher than the duration of the replication process. The proposed method is also effective

when dealing with asynchronous circuits (using D latches instead of flip-flops), where the same replication aid block and the same replication sequence may be used.
A further remark must be made concerning the relocation of routing resources. Since different paths are used while paralleling the original and replica interconnections, each of them will have a different propagation delay. This means that if the signal level at the output of the CLB source changes, the signal at the input of the CLB destination will show an interval of fuzziness. However, the impedance of the routing switches will limit the current flow in the interconnection, and hence this behaviour does not damage the FPGA. Nevertheless, and for transient analysis, the propagation delay associated to the parallel interconnections, shall be the longer of the two paths.
The LUTs in the CLB can also be configured as memory modules (RAMs) for user applications. However, the extension of this concept to the replication of LUT/RAMs is not feasible. The content of the LUT/RAMs may be read and written through the configuration memory, but there is no mechanism, other than to stop the system, capable of ensuring the coherency of the values, if there is a write attempt during the replication interval, as stated in [2]. Furthermore, since frames span an entire column of CLB slices, a given bit in all slices is written with the same command. Therefore, it is necessary to ensure that either all the remaining data in the slice is constant, or it is also changed externally through partial reconfiguration. Even not being replicated, LUT/RAMs should not lie in any column that could be affected by the replication process.
Depending on the method used to create the reconfiguration files, the replication procedure can also recover from errors caused by transient faults in the on-chip configuration memory cells. A typical example of such errors are SEUs in space environments, which modify the logic function originally implemented in the FPGA. Since Virtex FPGAs enable readback operations, a completely automatic read--modify-write procedure could be implemented to replicate the CLBs using local processing resources. In this case, any transient fault in the configuration memory is propagated and will affect the functionality of the CLB replica. On the other hand, if the reconfiguration files are generated from the initial configuration file stored in an external memory, any error due to SEUs is corrected when the affected blocks are replicated.

3 Online concurrent testing

The configurable structure of the CLB requires the use of a minimum number of test configurations to perform a complete test of its structure, with a specific set of test vectors applied to each test configuration. Since the implementation structure of the CLB primitives (LUTs, multiplexers, flip-flops) is not known, a hybrid fault model was considered [7] (see also [10, 11] for an extensive study concerning FPGA fault models). The BS infrastructure is reused to apply the 40 test vectors required to test each CLB, and to capture the test responses. Since the application of test vectors via the BS register would be intrusive, a user test register must be used (the Virtex family enables two BS user registers). The register created for this purpose comprises 13 cells, corresponding to the maximum number of CLB inputs, and is fully compliant

with the IEEE 1149.1 standard. The seven CLBs occupied by this register and the two CLBs occupied by the replication aid block, are the only FPGA hardware overhead that is implied by our test methodology. It accounts for less than 1% of the CLB resources of the Xilinx Virtex XCV200 device (array size = 28x42 CLBs). Each Virtex CLB comprises two slices that are exactly equal. In total, each CLB has 13 inputs (test vectors are applied to both slices of all CLBs under test simultaneously) and 12 outputs (6 from each slice). Since the outputs of each slice are captured independently, fault location can be resolved to a single slice.

The dynamic rotation mechanism used for releasing CLBs to be tested should have a minimum influence in the system operation, as well as reduced reconfiguration cost overhead. This cost depends on the number of reconfiguration frames needed to replicate and release each CLB, since a great number of frames would imply a longer test time. The impact of this process in the overall system operation is mainly related to the delays imposed by re-routed paths, which may now be longer (reducing the maximum frequency of operation). Two strategies were considered to rotate a free CLB across the FPGA logic space: horizontal and vertical. In the horizontal rotation strategy the free CLB rotates along a horizontal path covering all the CLBs in the array. The replication is performed between neighbouring CLBs, due to scarcity of routing resources, and to avoid higher path delays. The same principle applies to the vertical rotation strategy, where the free CLB is rotated along a vertical path.

The results of practical experiments performed over a subset of the ITC'99 benchmarks using these two strategies are presented in table 1. Generally, the vertical rotation scheme is seen to perform much better, be it in terms of the reduction in the maximum frequency of operation (7% in the average for vertical rotation, against 18% average for horizontal rotation) or in what concerns the size of the partial reconfiguration files. The vertical organization of the reconfiguration vectors explains why the size of the reconfiguration files is in the average 20% higher for horizontal rotation, which always involves two columns.

The influence over the maximum frequency of operation is explained by the pair of dedicated paths per CLB that propagate carry signals vertically between adjacent CLBs. When the rotation process breaks a dedicated carry path, due to the insertion of the free CLB, the propagation of this carry signal between the nearest adjacent CLBs (above and below the free CLB) is re-established through generic routing resources, increasing the path delay. If the implemented circuit has one or more of these carry signals, the horizontal rotation would break all the carry nets, increasing path delays, but the vertical rotation would break only one of them at a time. In this case, the vertical strategy becomes preferable. When no carry signals are used, two other factors must be considered: i) the number of signals with high fanout, and ii) the placement shape (rectangular, square, circular, etc.) and orientation (horizontal, vertical) of the circuits implemented in the FPGA. In rectangular / horizontal implementations, and when many high fanout signals are present, the horizontal strategy is preferable, since the maximum frequency of operation is less degraded (this could be more important than the size of reconfiguration files, when dealing with high-speed applications).

If we look into the mean size of the reconfiguration files per CLB, table 1 shows that the vertical rotation scheme also performs better (nearly 10% in average). This table shows that the mean size of the reconfiguration files per CLB increases as the number

of CLBs used in the implementation also increases, because it becomes more difficult to find good routing alternatives to those signals involved in the replication process.

Table 1. Frequency deviation, reconfiguration file size, and size per CLB

Circuit Refer.	Number of CLBs	Maximum frequency deviation (%)		Size of reconfiguration files in total (bytes)		Size of reconfiguration files per CLB (bytes)		Ratio (horiz. / vert.)
		Vert.	Horiz.	Vert.	Horiz.	Vert.	Horiz.	
B01	6	-5,5	0,0	48 350	56 102	8 058	9 350	16,0
B02	1	0,0	0,0	7 016	10 623	7 016	10 623	51,4
B03	11	-1,9	-4,9	120 705	138 484	10 973	12 589	14,7
B04	54	-6,1	-29,3	548 595	665 419	10 159	12 322	21,3
B05	103	-17,3	-36,9	1 130 985	1 286 031	10 980	12 485	13,7
B06	5	-2,7	0,0	45 291	53 503	9 058	10 700	18,1
B07	31	-23,6	-37,8	354 367	425 214	11 431	13 716	20,0
B08	17	-5,8	-5,8	150 093	178 339	8 829	10 490	18,8
B09	12	-1,8	-4,9	112 107	129 855	9 342	10 821	15,8
B10	20	-7,5	-7,6	195 571	245 455	9 778	12 272	25,5
B11	39	-10,5	-36,0	500 261	614 093	12 827	15 745	22,8
B12	119	0,0	-1,2	1 275 804	1 631 953	10 721	13 713	27,9
B13	37	-4,3	-42,8	258 827	332954	6 995	8 998	28,6
B14	333	-13,5	-47,8	5 195 444	6 070 485	15 601	18 229	16,8

The back and forth dynamic free-CLB rotation across the chip implies a variable latency. The time to again reach a given CLB alternates, according to the rotation direction, between a maximum and a minimum value, depending on the device size (number of CLB columns and rows). The fault detection latency is bounded by the following limits:

$$\tau_{scan_{max}} = ((\#CLB_{rows} \times \#CLB_{columns}) - 1) \times 2 \times (t_{reconf} + t_{test})$$

$$\tau_{scan_{min}} = 2 \times (t_{reconf} + t_{test})$$

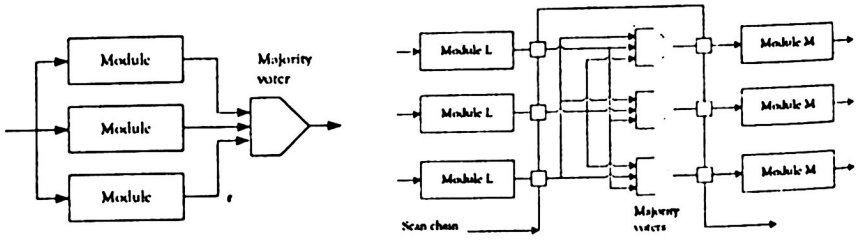
(t_{reconf} is time needed to complete a CLB replication and t_{test} is the time needed to test a free CLB)

When the rotation process is complete, the initial routing is restored. The whole process may then be repeated or paused, depending on the overall test strategy. In order to estimate the worst case fault detection latency, we must take into account the time needed to carry out each step of the proposed approach. The replication of each CLB dominates the total time per CLB, in particular in the case of those circuits requiring a replication aid block (in which case approx. 35 ms are required to complete the replication process). The various test configurations required to complete a structural test of each CLB, together with the time required to shift in and out test data, contribute with approximately 10 ms, leading to a total time per CLB

that is close to 45 ms. In a typical scenario, like those that correspond to the ITC'99 benchmark circuits (25% of CLBs requiring a replication aid block, 50% of CLBs not requiring such blocks, and 25% of empty CLBs), the total test time to cover the 1.176 CLBs in the XVC200 is above 40 seconds.

4 Enhanced fault tolerance

Active replication enables non-intrusive concurrent testing and logic space defragmentation, but cannot avoid performance degradation due to higher propagation delays, in the case of signals that are re-routed via longer interconnection paths. On the other hand, fault detection latency enables the propagation of fault effects, which may eventually lead to irreversible malfunction of the whole system. These restrictions may not be important in many application domains, and namely when the fault detection latency is small compared to system inertia. However, in the case of mission-critical applications, a higher reliability solution must be devised.
Triple modular redundancy (TMR) is the best known form of spatial redundancy, and may be represented as shown in figure 2.a. Three identical modules (M) receive the same inputs and drive a majority-voter that produces the circuit output. When more than one module fails, the circuit fails. Tolerance to multiple module failures may of course be achieved at the cost of providing higher redundancy, leading to NMR architectures. Replication may take place at various hierarchical levels, so each module may be a simple gate, or a much more complex resource, including mid-range functional blocks, components (e.g. a microprocessor), or even a complete system. In the basic configuration shown in figure 2.a, the reliability of the circuit will depend on the reliability of the voter (if the voter fails, the circuit fails). Special design and implementation techniques may be used to improve the reliability of the voter circuit. However, if such solutions are not considered satisfactory, the voting element itself may be replicated as well, leading to what is known as N-NMR architectures. Additionally, design diversity may be enforced to further enhance reliability.



a. TMR with a single voting element **b.** T-TMR with scan fault detection

Fig. 2. Fault tolerance via spatial redundancy

Particularly in the case of low level modular redundancy, DR-FPGAs bring two important advantages to the cost x benefit model of NMR implementations: **i)** since

each function may be implemented only when needed (and afterwards removed to release FPGA floor space), the additional spatial requirements of modular redundancy solely address those blocks that have to be implemented at any given time; **ii)** any defective resource that caused module failure may be identified and replaced, restoring the reliability index. The occurrence of a fault will be indicated by a discrepancy at the output of a module or voter. An internal 1149.1 scan chain able to capture the output of modules and voters, as shown in figure 2.b, enables the identification of the defective block. The fault masking properties of modular redundancy will ensure that circuit operation will not be affected, provided that a second module / voter in the same set does not fail, before the fault correction procedure is complete. The fault detection procedure launches a background task to readback the configuration bitstream of the area where the affected module is located. If an incoherency is found, the test controller restores the original configuration and eliminates the cause of the failure. If no error in the configuration bitstream is detected after the readback-and-compare operation, but the fault persists, the most probable reason is the existence of a physical defect. When the defective resource is found, it is flagged to avoid further usage and replaced to restore the reliability index. Notice that this solution confines the fault effect to the defective module (its output is masked), but does not eliminate the fault detection latency. However, the worst case fault latency is no longer given by the time taken to test the full CLB matrix, since the concurrent fault detection procedure is now looking for incoherencies at the module and voter outputs. The important thing to do is to identify the cause of the incoherency and to eliminate it. Appropriate action (i.e. correcting the contents of the reconfiguration memory or replicating the defective CLB) will then reestablish the reliability index and bring the circuit back to its full fault-tolerance features.

5 Conclusion

The active replication technique described in this presentation enabled the proposal of a truly non-intrusive online concurrent testing solution that was validated using an XCV200-based prototyping board. Notice that the structure of the current Xilinx Virtex II devices is the same as in the older XCV200 that was available when this project started (the embedded microprocessor cores and the higher number of reconfigurable resources make no difference from this point of view). When fault detection latency cannot be tolerated, an enhanced T-TMR architecture enables fault diagnosis and guarantees fault-free operation and fast recovery of the reliability index. The same active replication technique may also be used to defragment the FPGA logic space, ensuring sustainable performance by preventing excessive path delays and reducing the waiting time imposed on incoming functions [18-21]. The development of enhanced fault-tolerant T-TMR architectures is the subject of a new R&D project that has just started in May 2005. The main objective of this new project is to improve the solution proposed in the previous section, by using SR-FPGAs to develop self-repair architectures. An embedded T-TMR microprocessor controls the internal configuration access port (ICAP) of SR-FPGAs and triggers the self-reconfiguration procedure when a defective resource is identified [22].

References

1. R. Baumann, "Soft Errors in Advanced Computer Systems," *IEEE Design and Test of Computers*, Vol. 22, No. 3, pp. 258–266, May-June 2005.
2. W. Huang, E. J. McCluskey, "A Memory Coherence Technique for Online Transient Error Recovery of FPGA Configurations," *Proc. of the 9th ACM Int. Symposium on Field-Programmable Gate Arrays*, pp. 183-192, February 2001.
3. C. Stroud et al., "Built-In Self-Test of Logic Blocks in FPGAs (Finally, A Free Lunch: BIST Without Overhead!)," *14th IEEE VLSI Test Symposium*, pp. 387-392, April 1996.
4. C. Stroud, E. Lee, M. Abramovici, "BIST-Based Diagnostic of FPGA Logic Blocks," *Proc. of the International Test Conference*, pp. 539-547, Nov. 1997.
5. C. Stroud et al., "Built-In Self-Test of FPGA Interconnect," *Proc. of the International Test Conference*, pp. 404-411, Nov. 1998.
6. Doumar, T. Ohmameuda, H. Ito, "Design of an automatic testing for FPGAs," *IEEE European Test Workshop Compendium of Papers*, pp. 152-157, May 1999.
7. W. K. Huang, F. J. Meyer, X. Chen, F. Lombardi, "Testing Configurable LUT-Based FPGA's," *IEEE Trans. on VLSI Systems*, Vol. 6, No. 2, pp. 276-283, June 1998.
8. W. K. Huang, F. J. Meyer, F. Lombardi, "An approach for detecting multiple faulty FPGA logic blocks," *IEEE Trans. on Computers*, Vol. 49, No. 1, pp. 48-54, Jan. 2000.
9. T. Inoue, S. Miyazaki, H. Fujiwara, "Universal Fault Diagnosis for Look-up Table FPGAs," *IEEE D&T of Computers*, Vol. 15, No. 1, pp. 39-44, January-March 1998.
10. M. Renovell et al., "RAM-Based FPGA's: A Test Approach for the Configurable Logic," *IEEE Int. Conference on Design, Automation and Test in Europe*, pp. 82-88, Feb. 1998.
11. M. Renovell, J. M. Portal, J. Figueras, Y. Zorian, "Testing the interconnect of RAM-based FPGAs," *IEEE D&T of Computers*, Vol. 15, No. 1, pp. 45-50, January-March 1998.
12. N. R. Shnidman et al., "On-Line Fault Detection for Bus-Based Field Programmable Gate Arrays," *IEEE Trans. on VLSI Systems*, Vol. 6, No. 4, pp. 656-666, December 1998.
13. M. Abramovici et al., "On-Line Testing and Diagnosis of FPGAs with Roving STARs," *Proc. 5th IEEE Int. On-Line Testing Workshop*, pp. 2-7, July 1999.
14. L. Burress, P. K. Lala, "On-Line Testable Logic Design for FPGA Implementation," *Proc. of the International Test Conference*, pp. 471-478, November 1997.
15. M. Renovell et al., "Test Generation Optimization for a FPGA Application-Oriented Test Procedure," *Proc. of the 15th Design of Circuits and Integrated Systems Conference*, pp. 330-336, November 2000.
16. M. G. Gericota, G. R. Alves, M. L. Silva, J. M. Ferreira, "Active Replication: Towards a Truly SRAM-based FPGA On-Line Concurrent Testing," *Proc. of the 8th IEEE On-Line Testing Workshop*, pp. 165-169, July 2002.
17. Pol. di Torino ITC'99 benchmarks, available at http://www.cad.polito.it/tools/itc99.html
18. M. Gericota, G. Alves, M. Silva, J. Ferreira, "Run-time Defragmentation for Dynamically Reconfigurable Hardware", in *New Algorithms, Architectures and Applications for Reconfigurable Computing*, pp. 117-129, Springer, 2005, ISBN 1-4020-3127-0.
19. O. Diessel, H. El Gindy, M. Middendorf, H. Schmeck, B. Schmidt, "Dynamic scheduling of tasks on partially reconfigurable FPGAs," *IEE Proc.-Computer Digital Technology*, Vol. 147, No. 3, pp. 181-188, May 2000.
20. M. Vasilko, "DYNASTY: A Temporal Floorplanning Based CAD Framework for Dynamically Reconfigurable Logic Systems," *Proc. 9th Intl. Workshop on Field-Programmable Logic and Applications*, pp.124-133, Aug.-Sep. 1999.
21. M. Teich, et al., "Compile-time optimization of dynamic hardware reconfigurations," *Proc. Intl. Conf. on Par. and Distr. Proc. Techniques and Applications*, pp. 1097-1103, 1999.
22. M. G. Gericota, G. R. Alves, J. M. Ferreira, "A Self-Healing Real-Time System Based on Run-Time Self-Reconfiguration," *10th IEEE International Conference on Emerging Technologies and Factory Automation*, Catania, Italy, September 2005.

EXECUTION TIME COMPARISON OF ALGORITHMS FOR THE ASSIGNMENT OF REAL TIME TASKS FOR MULTIPROCESSORS

Héctor Silva-López[1], Sergio Suárez Guerra[2]

[1] Physics Department, CINVESTAV-IPN, Av. IPN. 2508, A.P. 14-740, 07360, DF, Mexico.
hsl@fis.cinvestav.mx

[2] Center for Computing Research, National Polytechnic Institute, Mexico, A.P. 75-476, C.P. 07738, Zacatenco, DF, México.
ssuarez@cic.ipn.mx

Abstract. Optimal scheduling of real-time tasks on multiprocessor systems is known to be computationally intractable for large task sets. Any practical scheduling algorithm for assign real-time tasks to a multiprocessor system presents a trade-off between its computational complexity and its performance. In this work, we give in this paper a survey on the execution time and performance of these algorithms, structuring this work in two steps. In the first step, we simulate their execution to obtain their performance and the execution time of four algorithms, using a low number of tasks. After these results were extrapolated to obtain the execution time that it would take for a bigger number of tasks than the simulated. In the second step, we chose two algorithms for partitioned its and executing on a shared memory system using Pthreads and on a distributed memory system employing Parallel Virtual Machine and Message Passing Interface tools.

1 Introduction

There are two strategies for real-time tasks scheduling on a multiprocessor system. In a Global scheme each occurrence of a real-time task may be executed on a different processor. In contrast, a partitioning scheme enforces that all occurrences of a particular task are executed on the same processor. Among the two methods, the partitioned method has received the most attention in the research literature. The main reason for this is that the partitioned method can easily be used to guarantee run-time performance (in terms of schedulability). Dhall and Liu in [1], two heuristic assignment schemes are proposed, referred to as the RMNF (Rate Monotonic Next-Fit) and the RMFF (Rate Monotonic First-Fit). The schemes are based on the next-fit and first-fit bin-packing heuristic, respectively. In both schemes, tasks are sorted in no decreasing order according of their periods before the assignment is started. The FFDUF (First-Fit Decreasing-Utilization Factor)

method is a variation of the first-fit heuristic scheme. Here tasks are sorted in the order their load factor [2]. In [3], a best-fit bin-packing heuristic is used as the basis for the monotonic best-fit (RMBF) schemes. Similar to RMFF, the RMBF attempts to assign tasks to processors that have been marked as full. In [4], two heuristic assignments proposed, the RMST (Rate Monotonic Small Tasks) algorithm. It is for systems in which tasks assigned to each processor are scheduled rate-monotonically. It first sorts the periodic tasks in nondecreasing order according to their parameters X_i's, which are calculated with the following equation:

$$X_i = \log_2 p_i - \lfloor \log_2 p_i \rfloor$$

It then assigns the tasks in this order on processors in the first-fit manner. The other algorithm is the RMGT (Rate Monotonic General Tasks), first partitions all periodic tasks two subsets according to their utilization. Tasks whose utilization is equal to or smaller than 1/3 are in one subset. These tasks are first assigned to processors according to RMST algorithm. Then, the large tasks whose utilization is larger than 1/3 are assigned on the first-fit basis to processors each of which has at most one task assigned by RMST algorithm.

The non-partitioned method has received considerably less attention, mainly because the following limitations. First, no efficient schedulability tests currently exist for the non-partitioned method. The only known necessary and sufficient schedulability test for non-partitioned method has an exponential time-complexity [5]. The complexity can be duced with sufficient schedulability tests to a polynomial [6, 7, 8, 9] or pseudo-polynomial [8] time complexity. Second, no efficient optimal priority-assignment scheme has been found for the non-partitioned method.

The purpose of this paper is to survey execution time of four algorithms for the assignment of Real-Time tasks for multiprocessors.

The remainder of this paper is organized as follows. In Section 2 the system model described. Later on in the Section 3, we present four representative algorithms which simulated to obtain the execution time for each Algorithm. Once obtained this time, results are extrapolated to obtain the execution time of a number of tasks are bigger than the simulated. In the Section 4 the two algorithms are assigned to a shared-memory multiprocessor system and to local memory distributed system. The obtained results are analyzed in the section 5. Finally in the section 6, summarizes the results reported in paper.

2 System model

We considerer a set of tasks $T=\{T_1,..., T_n\}$ of n periodic preemptive real-time tasks running on different processors. The tasks are independent (they don't share resources) have no precedence constrains. Each task T_i arrives in the system at time a_i. The life-time

of each task T_i consists of a fixed number of instances r_i. After the execution of r_i instances, the task leaves the system. The time interval between the arrival of the first instances of two consecutive tasks T_x and T_y is defined as $L_{xy} = a_y - a_x$. In this model, T_i is the period and C_i is the worst case computation time of task. In their seminar work, Liu and Layland [10] showed that the utilization of system is the amount of processor load in percentage in the system and it is expressed as:

$$U = \sum_{i=1}^{n} C_i / T_i \tag{2}$$

A schedule of periodic tasks is feasible if each task T_i is assigned at least C_i before its deadline at every instance. The problem of planning a set of tasks with small load factors is taken from [5], expressed as:

$$\alpha := \max_{i=1,\ldots,n} U_i \tag{3}$$

It represents the maximal load factor of any single task. For all practical purposes, we may assume that a task set contains only tasks if $\alpha \leq ½$.

We initially assume the following system model:

1) Tasks are independent, arrive periodically and can be preempted. Hence, at every moment, a dispatcher determines which task to execute. Tasks do not require exclusive access to any other resource than a processor.
2) The cost of pre-emption is zero because we consider cache misses that occur when a task arrives to be included in the execution time.
3) Worst case execution time is known a priori.
4) Only partitioned scheme algorithms are considered.

3 Task assignment algorithms simulation

The algorithms to simulate are the RMNF, RMFF, RMST and the RMGT. The simulation consists on executing from 100 to 1000 tasks with increments of 1. In each increment the algorithm is evaluated from 1 to increment. The tasks are generated in the following way:
1) The number of tasks of $100 \leq n \leq 1000$
2) The period is generated following uniform distribution function with $1 \leq T_i \leq 500$
3) The compute time is generated of $0 < C_i \leq \alpha T_i$
4) The performance of the system is measured using two values of load factor, $\alpha = 0.2$ and $\alpha = 0.5$
5) Every increment runs 10 times and an average is obtained

The executed time of each algorithm is obtained with a Pc Intel Pentium III 650 MHz with 128 Mb of RAM and running on the Operating System Linux. The function used for the measurements is psched_get_time ().

Since an optimal task assignment cannot be calculated for large task sets, we use the total load $\left(U = \sum_{i=1}^{k} u_i\right)$ used in [11], to obtain a Lower Bound for the number of processors required, where u_i is the utilization of a task T_i.

The results can be seen in the figure 1 and figure 2 with load factor of α =0.2 and α =0.5, the worst algorithm is the RMNF, followed by the RMFF, because they need a bigger number of processors to execute the same quantity of tasks than the others. The two algorithms proposed by Burchard are better, specifically for the RMGT needs a smaller number of processors than the RMST, for example for α =0.2 and 1000 tasks, the RMST needs 137 processors, while for the RMGT only needs 133 processors, for α =0.5 and 1000 tasks, the RMST occupies 339 processors and the RMGT only occupies 300 processors.

3.1 Execution time extrapolation for α=0.2 and α =0.5

In the table 1and table 2 represents execution time of each algorithm with α =0.2 and =0.5. The time in clock ticks.

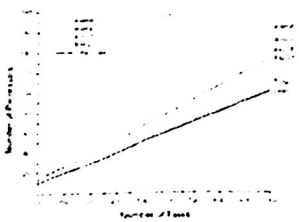


Fig. 1 Sequential simulation for α =0.2.

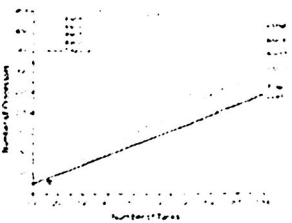


Fig. 2 Sequential simulation for α =0.5.

Table 1. Sequential Time, α=0.2

Alg \ Tasks	100	200	300	400	500	600	700	800	900	1000
Gen_tasks	2	3	4	6	7	8	10	10	12	14
RMNF	0	0	0	1	2	4	4	5	6	8
RMFF	1	9	33	74	151	261	410	632	934	1237
RMST	1	9	32	76	150	262	419	637	1207	1219
RMGT	1	9	31	76	151	262	419	634	895	1217
Total_Load	0	0	0	0	0	0	1	1	1	2

Table 2. Sequential Time, α=0.5

Alg \ Tasks	100	200	300	400	500	600	700	800	900	1000
Gen_tasks	2	3	4	5	7	8	9	11	12	14
RMNF	0	0	0	1	2	3	4	5	7	9
RMFF	2	15	43	108	228	378	611	895	1363	1835
RMST	2	14	42	110	219	369	595	875	1266	1794
RMGT	1	14	37	115	212	281	455	671	958	1324
Total_load	0	0	0	0	0	0	0	0	0	1

3.2 Polynomial approximation evaluation, α =0.2 and α =0.5

Using the approaches polynomial for each algorithm, we have calculated the time would take the program (in clock ticks) for a large number of tasks, see table 3 and table 4.
We can convert the clock ticks to seconds, with the following expression: clock ticks / CLK_TCK. Where: CLK_TCK is equal to 18.2.

Table 3. Execution time for n tasks, α=0.2

Alg \ Tasks	1,000	5,000	10,000	50,000	100,000	500,000	1,000,000
Gen tasks	13.43667	65.31667	130.16667	648.96667	1297.46667	6485.46667	12970.4667
RMNF	7.81394	162.94858	629.59433	15271.8583	60847.4333	1516361.83	6063029.33
RMFF	1244.6365	153486.713	1220438.05	151686883	1212564573	1.5148E+11	1.2117E+12
RMST	1309.15333	51092.8333	214572.433	5572409.23	22394705.2	561973073	2248946033
RMGT	1219.378	140923.05	1108327	135932407	1089586921	1.3598E+11	1.0876E+12
Tot Load	1.85121	84.08033	357.32133	9361.43333	37662.0333	945885.233	3785710.23
Total	3796.26965	345814.942	2544454.57	293216981	2324646006	2.8802E+11	2.3016E+12

4 Parallel programming

We choose the RMST and RMGT algorithms for obtained the better performance than the others four algorithms. In the following sections the two algorithms were implemented for a multiprocessor system and for a distributed system.

Table 4. Execution time for n tasks, α=0.5

Alg \ Tasks	1,000	5,000	10,000	50,000	100,000	500,000	1,000,000
Gen_tasks	13.47	66.55	132.9	663.7	1327.2	6635.2	13270.2
RMNF	8.26943	256.13583	1043.24333	26431.0833	105893.333	2650689.33	10604429.3
RMFF	1845.26233	256048.158	2081968.63	263781271	2113875044	2.646E+11	2.1172E+12
RMST	1836.90367	267825.117	2209237.57	285183961	2274161939	2.8502E+11	2.2809E+12
RMGT	1321.13	197046.45	1638365.6	211667789	1700557618	2.133E+11	1.7071E+12
Total_load	2.005	39139571.3	7349812549	7.6553E+14	1.0147E+17	8.1499E+21	1.0468E+24
Total	5027.04043	39860813.7	7355743297	7.6553E+14	1.0147E+17	8.1499E+21	1.0468E+24

4.1 Parallel programming for shared memory

The execution of parallel programs for shared memory was realized in a Multiprocessing Computer integrated by 4 Processors (Pentium III to 750 MHz) in cascade and an Operating System Linux NET4.0 for Linux 2.4

For the implementation with Pthreads, the time were obtained in second for load factors of α=0.2 and α=0.5, the results are compared between the RMST and RMGT algorithms. In the figure 3, we can observe that the execution time for the RMST and RMGT algorithms are the same for a maximum number of threads of 10, while in the figure 4, the RMST algorithm has an execution time less than the RMGT algorithm. In the two figures is obtained that when grow the number of threads, the execution time is the same when the number of threads starting of four threads, due principally because the maximum number of processors in the computer used is four.

4.2 Parallel programming for distributed memory

In this part of paper two tools were used for the distributed memory. The first tool is Parallel Virtual Machine (PVM) and the second tool is Message Passing Interface (MPI).
For the implementation of these two tools we uses a Cluster of 9 computers (1 server and 8 clients) Pentium II to 450 MHz and 256 Mb of RAM, with an Operating System LINUX Net Hat. For the communication of each node it uses a Switch Fast Ethernet (Switch Intel Express 510T 10/100 Fast Ethernet) and cabled type FTP.

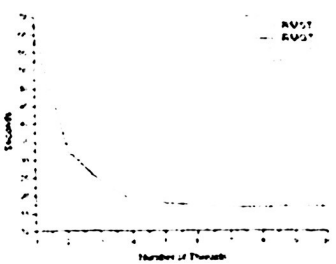


Fig. 3 Threads for α=0.2.

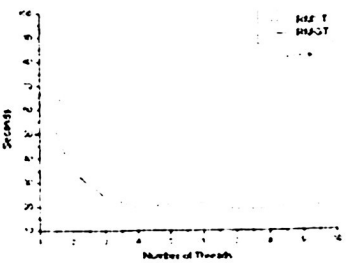


Fig. 4 Threads for α=0.5.

4.2.1 Performance Model

The objective of Performance Model is developing mathematical expressions that specify the execution time. The execution time of a parallel program is the lapse time since the first processor begins its execution until the last processor finishes its execution.

Because of space restrictions, we do not include here the details of the Performance Model of our application.

4.2.2 Parallel Virtual Machine.

Based on the communications time, to the model and the experimental results the following data were obtained, the data can be observed in the table 7, the time in seconds.
In the figure 5 correspond to experimental results and to performance model. We can appreciate that the algorithm RMST has a similar behaviour to performance model, while for the RMGT algorithm has a execution time bigger that the RMST algorithm. Based on the communications time, to the model and the experimental results, the following corresponding data were obtained for α=0.5, they are shown in the table 8, the time in seconds.

Table 7. Theoretical and experimental results with PVM, α=0.2

Slaves	Tcomunications	Tmodel	Texp-RMST	Texp-RMGT
1	0.253	78.41	97.50	125.27
2	0.429	39.51	52.09	66.82
3	0.606	26.66	34.17	42.40
4	0.782	20.32	27.48	35.41
5	0.958	16.59	20.64	25.87
6	1.134	14.14	19.24	24.24
7	1.311	11.86	14.99	20.54
8	1.486	9.24	13.95	18.78

Table 8. Theoretical and experimental results with PVM, α=0.5

Slaves	Tcomunications	Tmodel	Texp-RMST	Texp-RMGT
1	0.253	78.41	99.07	123.62
2	0.429	39.51	59.48	73.60
3	0.606	26.66	34.24	58.30
4	0.782	20.32	27.30	48.43
5	0.958	16.59	20.42	35.98
6	1.134	14.14	18.26	30.85
7	1.311	11.86	15.23	27.89
8	1.486	9.24	13.76	26.60

In the figure 6 correspond when the load factor is equal to 0.5, the RMST algorithm has behaviour seemed to presented in the figure 5, compared against the model. But the RMGT algorithm has a different behaviour to present in the figure 5. This is due mainly to great load factor (0.5), and for the number of tasks to be assigned to the processors by the RMFF algorithm is bigger than the load factor (0.2). Besides the RMFF algorithm presents a bad performance.

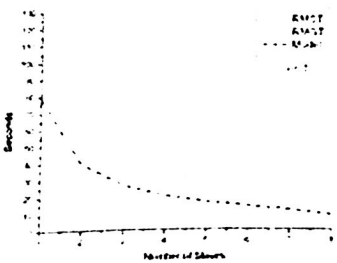


Fig. 5 PVM, α=0.2.

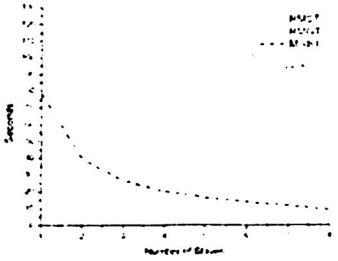


Fig. 6 PVM, α=0.5.

4.2.3 Message Passing Interface

Based in the communication time, to the model and the experimental results for the load factor equal to 0.2, we can observed in the table 9 the results obtained when the RMST and RMGT algorithms were executed in MPI, the time in seconds.

Table 9. Theoretical and experimental results with MPI, α=0.2.

Slaves	Tcomunications	Tmodel	Texp-RMST	Texp-RMGT
1	0.253	78.41	122.35	125.42
2	0.429	39.51	62.84	78.78
3	0.606	26.66	41.95	50.07
4	0.782	20.32	31.08	32.37
5	0.958	16.59	25.60	26.28
6	1.134	14.14	21.11	21.36
7	1.311	11.86	18.14	18.34
8	1.486	9.24	16.11	16.33

In the figure 7 a similar behaviour but no equal is observed in the figure 5, but in MPI for the RMST algorithm has behaviour almost similar to the RMGT algorithm when the

number of slaves are between 4 and 8 and when the number of slaves is equal to 1. Based in the communications time, to the model and the experimental results for α=0.5 (see table 10), the time in seconds.

Table 10. Theoretical and experimental results with MPI, α=0.5.

Slaves	Tcomunications	Tmodel	Texp-RMST	Texp-RMGT
1	0.253	78.41	173.95	124.21
2	0.429	39.51	85.06	65.86
3	0.606	26.66	55.34	50.26
4	0.782	20.32	42.82	40.64
5	0.958	16.59	33.21	32.22
6	1.134	14.14	28.15	22.86
7	1.311	11.86	24.28	21.85
8	1.486	9.24	21.29	20.25

In the figure 8 the behaviour of the algorithms is different compared with the figures 5, 6 and 7, the RMST algorithm consumes more execution time than the RMGT and model. It has behaviour almost similar to the RMGT algorithm when the number of slaves is from 4 to 5 and from 7 to 8 slaves.

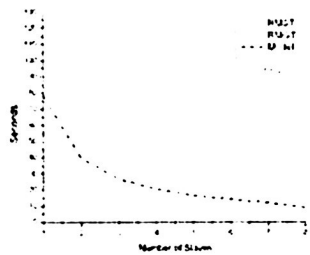


Fig. 7 MPI, α=0.2.

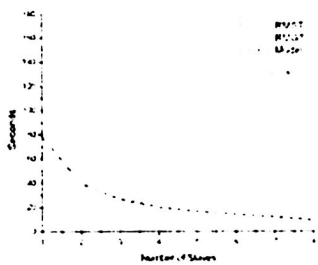


Fig. 8 MPI, α=0.5.

5 Results Comparison

In this section the execution times are presented for executing the RMGT algorithm in the sequential time as in the parallel time. The RMGT algorithm was chosen because it has a better performance that the RMST algorithm and as the execution times are very similar, it doesn't have any interest in presenting the two algorithms.

In sequential form the RMGT algorithm present an execution time less than RMST gorithm for a load factor of 0.2 and 0.5, the times obtained in experimental and theoretical form for 1,000 tasks can be observed in the table 11 (the time in seconds). In this table execution times in theoretical and experimental form are very similar for that we concludes that the theoretical results presented in the table 2 are very near to the real execution for a quantity of a lot tasks.

Table 11. Sequential Time for RMGT

Load Factor	Experimental	Theoretical
0.2	66.87	70.00
0.5	72.75	72.59

The parallel times for threads, PVM and MPI can be observed in the table 12, for load factor of 0.2 and 0.5, the time in seconds.

Table 12. Parallel Time for RMGT

Load Factor	Threads	PVM	MPI
0.2	17.18	22.35	20.57
0.5	20.56	30.33	24.29

In the table 12, the time presented for the threads is obtained of an average between time obtained of 4 and 10 threads, for the two load factors. For PVM and MPI the average time was obtained between 5 and 8 slaves.

The time presented is the desired, for PVM and MPI have a execution time bigger than the execution time of the threads, this is due to the communication time is implicit in execution of a distributed system.

6 Conclusions

We have analyzed the performance of four algorithms for the real time tasks scheduling in multiprocessor systems where their executed time were obtained. These algorithms are the RMNF, RMFF, RMST and the RMGT. In the first part of this work, the executed time obtained when simulating them were extrapolated to obtain approximate polynomial that give us an idea of the time that would take each algorithm in executing for a large number of tasks, being the time practically impossible of carrying out. For example, for load factor equal to 0.2, it would be impossible its execution after of 50,000 tasks since would take the program in execute in 6.215 months and for a load factor of 0.5, it is possible after of 10,000 tasks since would take 12.82 years.

In the second part of this work, two algorithms were chosen whose performance was much better than the other ones. The RMST and RMGT algorithms were partitioned for shared memory using threads and for distributed memory using PVM and MPI tools.

The final comparison of results shows us that when obtaining the approximate polynomials they give very near results to the obtained in experimental form, for what, the results were presented for a large number of tasks that not this very far from the reality. On the other hand, to the partitioned two algorithms the results were obtained that we were expected, that is, for the threads from 4 to 8 threads, the execution time vary very little besides were compared with the obtained in PVM and MPI. Using threads presents a better execution time than in PVM and MPI, because to execution time we need to add the communication time, implicit in a distributed system.

References

[1] J. Y. T. Leung and J. Whitehead. "On the complexity of fixed-priority scheduling of periodic, real time tasks". Performance Evaluation, 2(4):237-250, December 1982.

[2] S.K. Dhall and C. L. Liu. "On a real time scheduling problem". Operations Research, 26(1):127-140, January/February 1978.

[3] S. Davari and S. K. Dhall, "An on line algorithm for real time allocation", 19th Ann. Hawaii Int'l Conf. System Sciences, pp 133-141, 1986.

[4] Y. Oh and S. H. Son. "Tight performance bounds of heuristics for a real time scheduling problem", Technical Report CS-93-24, Univ. of Virginia, Dept. of Computer Science, May 1993.

[5] Burchard, J. Liebeherr, Y. Oh, and S.H, Son. "New strategies for assigning real time tasks to multiprocessor systems". IEEE Transactions on Computers, 44(12):1429-1442, December 1995.

[6] J.Y. T. Leung. "A new algorithm for scheduling periodic, real time tasks". Algorithmica, 4(2):209-219, 1989.

[7] B. Andersson. "Adaption of time-sensitive tasks on shared memory multiprocessor: A framework suggestion. Master's thesis", Department of Computer Engineering, Chalmers University of Technology, January 1999.

[8] S. Lauzac, R. Melhem, and D. Mossé. "Comparison of global and partitioning schemes for scheduling rate monotonic tasks on a multiprocessor". In 10th Euromicro Workshop on Real Time Systems, pages 188-195, Berlin, Germany, June 17-19, 1998.

[9] L. Lundberg. "Multiprocessor scheduling of age constraint processes". In 5th International Conference on Real Time Computing Systems and Applications, Hiroshima, Japan, October 27-29, 1998.

[10] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard Real Time environment", J. ACM, Vol. 20, no. 1, pp. 46-61, Jan. 1973.

[11] A. Burchard, J. Liebeherr, Y. Oh, and S.H, Son. "A linear time Online Task Assignment scheme for multiprocessor systems", Proc. 11th IEEE Workshop Real-Time Operating Systems and Software, pp. 28-31, May 1994.

A Soft Methodology for On-Chip Multiprocessor Design

Alejandro Matute[1], Marcos de Alba[1], and Elias Mizan[2]

[1] Tecnológico de Monterrey, Departamento de Ciencias Computacionales
Atizapán, Estado de México, México 52926
(A00464063, marcos.de.alba)@itesm.mx
[2] The University of Texas at Austin, Electrical and Computer Engineering Department
Austin, TX 78751
emizan@ece.utexas.edu

Abstract. We propose a software methodology towards the design of on-chip multiprocessors at the architecture level. This methodology will be helpful to explore multiprocessor characteristics like reliability, security, redundancy and soft-error detection and correction, among others. The technique proposed provides a clean design without loss of functionality or performance. One of its advantages is that it is scalable for the design of on-chip multiprocessors with up to n cores. We explain its application with two architecture organizations. In the first, two or more CPU cores are synchronized by executing their own applications in bulks of the same number of instructions at each core. In the second, CPU cores execute applications performing as many instructions as possible without synchronizing.

1 Introduction

The design and implementation of fully-reliable synchronization strategies between microprocessor cores is an important requirement in multiprocessor design. Architect-level simulation is one of the initial steps for multiprocessor design it provides flexibility for fast update of processor components in software, speeding up the evaluation of ideas allowing architects to explore the design space. The lack of correct multiprocessor simulators can cause misleading communication between CPU cores, affecting performance, functionality that can lead to wrong conclusions.

In the design of a superscalar microprocessor a simulator like [1] is very helpful for exploring its design space. To design the architecture of an on-chip multiprocessor we utilize a uniprocessor simulator and make modifications to it. There are different ways to accomplish this; we believe that the methodology we propose is the more natural and efficient regarding programming and functionality. This suggests replicating simulated superscalar microprocessors through multiple operating system level processes. This means that the entire operating system context of a superscalar microprocessor simulation is replicated, so that any multiprocessor architecture can be designed by multiple replicated uniprocessors.

To provide communication among different uniprocessors, or CPU cores for simplicity, we define shared variables. For example, a four-core symmetric multiprocessor, SMP, can be designed generating four identical OS-level processes of the uniprocessor simulator and defining a set of shared variables that represent shared hardware resources among the CPU cores, like a level-two unified cache. To explore replacement and writing policies on shared resources it is proposed the definition of programming subroutines which parameters include shared resources. These are implemented straightforward using OS mutual exclusion techniques.

In this work we define two multiprocessor architecture schemes. In the first, N instructions are executed at each CPU core, to assure this a synchronization mechanism is defined. In the second, every core executes freely, that is, without having to synchronize.

2 Related Work

There has been a great amount of multiprocessor design research during the last decades. Multiprocessors were originally built by interconnecting several computers or by interconnecting several CPU cores. More recently, due to the large increase on transistor density researchers have integrated two or more CPU cores on a single chip to provide higher performance, more reliability and better power efficiency.

To explore multiprocessor design, there have been proposed different strategies, including: Theoretical approaches, interconnected-CPU multiprocessors, chip multiprocessors, simulators and emulators.

Since the focus of this work is the description of a software methodology for multiprocessor design, we do not discuss hardware-based multiprocessor approaches.

There are available multiprocessor simulators like [3-6]. However, they are targeted for designing off-chip multiprocessors. In [11] it is mentioned an architecture level multiprocessor simulator, but it is not available and no further information can be obtained. Thus, we decided to concentrate on the architectural design of on-chip multiprocessors developing our own simulation infrastructure using [11] as a baseline.

As described in [2], we are exploring techniques to improve the design of on-chip multiprocessors and we have utilized the methodology explained in this paper as the main baseline simulation technique to explore multiprocessor characteristics like reliability, security, redundancy and soft-error detection and correction, among others.

The rest of the paper is organized as follows. In Section 2 we briefly explain the two previously defined simulation schemes for a multiprocessor system and the disadvantages of its loop-based solution. In Section 3 we describe their software implementation and in Section 4 we close with some results that validate our methodology.

3 Multiprocessor Schemes

The multiprocessor schemes we propose are described next.

3.1 Free running

n microprocessor cores run in parallel different applications. They execute different kinds of instructions and share input/output modules. Synchronization is required 3for a shared memory organization. A simulation of this scheme provides fast response, which is suitable for performance or throughput studies on a multiprocessor design. An example of a dual-core multiprocessor with this scheme is shown on Figure 1.

3.2 N instructions per core

n microprocessor cores run the same application in parallel. In this scheme, every core executes the same instructions in chunks of N. That is, at the beginning of an executed simulation in a dual-core multiprocessor, core 0 executes the first N instructions, and then core 1 executes the first N instructions. After this, core 0 executes the next N instructions of the application and core 1 the same next N instructions. Later on, core 0 will execute the third set of N instructions of the program and so on. The purpose of this scheme is to replicate the execution on a given number of cores; this organization is very helpful for investigating reliable microprocessors.

The intention of replicating code execution is to catch up when an execution error occurs on the leading core and being capable of detecting it on the trailing core to correct the execution of the leading core. Figure 2 represents an organization with this scheme.

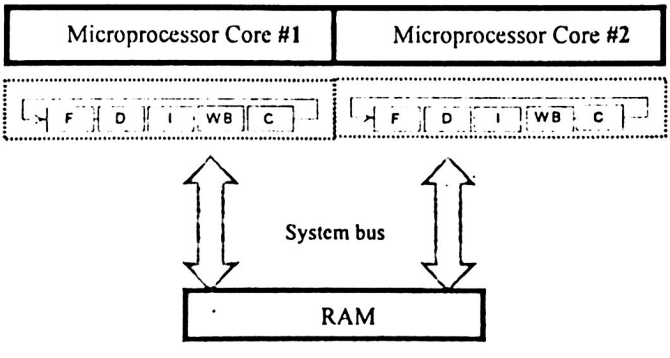


Fig. 1. Free running multiprocessor scheme. N microprocessor cores execute simultaneously without synchronizing

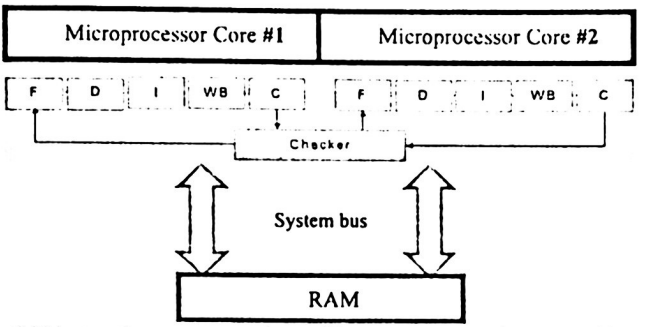


Fig. 2 N instructions per core scheme. The same program is executed by two microprocessor cores each executing the same chunks of N instructions

3.3 Loop-based scheme

Another scheme to provide a multiprocessor simulation environment is to replicate the uniprocessor resources by declaring arrays instead of single variables. However, comparing this approach with the previous two we noticed the same result would be produced without modifying all source code files. Therefore, maintainability and functionability would be improved. We also decide not to use this approach to keep instruction-level execution synchronization control on cores.

4 Software Implementation of Multiprocessor Schemes

In this section we discuss how we implemented the multiprocessor schemes on a base superscalar microprocessor simulator to produce a reliable simulator for on-chip multiprocessors. Our goal is to demonstrate that this is an easy-to-adapt architecture-independent mechanism for any uniprocessor simulator which provides a more scalable and flexible architecture level analysis of uniprocessor details (like bus traffic, cache sizes or registers characteristics), unlike other multiprocessor simulators that focus their analysis details on the inter-processor communication. We also believe this approach could be considered as a good option when other multiprocessor simulators like SimpleScalar Multiprocessor Simulator are found to be not available.

The operating system we use is Linux. The base simulator has a binary file that is assigned to an operating system process. By replicating the running process of the simulator [1] at the beginning of the simulation, two or more microprocessor cores would be easily emulated with very few changes in the original code. The proposed methodology consists on generating multiple processes of the same executable through *fork()* calls to the operating system.

The simplicity of this implementation keeps the functionality and performance of the simulator and provides a scalable solution. By just making *n* calls to *fork()* n CPU cores will be running the same application.

4.1 Free running implementation

The free running implementation requires no synchronization, since every core executes instructions freely. A shared memory implementation of this scheme would require declaring shared memory variables and the addition of cache coherence protocols like [7] or [8]. Our initial approach was to initially explore a non-shared memory multiprocessor organization.

4.2 N instructions per core implementation

A more common multiprocessor organization has shared memory among the available CPU cores. To implement this scheme on the simulator we designed the semaphores described in [9] in order to synchronize the cores and declared shared variables that represent shared hardware resources. With additional shared memory this type of organization can be used to model on-chip multiprocessors with soft-error detection and recovery techniques.

The semaphores provide a source to block a microprocessor core until the next one completes a given set of instructions.

Assuming a multiprocessor with two cores, the use of a single semaphore can control their execution, the software implementation of this synchronization scheme is represented in Figure 3.

In Figure 3 the block named Code #1 represents the instructions to be executed by the core 1. The block Code #2 represents the instructions to be executed by the core 2. The execution of Code #2 depends on whether core 1 has completed executing Code #1; if it has not, then core 2 will be stalled until core 1 completes its work. Similarly, Code #1 depends on the execution of Code #2 on the next round.

To synchronize multiprocessor cores at the instruction level, we set the control barrier before the commit stage of the pipeline in each core. The code fragment shown on Figure 4 demonstrates this idea on software.

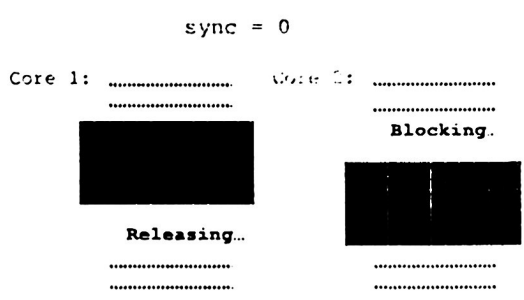


Fig. 3 Semaphore-based synchronization mechanism

```
//instruction pipeline
begin
if (pid == MICRO_0) P(SEM0);
else P(SEM1);
instr_commit();
if (pid == MICRO_1) V(SEM0);
else V(SEM1);
...
instr_issue();
...
end
```

Fig. 4 Software implementation of a synchronized pipeline

Notice that the previous code could also be implemented by replacing the conditional sentences by a hash table to store semaphore IDs and using every CPU core ID as the key to access the table, providing a scalable scheme up to *n* microprocessor cores.

A hardware implementation of a synchronized pipeline would look like that shown in Figure 5.

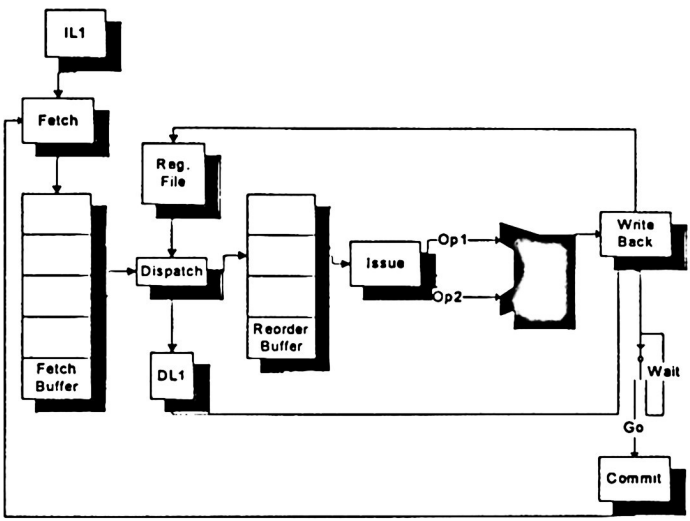


Fig. 5 Hardware implementation of a synchronized pipeline

On Figure 5, the semaphore acts like a double pole switch. On the GO position the pipeline allows the flow of data from commit to fetch. On the STOP/WAIT position the pipeline is stalled until the controlling core executes its corresponding set of instructions. This synchronizing mechanism allows multiple cores to work coordinated.

5 Experimentation and Results

To validate our methodology we executed simulations with a subset of SPECint2000 on our multiprocessor simulator derived from the superscalar microprocessor simulator [1]. For all the benchmarks listed in Table 1 we simulated 100 million instructions.

Table 1. Benchmarks and dataset inputs used for simulations

Benchmark	Input	Benchmark	Input
Bzip2	Input.graphic	**vpr**	net.in, arch.in, place.in
Gcc	integrate.i	**crafty**	crafty.in
Gzip	Input.graphic	**gap**	ref.in
parser	ref.in	**mcf**	inp.in
twolf	ref	**perlbmk**	perfect.pl

We first simulated a superscalar uniprocessor, which characteristics are listed on Table 2. These characteristics are similar to those found in a modern microprocessor like [10]. Then, we simulated a dual-core multiprocessor to verify its correctness.

Table 2. Configuration characteristics of superscalar microprocessor

Component	Config.	Component	Config.
Fetch width	4	L1 D-cache	16KB, 4-way
Branch prediction miss penalti	3 cycles	L2 D-cache	256KB, 4-way
Branch predictor access time	1 cycle	I-Cache	16KB, direct
Branch predictor type	Bimodal	Mem. Latencies	18, 2 (cycles)
Branch prediction table size	2048	Bus width	8
Two-level branch prediction table conf.	1 1024 8 0	I-TLB	4KB
Call/return stack size	8	D-TLB	4KB
BTB size and associativity	512 4	TLB miss penalty	30 cycles
Dispatch width	4	Int ALUs	4
Issue witdth	4	Int Multipliers	1
Write back width	4	Memory ports	2
Instruction window size	16	FP ALUs	4
Load/store window size	8	FP Multipliers	1

The results of the simulations are shown in Figures 6 to 9. Figure 6 shows the performance rate of the uniprocessor as the number of committed instructions per clock cycle (IPC). Figure 7 shows the miss rate of the first level data cache on the baseline superscalar uniprocessor. The miss rate is defined as the ratio of number of misses among all the requests to the first level data cache. Figure 8 shows the performance rate of each processor core on the on-chip multiprocessor. Figure 9 shows the miss rate of each of the first level data caches in the multiprocessor. As can be seen the results on Figure 6 and 8 are identical as those in Figures 7 and 9. As expected, our results were equal, because the characteristics in the core processors are equal to those of the uniprocessor and each core in the multiprocessor executes the same benchmark. We are assuming ideal inter-core communication conditions. A communication model with no ideal conditions would impact in the multiprocessor's performance. However, the miss rate of the first level data cache would not be affected, because we have defined a first level data cache per core. We are aware of the fact that considering a shared second level data cache would impact the first level data cache miss rate, so we will explore this in future work. As stated above, the purpose of this multiprocessor configuration is to provide execution redundancy in order to achieve multiprocessor characteristics like reliability, security, redundancy and soft-error detection and correction using this methodology as the main baseline simulation tool used in [2].

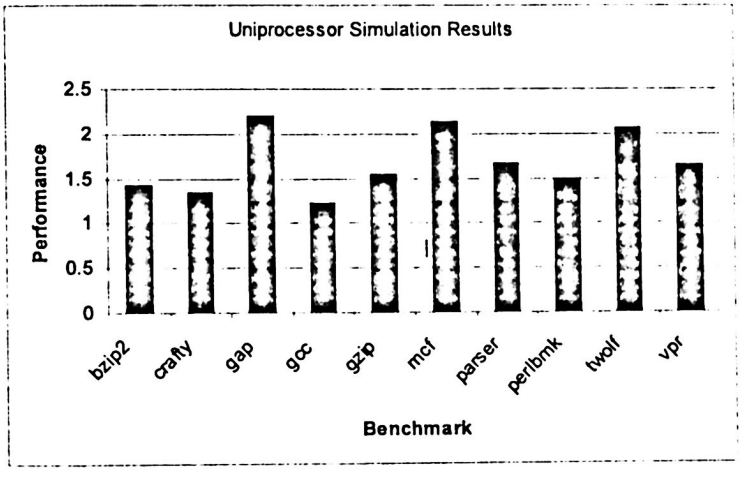


Fig. 6 Performance results of superscalar microprocessor (number of committed instructions per clock cycle)

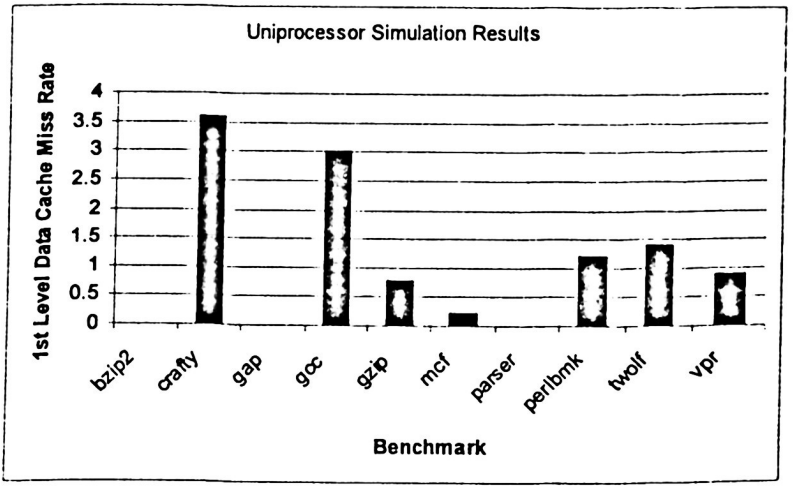


Fig. 7. First level data cache miss rate (%) results of superscalar microprocessor (ratio of misses among all the requests to the first level data cache)

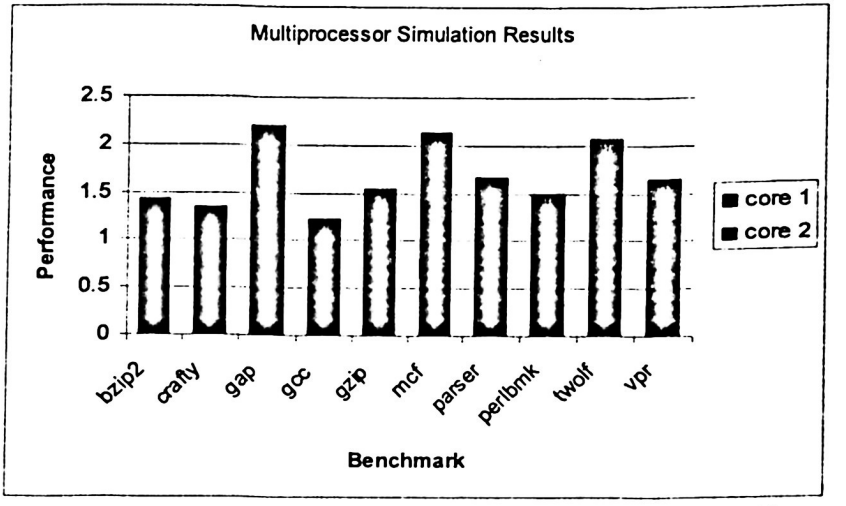


Fig. 8 Performance results of cores in dual-microprocessor (number of committed instructions per clock cycle)

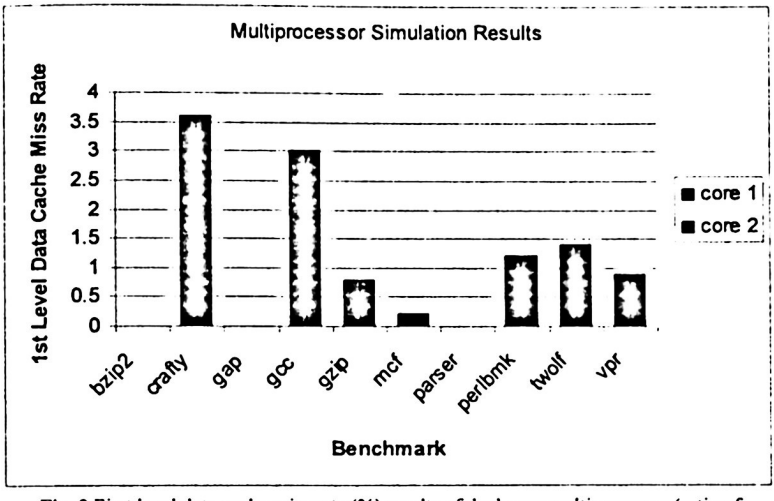


Fig. 9 First level data cache miss rate (%) results of dual-core multiprocessor (ratio of misses among all the requests to the first level data cache)

6 Conclusions and Future Work

We have proposed a methodology for on-chip multiprocessor design using a uniprocessor simulator as a baseline and operating system process replication functions along with shared memory variables. Our methodology is simple and useful to explore new architectural ideas for on-chip multiprocessors. We demonstrated that the multiprocessor simulator provides the same results as the uniprocessor simulator, but on multiple processor cores under ideal inter-core communication conditions, which is a very useful technique for investigating soft-error detection and correction as well as for exploring fault-tolerant multiprocessors.

The next step in our research is to investigate shared resources in hardware and to evaluate new sharing policies among them.

As shown in [2], we have also proved a technique for the design of fault-toleratn on-chip multiprocessors and we have found this methodology very useful. It has allowed us to explore multiprocessor characteristics like reliability, security, redundancy and soft-error detection and correction.

7 References

[1] Larson, E., Chatterjee, S., Austin, T. MASE: A Novel Infrastructure for Detailed Microarchitectural Modeling. On ISPASS 2001. Tucson, Arizona.

[2] Mizan, E. and de Alba, M. "Fault-Tolerant CMP Design Using a Write Cache Checker", on DSN 2005 The International Conference on Dependable Systems and Networks". Yokohama, Japan. July 2005.

[3] Pai, V. S., Ranganathan, P. and Adve, S. RSIM: Rice Simulator for ILP Multiprocessors. http://softlib.rice.edu/rsim.html

[4] Matloff, N., Rich, K. Mulsim: Multiprocessor Simulator. http://heather.cs.ucdavis.edu/~matloff/MulSim/MulSimDoc.html

[5] Sunada, D., Glasco, D. and Flynn M. ABSS: SPARC multiprocessor simulator. In proceedings of the 8th Workshop on Synthesis and System Integration of Mixed Technologies (SASIMI '98).

[6] Chen, X. Hong, J. Gao, Y. Li, X. Zheng, S. Design and Implementation of Multiprocessor Simulator Simdsm, Minimicro Systems, Shenyang, 2000. Vol. 21; Part 2, pp. 186-189.

[7] Archibald, J. and Baer, J.L. Cache Coherence Protocols: Evaluation using a Multiprocessor Model, ACM, Trans. On Computer Systems 4:4, 1986, (November), pp. 273-298.

[8] Agarwal, A., Simoni, R., Hennesy, J., Horowitz, M. An Evaluation of Directory Schemes for Cache Coherence, In Proc. Of the 15th ISCA, June 1998, pp. 280-289.

[9] Gómez, C., R. Memoria compartida, semáforos y colas de mensajes. Tecnológico de Monterrey, 2002, pp. 16.

[10] Hinton, Glenn (et. al.). The Microarchitecture of Pentium® 4 Processor.

[11] The Simplescalar Web Site. http://www.simplescalar.com.

Register file optimization for low power using the zero detect technique.

Moisés A. Zárate S. [1], Luis Alfonso Villa Vargas [2]
Oscar Camacho Nieto [3], Osvaldo Espinosa Sosa [4]

[1,3,4] Centro de Investigación en Computación CIC-IPN
Av. Juan de Dios Bátiz esq. Miguel Othón de Mendizabal S/N
Col. Nueva Industrial Vallejo, C.P. 07738. México, DF.
TEL. 5729-600 Ext.: 56519
moizarate@correo.cic.ipn.mx, oscarc@cic.ipn.mx, espinosa@cic.ipn.mx
[2] Instituto Mexicano del Petróleo IMP
lvilla@imp.mx

Abstract. In this paper is presented an improvement in the reduction of the energy consumption for the register file of a superscalar processors with the technique of Zero Detect (detection of chains of bits equal to zeros), for which the registration is divided in chains of 32, 16, 8 y 4 *bits* (*word, integer, byte* and *nibble*), with the intention of reduce significantly the energy that wastes away when having a read or write access to the register file of this type, since the technique diminishes the transfers of chains of bits for each one of the accesses already mentioned. The carried out tests were made with the simulator Simplescalar V3.0 and with Spec95 benchmarks for each one of divisions of chains, being obtained savings in the energy consumption of up to 50%.

Words key: *Zero Detect*, Register files, Energy consumption.

1 Introduction

As it is known, the register file is a fundamental component of superscalar processors which consume a considerable amount of energy when processor needs to access to those file contents. For that reason, several techniques have been created in order to improve the performance increasing the hardware, but to cause of this, it has a higher energy consumption which is wanted to be diminished without affecting the performance [7]. One of the most important techniques on energy reduction is when having a line bit to indicate zero chains [2]. Others have sectioned those register files with multiple sub-banks in order to access every time only one bank which has an active bit line for each one of them [6]. Other techniques had a little transcendence such as separation of a register zero [1, 5], which consists on detecting single registers that contains zero values within operations that are carried out inside the functional units, which is very significant since when having an access to read the register, a lot of energy is wasted because bits are only read with zero values. It is also important to mention that others have proposed implementation of techniques that reduce multiple access that they are had when making a reading or writing to the register file [4, 8], achieving with it the energy reduction or increment in performance.

Techniques for detection of zeros in register files, caches [3], etc. are very importan since thank to them good results have been obtained for energy reduction inside superscalar processors and these techniques are susceptible of continuous improving.

2 Proposal

According to the previous necessities, we create our technique called Zero Detect which is an extension of techniques mentioned previously. It is important to mention tha from the total of registers that are used for the operations inside the functional units, can observe that 49% of them are similar to zero. As it is known, all registers that are use inside the architecture are physical, and they were born with zero value from renaming the logical registers with zero register detections were made by each reading access an writing. They can also meet groups of bits with zero value inside the registers with valu different from zero, for this reason the fundamental part of this technique is the detectio of chains of nibbles, bytes, integers and words equal to zero, for registers equal an different from zero.

Once obtained these detections of zero chains, a flag bit is placed by each one of them the activation of this bit represents a saving in the energy consumption, because only reading of it is made, avoiding the reading of all the bits of the zero chain.

2.1 The major contributor device that contains registers and detections of zer registers in readings and writings.

Inside the architecture the biggest component that offers registers to the arithmetic logic operations that are executed inside the functional units is the register file and writeback stage (fig. 1). Of these two, we only focus on the register file, which is biggest component that gives of registers, inside it only physical registers exist, of which it has been shown that the mentioned contribution is obtained from the physique ones because almost all the operations depend on the renaming of the logical, therefore, it was spoken of the bank of physical registers, that which doesn't mean that one had a division of the register file in physical and logical, it is only a short term that we will use understand that we only focus ourselves on the physical registers.

Inside all the arithmetic logical operations that were carried out with the physical register file, they were had to make two reading accesses (fig. 2), to be able to know values of the register sources and a writing access (fig. 2) to place the result of operation inside a register destination. With that mentioned, detections of zero registrations were made every time that was carried out a reading access or writing.

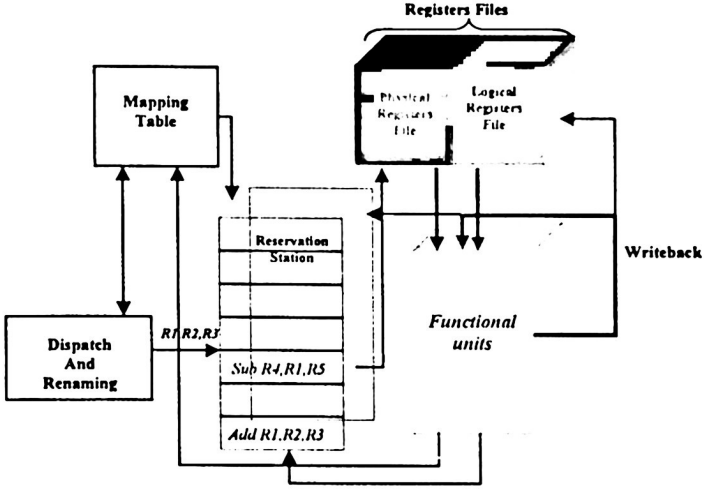


Fig. 1 Stages of superscalar processor, standing out those that contain registers.

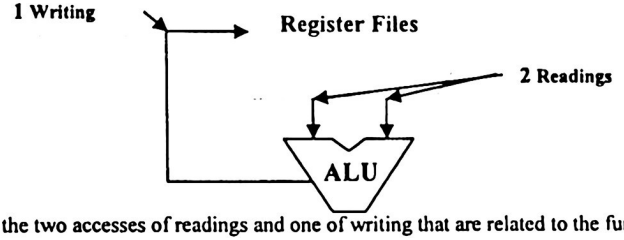


Fig. 2 It shows the two accesses of readings and one of writing that are related to the functional unit

2.2 Chain division inside the registers equal to zero and different from zero.

With the mentioned previously, once the results of all the registers are obtained with same values and different from zeros for each reading access and writings, were carried out divisions in groups of chains of nibbles, bytes, integers and words (fig. 3) inside them, which are good for us to have data but exact on the number of zero chains that use a superscalar processor. With these data is proposed the Zero Detect technique, which is explained next.

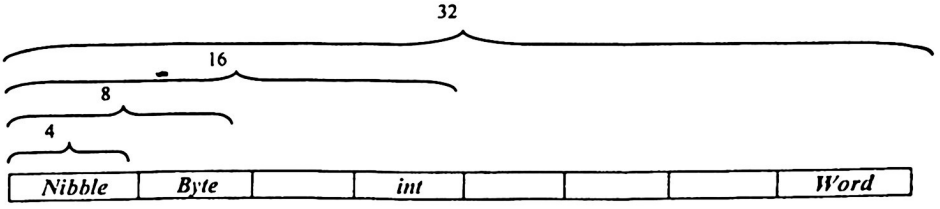


Fig. 3 Chain divisions in 4, 8, 16 and 32 bits (nibble, byte, int. and word) that were made insic each register.

2.3 Zero Detect Technique

The Zero Detect technique consists on placing a flag bit for each nibble chain, byte, and Word. The placement of this bit inside of each one of the registers increases a colum. for each chain division, like you can appreciate in the figure 4.

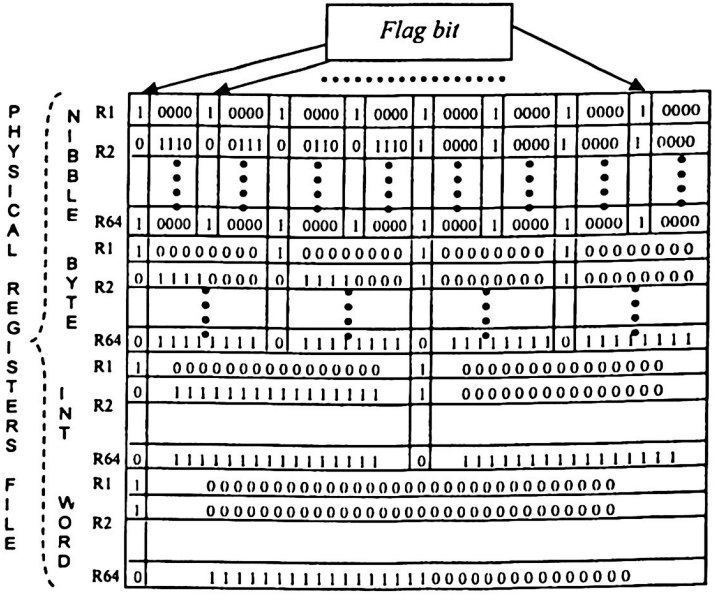


Fig. 4 It shows the Zero Detect technique applied to chains of nibbles, bytes, integers and words, which is activated when the bit flag contains a value of one, in the event of containing zero value it is not activated.

The columns containing the flag bit were the first ones in being activated to be able to know their value, which are activated when it contains a 1; this is good since alone the reading of that flag bit would be made avoiding the qualification from the whole chain to which makes reference this flag. However, if the flag bit is different from zero one would have to make the reading of the whole chain causing an extra expense of energy, because one makes the reading of the bit more the reading of the contained bits in the chains.

With the above mentioned we can observe that if the value of the bits flags were similar to 1, the saving of readings of 32 bits was obtained (words), 16 bits (integers), 8 bits (bytes) and 4 bits (nibbles), but when the value of the bit was 0 they were had to enable the chains to be able to carry out the readings of them in a second cycle, that which resulted in the reading of 33 bits (words), 17 bits (integers), 9 bits (bytes) and 5 bits (nibbles). This indicates a lost one in the saving of readings and in the time of execution due to the second cycle, but even so you speculates that the technical Zero Detect presented an energy saving.

The explanation is that at level alone transistor is wasted away energy of the activation of the 6 transistors of the flag bit and it was avoided that the 192 (words), 96 (integers), 48 (bytes) and 24 (nibbles) remaining transistors were activated.

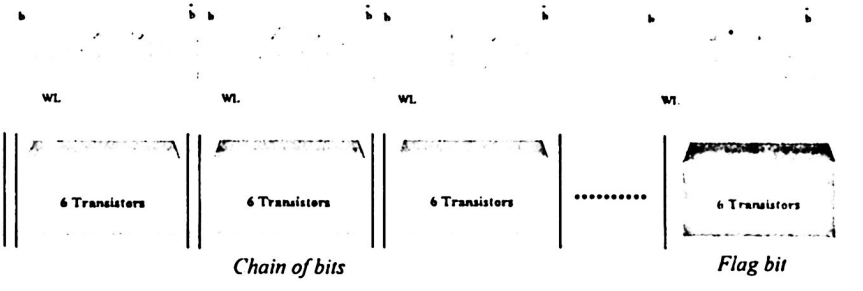


Fig. 5 It shows the number of transistors of the flag bit and of the bit's that contain the chains.

3 Methodology

To be able to know with detail the benefits that can be carried out inside a processor superscalar Alpha-21264 we use Simplescalar V3.0 simulator, because the physical development of a new processor represents a very high cost, which would have to be carried out for each experiment. The modifications that were made inside the simulator to be able to know the stage where they used most of the registers, the detections of registers zero, the divisions of chains and the implementation of the technical Zero Detect, was

proven by means of the work loads SPEC95, which are programs that were compiled the simulator.

Figure 1 is the graphic representation of the simulator with the first modification mad which consists on placing the accountants A and B inside the issue functions and accountant C in the writeback stage to know the stage where is wasted away most of registrations. In the table 1 descriptions of the benchmarks of the SPEC95 are presented.

Name of the *Benchmark*	Type of SPEC95		Description
	Integer	*Float Point*	
Compress {test}	✓		Compresses and uncompresses file in memory
li {*test*}	✓		Lisp interpreter
ijpeg {*test*}	✓		Graphic compression and decompression
Go {test}	✓		Artificial Intelligence, plays the game of Go
Vortex {*test*}	✓		A database program
M88ksim {*test*}	✓		Moto 88K Chip simulator, runs test program
gcc {test}	✓		New Version of GCC, builds SPARC code
perl {test}		✓	Manipulate strings (anagrams) and prime numbers in Perl.
applu {test}		✓	Parabolic/elliptic partial differential equations
apsi {test}		✓	Solve temp., wind, velocity and dist. of pollutants
turb3d {test}		✓	Simulate isotropic, homogeneous turbulence in cube

Table 1. Benchmarks and their description.

The benchmarks exposed in the publication are apsi, turb3d, go, applu, and ijpeg, due the space reduction that one has for the publication, however these benchmarks chosen to be representative.

4 Results

4.1 Analysis of the biggest consumer of registers inside the architecture.

The obtained results from simulations are shown in the figure 6, in the one one observes that the biggest consumer of registers inside the architecture of the process is the registers files of physical with 64%, for this reason we only focus ourselves to detection of registers zeros that they are obtained by each reading or write access in process. The writeback stage used 36% which almost represents 50% of what is used the physical register file.

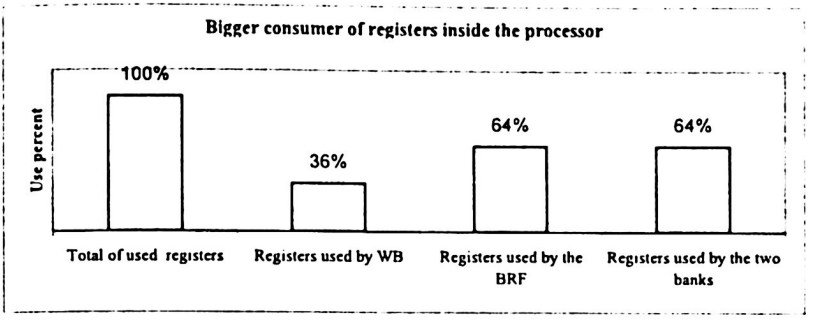


Fig. 6 It shows: 1. - Registers used by Write Back (they are all the values that are expected by a register, to be able to carry out the execution of an operation), 2. - Registers used by the registers files of physical (they are all the registers of the renaming stage that are read and they are written to liberate dependences), 3. - Registers used by the registers files of Logical (they are all the source registers inside the processor), 4. - Registers used by the banks (it indicates the total of registers in charge of inside the physical and logical register files), 5. - Total of used registers (it indicates the total number of registers that are used inside the processor).

4.2 Evaluation of zero registers detection in each reading access or writing.

The results of the detections of registers with value similar to zero in accesses of readings as well as in writings were obtained with the required simulations, which are shown in the figure 7. In her it is observed that 48% of accesses of readings were registers with value similar to zero and for the case accesses of writings one had 49% of detections of zero registers.

As it was observed, almost 50% of the registers that were used inside the architecture contained a value similar to zero, with that which several techniques could be implemented to be able to have a better performance and/or an energy saving, in our case we intended dividing the same registers and different from zeros in groups of chains of nibbles (4 bits), bytes (8 bits), integers (16 bits) and words like it is shown in the figure 3, which gives us an improvement of techniques for detections of zero registers , because it is increased the number of detections.

4.3 Evaluation of different divisions inside the zero registers.

The obtained results from different simulations are shown in figure 8, showing that 43% of readings and 42% of writings were chains of words similar to zeros. As we can observe it has a considerable quantity of chains of 32 zero bits inside the instructions to execute inside the processor. The percentages of 55% of readings and 53% of writings are also

shown, they were integers chains similar to zeros, which indicates a higher detection chains zeros in comparative with the word's chains. However it was obtained that 63% readings and 59% of writings were chains of zero bytes, giving a higher detection of zero chains as a result in comparative with words and integers. In the case of detections chains of nibbles with value similar to zero, 72% is shown for the case of readings and 67% for the case of writings. However it was obtained that 63% of readings and 59% writings were chains of zero bytes, giving a higher detection of zero chains as a result comparative with words and integers. In the case of detections of chains of nibbles with value similar to zero, 72% is shown for the case of readings and 67% for the case writings. With this it was demonstrated that when having a smaller division of word, byte and integer are obtained higher detections of chains with value similar to zero.

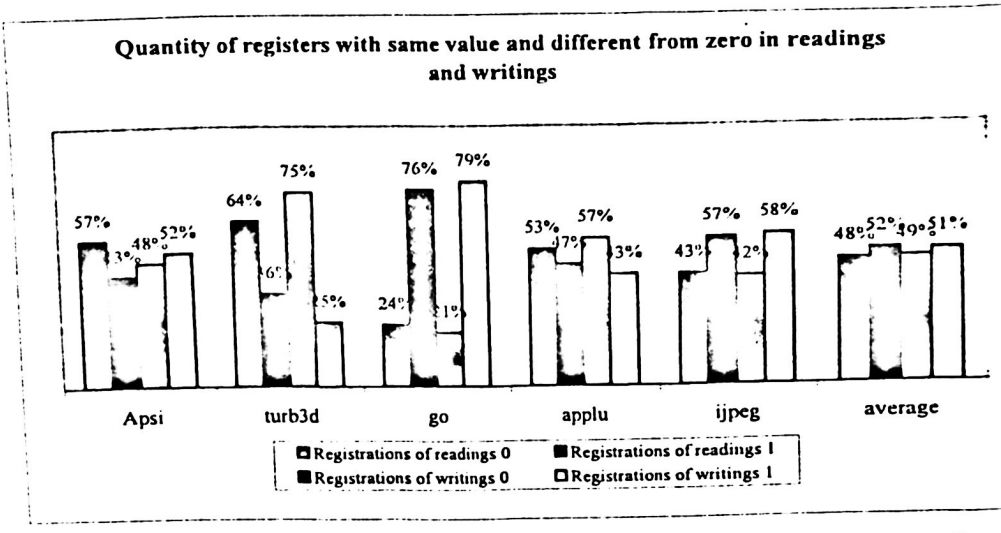


Fig. 7 Graph showing the percent of total registers with value similar to zero and one in reading and writing accesses.

Divisions of zero chains of are an important base to optimize the register files and reduce the energy consumption, due to this, the zero detect technique arises, which consented in adding a bit for each chain, like it is shown in the figure 4. This bit activated when a presented chain of zeros is detected a value similar to one, otherwise preserve its inactive state and the value that it shows are similar to zero.

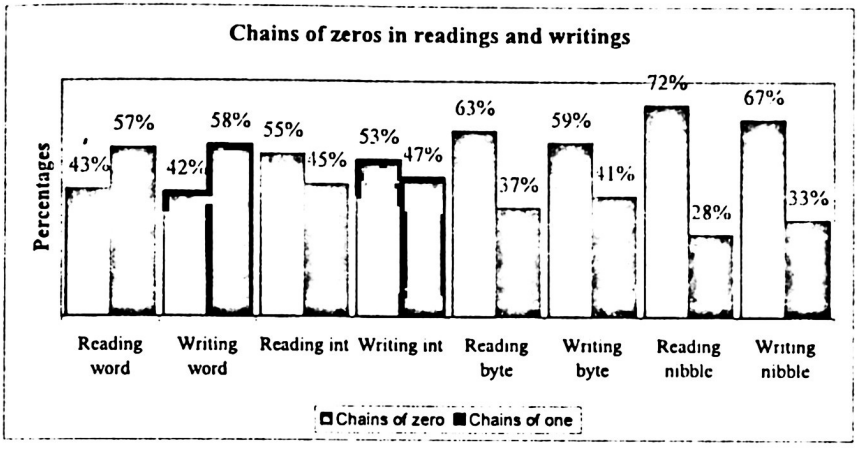


Fig. 8 Graphic showing the quantity of chains of nibbles, bytes, integers and words with value similar to zero in readings and writings.

4.4 Evaluation of the Zero Detect technique with different divisions.

As it was mentioned previously the Zero Detect technique consists on adding a flag bit to all the chains of words, integers, bytes and nibbles, for which, the results they generate are shown in the figure 9, alone they are of the activation of the flag bit. In the figure 9, it is observed that 40% of readings and 39% of writings were word's chains similar to zero. It is also shown that 49% of readings and 47% of writings were integer chains similar to zero. With the previous results you speculate that almost one had 40% of energy saving for the case of word, but for the case of integers it was almost 48%, indicating a higher optimization in the reduction of energy consumption.

For the case of bytes the graph shows that 50% readings and 46% of writings were byte's chains similar to zeros. The results showed that you almost speculates 50% of energy reduction, that which is better in comparative with word and int. Lastly in the figure 9, it is shown that 47% of readings and 42% of writings were nibble's chains similar to zero, with these values one had an optimization of little but of 45% of energy saving, that which was smaller than the int. detections and byte, but even so it was good.

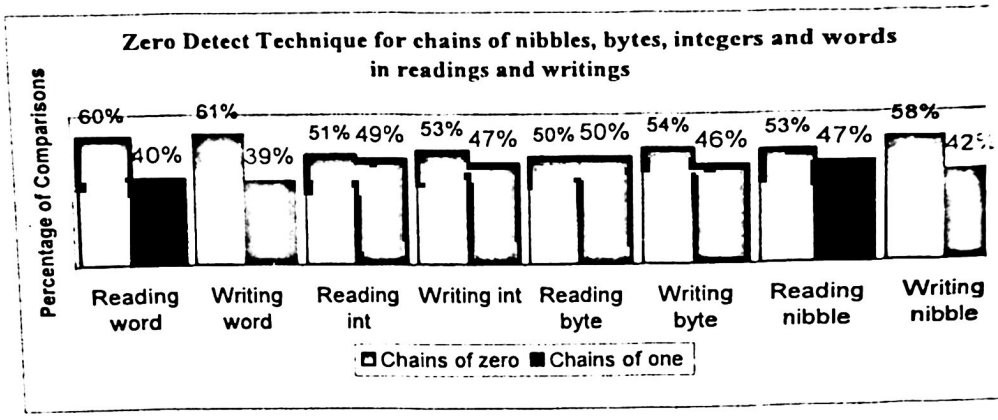


Fig. 9 Graph that shows the behaviour of Zero Detect technique for the chains of nibbles, bytes integers and words with the same and different value to zero in readings and writings.

4.5 Comparison between the renaming of a normal superscalar processor Vs. modified version with Zero Detect technique.

With the results mentioned previously it was proven that Zero Detect technique good enough, and although the performance penalties with 5% like it is shown in fig 10, we can speculate that register file was optimized in general with almost 50% of ener saving, this demonstrates that techniques for zero detections are a very important base performance and energy consumption saving.

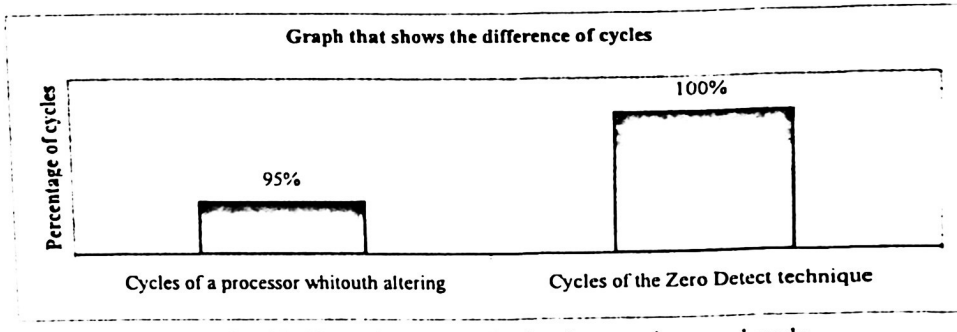


Fig. 10 Graphic illustrating the penalization due to an increased cycle.

Conclusions and future Works

All the modifications that are made inside a superscalar processor are very expensive, because the prototype doesn't work in the first experiment and this makes necessary to carry out more experiments causing a higher consumption of physical materials, without forget to take into account that sending to implement the design is very expensive. To be able to save costs in all the tests that were carried out, we already occupy one of the standard simulators called Simplescalar V3.0, which demonstrated to be efficient and able to create several models that give us an improvement in the optimization of the register bank since the result is a reflection of real tests under an energy consumption scope, therefore, we concluded that Zero Detect model created with the simulator was good since it helps to optimize the physical register file with almost 50% of energy saving, what indicates at transistor level that only it wasted away energy when being activated the 6 transistors of the flag bit in all the columns belonging to each chain, despite of penalization of 5% that we obtained in the performance.

With the percentages obtained due to optimization of energy consumption that Zero Detect technique presents, it opens the possibility of the creation of new models that allow similar consumption savings but also without having the penalization of performance. We could obtain also much precise data with software like TANNER o PSPICE in order to implement it in a Programmable Logic Device (PLD) and simulate it in power simulators.

References

[1] Jessica Hui-Chun Tseng. Energy-Efficient Register File Design. Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of Master of Science in Electrial Engineering and Computer Science at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY December 1999

[2] R. Balasubramonian, S. Dwarkadas, and D.H. Albonesi. Reducing the complexity of the register file in dynamic superscalar processors. In *MICRO-34*, December 2001.

[3] Luis Villa, Michael Zhang, and Krste Asanovi´c. Dynamic zero compression for cache energy reduction. In *33rd International symposium on Microarchitecture* , Monterey, CA, December 2000

[4] A. Seznec, E. Toullec, and O. Rochecouste. Register write specialization register read specialization: A path to complexity-effective wide-issue superscalar processors. In *MICRO-35*, Istanbul, Turkey, November 2002.

[5] G. Kane and J. Heinrich. *MIPS RISC Architecture (R2000/R3000)*. Prentice Hall, 1992.

[6] J.-L. Cruz, A. Gonzalez, M. Valero, and N. P. Topham. Multiple-banked register file architectures. In *ISCA-27*, pages 316–325, 2000.

[7] Teresa Monreal Arnal. Technical Hardware to Optimize the Use of the Registrations in Processors Superescalares. University of Zaragoza, Department of Computer science and Engineering of Systems, June of the 2003.

[8] J. Tseng and K. Asanovi´c. Energy-efficient register access. In *Proc. XIII Symposium on Integrated Circuits and Systems Design*, Manaus, Brazil, September 2000.

Digitally Programmable Analog Instrumentation Modu for Processing Vital Signs

Sandra Díaz Pier, Marcos Rubén de Alba Rosano

Tecnológico de Monterrey
Departamento de Ingeniería Eléctrica y Electrónica
Atizapán, Estado de México, México, 52926
{A00455576, marcos.de.alba}@itesm.mx

Abstract. Current data acquisition systems have a strong trend on transforming physical signals retrieved from the environment to a digital form regardless of the loss of information on acquired data due to the conversion process. Purely analog processors were the beginning of computer sciences, but they were difficult to manipulate and program so they were left aside even when they had very positive characteristics, like high-quality signal and fast response. In this paper we describe a digitally programmable analog module for signal processing vital signs, which goal is to provide high-quality digital representation of analog signals into an entire user-interactive device. The module can be programmed by using C language or by a friendly menu. The signal processing on the module has a mode of operation that can be chosen to be purely analog or digital depending on a particular application.

1 Introduction

The world we live in is analog. Through technology, we are able to detect, process manipulate any signal generated by nature. Mainly, for storage and processing purpose most commonly analog signals are converted into a digital format. In this form information can be stored, displayed and manipulated to program several devices actuators to respond on predefined levels of tolerance. As described by Tamcis Roska [1], the ability to store programs and data was the key fact that promoted the stron migration of processing to a digital form. However, purely analog signals that transformed into digital format lose information, which could be important in som implementations. The world is so digitally focused that many problems that could solved by using simple analog circuits are solved by using high-level programming more complex digital circuits.

The digital programming world has extended into hardware systems design to mal implementations more versatile which accelerate the design process. Field Programmabl Gate Arrays, for example, are being used more and more every day in computationa

embedded and electronic designs. Current trends indicate that these will be the future for customizable systems development. They are accessible from many levels of programming and have great versatility for emulation.

Our motivation for developing this project is to design a device to improve the way analog signals, like metabolic signals of organisms and environmental variables like temperature, humidity, light and motion, among many others, interact with electronic and data acquisition tools to create customized instrumentation applications. Our approach is to propose an instrumentation module capable of manipulating several analog signals by programming. Therefore, it can be set up for different applications by software. Although there are a few test boards and integrated circuits that can customize analog components by software, most of them are specialized in automotive [2] and industrial applications, with characteristics that are far from being suitable for bio-electronic instrumentation like safety isolation stages.

In this paper we will describe the implementation of a programmable analog instrumentation module. It is designed to be programmed using a software interface within an embedded microprocessor in a FPGA.

The rest of the paper is organized as follows. On the second section we will cover previous wok on the area. On the third section described the design of the proposed module. On the fourth we discuss results that validate the design. On the fifth we derive some conclusions. Finally, we provide some guidelines to the next steps of our work.

2 Previous Work

Computational sciences were originated with the purpose of emulating and improving processing of data of nature source. Rapid technological development during the last decades has brought systems capable of computing huge collections of data in fractions of a second. Current technology has the advantage of storing and retrieving great amounts of data in fractions of a second. Hybrid systems, that is those that manipulate analog and digital signals are very useful for several applications and result in the generation of extremely efficient systems.

Some of the most interesting computational trends include: the development of molecular electronics, analog cellular computers and the interconnection of organic elements with silicon and data processing devices to achieve neural and biological functions. Mixing organics with electronics will be more frequently used in the near future to store information, to create parallel manipulation of data and to integrate new sensitive components and pathways in a similar way our brain and body do. This allows such capabilities to be embedded within the main structure of the material used to build the device [1]. These research topics compose a strong goal in computer sciences, because they can reduce the gap that separates the way nature generates, stores, manages and produces signals from the way humans emulate them. Important work is being developed on these areas. Results obtained so far indicate that remarkable performance can be

achieved using these novel techniques that in the near future might promote the evolutio of computational sciences into something very different from what we use today.

Beyond the bio-electronic interaction at molecular and silicon integrated level, anothe kind of relationship is in development between biological and artificial organisms. new relationship consists in sharing signals to control activities at both ends transforming actions from one world to alter conditions in the other and vice vers Signals produced by metabolic and physiological activities in animals are being use today to control computers, machinery, robots, therapy aid equipment [3] and othe apparatuses [4], while sensors on those devices can be used to indicate operatin conditions to the living being. The virtual reality field has undoubtedly turned its eyes human-computer interaction at hardware level to improve the inmersive characteristics simulations. Virtual reality developers used to be concerned only about software utilitie and ways to make more realistic graphics but today some of them have realized merging perceptive capabilities of humans with software design significantly improve the interaction between living organisms and virtual scenarios [5].

3 Digitally Programmable Analog Instrumentation Module

The system we implemented consists in an analog module conformed by processing channels, an FPGA with a MicroBlaze embedded microprocessor configure for control and configuration of the entire device and flexible output stage that provide several options to output the retrieved signal and communicates with other equipmen units. The analog unit requires two 9-volt batteries and is electronically isolated wit optocouplers from the FPGA (input and output). It also includes safety requirements bio-electronic interfaces.

Next we will describe every block of our design in detail.

3.1 Description of the Analog Unit

Four data acquisition channels, as described before, constitute the analog unit. Two those channels are focused for instrumentation applications and consist in differentia input, high-impedance, low-noise amplifier, a band-pass filter, a general-purpos operational amplifier, and an integration/inverter block. Each of those components digital potentiometers, used to establish the value for gains and cutoff frequencies and enable/disable pins to redirect the signal through the path that better satisfies the user needs. Digital potentiometers were used because they allow digital command outputs the MicroBlaze processor to be translated into an analog value, in this case resistance. Figure 1 is shown the block diagram of a differential input channel.

3.2 Noise handling

In this analog design we prevent noise from altering the signals we acquire by including bypass capacitors between positive voltage and ground and between negative voltage and ground in most of our principal integrated circuits. We also use bypass filtering between +5V voltage and ground in the Analog to Digital converter. We use tantalum capacitors to improve the noise filtering in our bypass arrangements because they are suitable for high speed and high frequency applications. In our prototype printed circuit board (PCB) the grounding covers an acceptable percentage of the plaque and has a surrounding shape to help isolate the entire circuit. Very susceptible input pins, like amplifiers' inverting input, have been surrounded and isolated from the rest of the board (from the non-inverting inputs in the specific case of amplifiers) by placing ground lines around them. In our future work we will consider other noise reduction techniques like shielding to reduce the effects of other electromagnetic sources of noise.

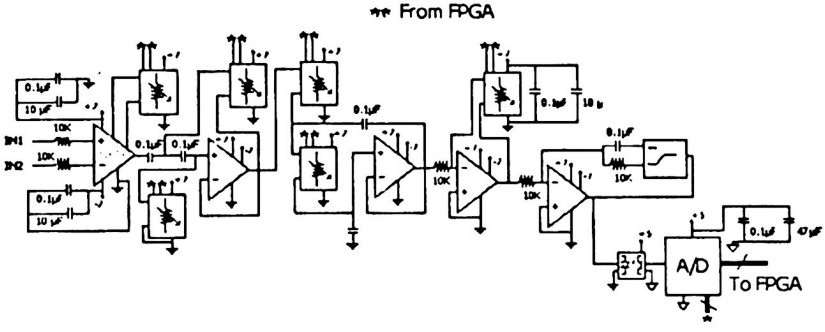


Fig. 1. Block diagram of Differential Input Channel.

The other two channels are targeted for use with sensors or with non-differential input sources. They consist on rail-to-rail operational amplifiers, which gains are also controlled using digital potentiometers. Figure 2 shows the block diagram of a non-differential channel.

3.3 Digital unit: Spartan 3 FPGA and MicroBlaze Embedded Microprocessor

Every input channel has an analog-to-digital converter at the end of the processing flow. In the block diagram this is shown as the connector to the embedded microprocessor within a Spartan 3 FPGA. As we stated before, some analog signals need to be treated in its pure form to preserve all available information, but we think it is important to be able

to interact with other digital devices too. Inside the FPGA resides the structure MicroBlaze processor with customized output and input ports, a timer is dedicated display the signal when graphic mode is selected and controllers for memory and inter buses.

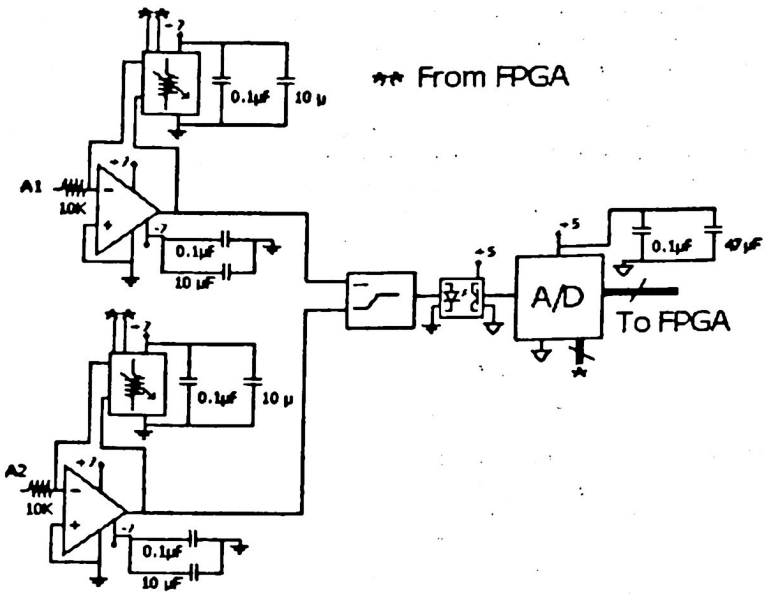


Fig. 2. Block Diagram of Non-Differential Channel

Code in C++ language was developed for programming the components of the anal module, as well as to control the input signal, which has been pre-processed by the analo block. The configuration program includes several modes of operation and control for analog unit. The initialization of the system consists on changing the digit potentiometers to obtain predefined primary gains for every channel. The initial selecte channel is differential input channel 1; signal graphs on VGA, the digital mode integration are disabled by default.

The following code corresponds to the initialization of channel 1 of our design. A signal is output on the FPGA's pins to change the value of the digital potentiometers achieve startup configuration.

```
xil_printf("Ajusting Initial Parameters for Analogic
Module... \n\r");

//INSTRUMENTATION1
//Low-freq. counter
reslp[0] = sqrt(1/(pow(freq_lp[0]*3.1416,2) *
0.00000000000001));
goal_clkcount_ad5220[0][5] = (Xuint32)(reslp[0]/RESISTANCE);
goal_clkcount_ad5220[0][6] = (Xuint32)(reslp[0]/RESISTANCE);
//High-freq counter
reshp[0] = sqrt(1/(pow(freq_hp[0]*3.1416,2) *
0.00000000000001));
goal_clkcount_ad5220[0][2] = (Xuint32)(reshp[0] /
RESISTANCE);
goal_clkcount_ad5220[0][3] = (Xuint32)(reshp[0] /
RESISTANCE);
//Instrumental amplifier counter
goal_clkcount_ad5220[0][1] =
(Xuint32)(49400/((gain_inst_amp[0]-1)*RESISTANCE));
//Amplifier counter
goal_clkcount_ad5220[0][4] =
(Xuint32)((gain_gen_amp[0]*10000)/RESISTANCE);

//Generate signal for each potentiometer
for(i = 0; i < 6; i++){
      while(clkcount_ad5220[0][i+1] !=
      goal_clkcount_ad5220[0][i+1])
            clockcount_pots (i+1);
}
```

Once the initialization is finished, a *menu* is displayed on a Communication Software (Hyperterminal) using Serial Rx Tx. Our system has two basic modes of operation: purely analog (selected by default), or analog-digital. On purely analog mode none of its digital treatment capabilities is activated and the user can only adjust parameters on the analog section. On analog-digital mode the user can adjust every component as on the previous operation mode, but the features for digital processing are available: detection of minimum and maximum values, offset nulling, graphic output and latching (many others, like digital filters, could be developed to cover the user's needs due to the flexibility offered by the Microblaze processor) and the output section is fully operational.

One important characteristic of the second mode is that the program is allowed to reduce the gain on the input amplifiers automatically to constraint the signal received as input in the ADC to avoid saturation. Automatic gain modifications are only disabled if purely analog mode is selected. Figure 3 below shows the response of the differential input channel to a mioelectric signal generated by a human biceps muscle.

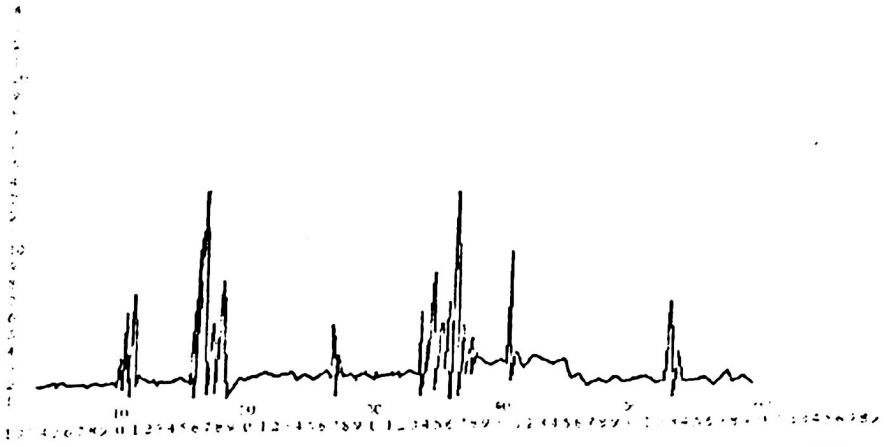


Fig. 3. VGA output of a post-processed mioelectric signal obtained using the differential inp channel and Ag / ClAg electrodes placed on the biceps muscle of a human tester.

3.4 Output stage

Our system has a very accurate post-digital module attached to a digital-to-analc converter with current-variations-based on radio frequency (RF) transmissions, puls width modulation (PWM) and analog voltage level output. The set of output capabilitie for our application are designed to be supportive for wireless and long-wire transmissions, as well as for coupling with other devices, microcontrollers and actuators.

Up to this point we have described an implementation that is capable of processir analog signals responding to constraints specified by the user with software. Then, digitally post-process the signal output by the analog stage, provides a very flexible outp section with current-variations-based and RF transmissions, PWM and analog voltag level. Next we discuss results obtained with our module.

4 Results

We were able to integrate our design and manipulate signals received at external senso and transducers. We noticed some limitations on the detection levels for the digital mod of operation, because we could only work with signals within a 5V limit. Beyond that, system proved to be configurable for amplification gains between 0 and 10,000. filtering stage worked properly above the band-pass bandwidth of 150 Hz. Digit treatment of signal was proven to work as expected for the signals we worked wit

However, more work in refining the design is needed to fully interact with analog signals that can vary from high positive to high negative levels.

5 Conclusions

The design described on this paper represents a very flexible tool for implementing bio-electronic interfaces. It is built to merge the natural and artificial worlds into cooperative inter-feedback applications. We developed an analog unit that can be programmed at high level or configured using *menus* to provide an accurate processing unit. It integrates features from both the analog and digital worlds, trying to create a specific purpose system responding to the emerging development areas of organic-artificial interaction at the macro level.

6 Future Works

We want to extend our work improving the programming limits and components of our design to cover a wider set of applicative options. Later, we want to take our design into an integrated circuit to have a single chip extremely flexible and programmable solution for bio-electronic interface design and implementation.

The future of our project targets the creation of an application-specific integrated circuit (ASIC) which can be used to develop several applications by specifying constraints with software and which can manipulate signals analogically. It can transform analog into digital, if necessary, and include the features needed to merge natural with artificial systems, everything one chip.

References

[1] Roska, T. Analogic Computing: System aspects of Analogic CNN Sensor Computers. 2000 STH IEEE International Workshop on Cellular Neural Networks and Their Applications Proceedings, 2000.

[2] Melexis Microelectronic Systems. MLX90308. Programmable Sensor Interface, 2004.

[3] Upshaw, B.J. Sinkjaer, T. Real-time digital signal processing of electroneurographic signals. Engineering in Medicine and Biology Society, 1994. Engineering Advances: New Opportunities for Biomedical Engineers. Proceedings of the 16th Annual International Conference of the IEEE. Pages. 1346 - 1347 vol.2, 1994.

[4] Diaz, M. and Rudomin, I. Object, function, action for tangible interface design. ISMAR 04. Pages: 106 – 112, 2004.

[5] Díaz, M., Hernández E., Escalona, L. Rudomin, I., Rivera, D. Capturing Water and Sound Waves to Interact with Virtual Nature. ISMAR 2003.

Environmental Sounds Recognition System, Using the Speech Recognition System Techniques

Omar Aranda , Héctor M. Pérez , Mariko Nakano

Instituto Politécnico Nacional
SEPI-ESIME Culhuacan
Av. Santa Ana, 1000, Col. San Francisco Culhuacan, C. P. 04430, Coyoacan, México D. F.
arandauribe@calmecac.esimecu.ipn.mx

Abstract. This paper describes an environmental sounds recognition system using LPC-Cepstral coefficients as feature vectors and an artificial neural network backpropagation as recognition method. LPC-Cepstral data are totally dependents of the sound-source from which are computed. This system is evaluated using a database containing files from four different sound-sources under a variety of recording conditions. The training patterns used in the network-training ad testing processes, are extracted from the Discrete Fourier transform magnitude of the LPC-Cepstral matrices. The global percentages of verification and identification obtained in the network-testing process are 90.42% and 89.5%. Basically the idea here is to apply the techniques found in speech recognition systems to an environmental sounds recognition system.

Keywords- Artificial Neural Network, LPC-Cepstral Analysis, Discrete Fourier Transform.

Resumen. Este artículo describe un sistema de reconocimiento de sonidos ambientales utilizando como vectores característicos los coeficientes LPC-Cepstral y una red neuronal artificial backpropagation como método de reconocimiento. Los datos LPC-Cepstral son totalmente dependientes de la fuente de sonido de la cual son extraídos. Este sistema es evaluado con una base de datos que contiene archivos de cuatro fuentes de sonido diferentes grabados bajo diversas condiciones. Los patrones de entrenamiento son extraídos de la magnitud de la transformada de Discreta de Fourier. Los porcentajes de verificación e identificación obtenidos en la etapa de prueba de la red son 90.42% y 89.5% respectivamente. Básicamente la idea es aplicar las técnicas utilizadas en los sistemas de reconocimiento de hablante a un sistema de reconocimiento de sonidos ambientales.

Palabras clave- Red Neuronal Artificial, Análisis LPC-Cepstral, Transformada Discreta de Fourier.

1. Introduction.

Signals that humans can hear are one of the most important sources of informatic Humans obtain much information from not only voices but also non-verbal sounds. panoply of sounds in our daily lives, called "environmental sounds", are important to understand the surroundings. However, they have been little studied except as noi interfering with speech recognition systems. Much less effort has been directed towa systems capable of detecting, isolating, and identifying the panoply of sounds that fill every-day acoustic environment. In recent years, as the development of robots which behave in the real world, machine tools which have the intelligence to look themselves and their peripheral devices, and failure detection in electro domestic devis several studies on recognition of environmental sounds appeared [1-4]. These stud mainly focused on recognition of sound sources.

For recognition, environmental sounds have the following problems to be solved:

1. Environmental sounds are so various and changeful that they are hard to previously.
2. The environmental sounds are not regular in time.

Problem 1 means that we can use the parametric models as strategy for the environmen sounds recognition process, one kind of this parametric model is the artificial neu network backpropagation that uses a supervised learning algorithm. Problem 2 means as in speech recognition systems, sounds must be made regular or stationary on a interval, after this specific features can be extracted from these sounds and a neur network can be trained (recognition process). The idea here is to apply the methodolo found in speech recognition systems to verification and identification of environmen sounds using LPC-Cepstral analysis and an artificial neural network back-propagation recognizing method.

2. Proposed System.

Figure 1 shows the proposed system. This system consist of four sequential processe first a common database of environmental sounds is obtained, after this a segmentatic algorithm is applied to each token (file) of this database; third LPC-Cepstral features extracted from each segmented file and the DFT is computed from these coefficien finally the DFT magnitude is computed and a training strategy is adopted. The decision taken at the final process and a recognition percentage is computed.

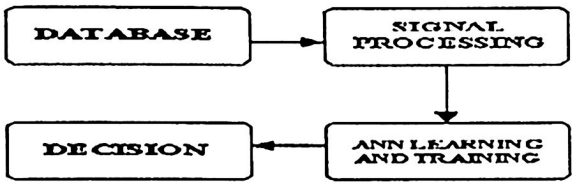


Fig. 1 Proposed environmental sounds recognition system.

2.1 Database Acquisition.

A database containing four different sound-sources was created, files were obtained from an online database. Files were used in the environmental sounds recognition system develop and evaluation. This Database contains 320 files (items). An endpoint algorithm was applied to each signal; this means that we separate portions of the signal stream containing the sound from the portions containing only background noise, which represents computational load to the system. Files are digitalized at 64,000 bits/second. Background sound levels were typically 25 to 30 dB below signal levels.

2.2 Signal Processing.

Figure 2 shows the applied processes in the signal analysis. With this signal analysis a high efficiency of feature extraction is obtained, this facilitates to the neural network the recognition process, this means that higher percentages of verification and identification can be obtained.

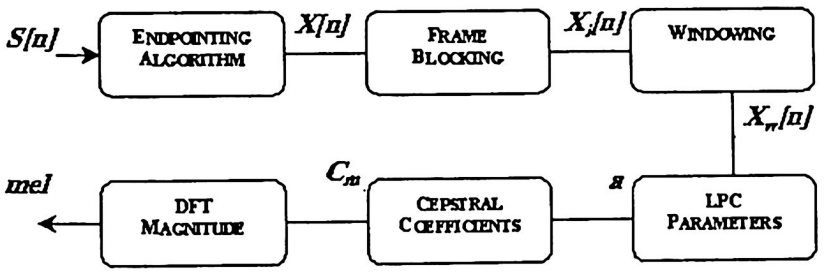


Fig. 2 Shows the methodology used in the signal analysis process

2.2.1 End-pointing Algorithm.

In time domain, magnitude, energy, power, maximums and minimums can be comput from which, the energy is used. Once the energy was calculated, a reference is obtain with this reference the signal can be limited.

In the discrete case the energy is defined as:

$$E[n] = \sum_{n=-\infty}^{\infty} s^2[n] \tag{1}$$

Now, a gamma constant is defined, this constant indicates the number of samples tak from the signal.

In our case the sampling frequency is 8000 Hz. The following step is to make relationship between the sound signal and the gamma constant:

$$E[n] = [(1-\gamma) * E_{n-1}] + [\gamma * y_n^2] \tag{3}$$

To each file stored in the database an end-pointing algorithm was applied. In order to the signal two thresholds of 20 and 10 the maximum energy must be defined, corresponds to the percentage taken form the signal. This algorithm compares thresholds with each sample of the energy until a sample is greater or equal to the thresholds, indicating the signal's beginning.

2.2.2 Frame Blocking and Windowing.

The sound signal, $\hat{S}[n]$, is blocked into frames of 240 samples that corresponds to msec, in which voice is considered stationary [6], with adjacent frames being separate by 120 samples. The use of frames implies three parameters: frame size, frame increme and frame overlapping:

$$S_f = I_f + O_f \tag{4}$$

To the sequence of analysis frames generated from each end-pointed file was applied windowing algorithm, this means that a 240-point Hamming window was used:

$$\hat{S}_w[n] = \hat{S}[n]\, W[n] \tag{5}$$

Where $0 < n < N-1$, N is the number of samples in the analysis frame (240 samples) and $W[n]$ is a Hamming window. The frame advancement rate was chosen to yield frames that overlapped at least 50%, and so that the total number of frames between the signal endpoints was at least 64, specifically one second of each signal was analyzed. We used a Hamming window, a typical window used for the autocorrelation method of LPC. This windowing has repercussion in the time responses of the algorithms used but the recognition percentages are improved [6].

2.2.3 LPC Parameters and LPC-Cepstral Coefficients.

In each window 17 LPC coefficients were calculated with Levinson-Durbin recursion. LPC-Cepstral coefficients can be derived directly from the set of LPC coefficients using the recursion:

$$C[n] = -a[n] - \frac{1}{n}\sum_{k=1}^{n=1} kC[k]a[n-k] \qquad (6)$$

Where $n > 0$, $C_0 = a_0 = 1, k > p$ and $a[n]$ represents the linear prediction coefficients. The number of frames generated for each signal was of 64. The result in effect was that each signal was represented by a 17 by 64 array of Cepstral coefficients, with the 64 rows representing time and the 17 columns representing frequency.

A 64-point DFT was then calculated for each column in the matrix and the first 32 points of this symmetrical transform retained. The resulting square matrix is a two dimensional Cepstral representation of the input signal. Each column corresponds to a particular spectral frequency, and ach row corresponds to a temporal frequency. The first column contains the DFT of the power envelope of the signal. The first row contains the DFT of the average signal spectrum. The first element of the first column contains the average signal power level. It's typical of two-dimensional Cepstral representations of acoustic signals, and certainly for our signals, that this corner element is the largest component and the size of the components in the first row and first column are larger than the size of interior matrix components. After this the DFT magnitude for each column in the matrix is computed. The LPC-Cepstral coefficients, which are the Fourier transform representation of the spectrum, have been shown to be more robust for speech recognition than the LPC coefficients; in this case we applied this method.

2.3 Feature Extraction and Neural Network Learning and Training.

Sets of two coefficients were selected from the 1024 elements of each DFT magnitu matrix to serve as feature vectors for use in sounds verification and identification. coefficients chosen were take form the first and second columns of the two dimensio Cepstral matrices.

The used model is an artificial neural network backpropagation. The traditional backpropagation algorithm [5] is used. For each sound pattern, 50 sound files were in the network training process. The sound samples are first normalized so that average magnitude becomes zero and the standard deviation is one. Clusters, or class were formed by grouping the feature vectors for each type of sound. For the netwo training, the ideal number of hidden-layer neurons was chosen from the experimen work. The hope, of course, is that all the samples of each sound will cluster together that space and that cluster for different sounds will be rejected.

3. Results.

Two neural networks, per sound-source, were trained, because of we used two featur vectors from the DFT magnitude matrix of each sound-source. Four stages (one per neur network) were necessary for the network training and each stage corresponds to each stored in the database. 32 input-layer neurons were necessary for the neural netwo training, 10, 15 and 20 hidden-layer neurons were used in this neural network and the results were obtained with 20 neurons; 1 output-layer neuron, per network, was necessa for verify the source-sounds and 4 output-layer neurons in the identification process. training process for verification and identification consist of a matrix with the traini patterns, each network had to be trained with all patterns from all sound-sources.

3.1 Neural Network Testing.

Once the network has been trained, is used in the verification and identification process in this case, the sounds produced by cars, boats, motorcycles and airplanes. verification and identification percentages for each tested neural network can visualized in table 1 and table 2. Sounds-sources that have similarity, as motorcycles, car and airplanes, present higher percentages of false verification than those that don't hav similarities.

Table 1. Percentage of Verification for each Artificial Neural Network (ANN).

Sound-Sources	Percentage of Verification Corresponding to Testing Patterns					
	Training Patterns	Partial Total %	ANN Boats	ANN Airplanes	ANN Cars	ANN Motorcycles
Boat's Sounds	100%	98.33%	96.66%	0%	0%	0%
Airplane's Sounds	100%	88.33%	3.33%	76.66%	10%	10%
Car's Sounds	100%	86.7%	3.33%	10%	73.4%	13.33%
Motorcycle's Sounds	100%	88.33%	0%	6.66%	10%	76.66.%
% False Verification			2.22%	5.55%	6.66%	7.77%
Global % of verification.		90.42%				

The percentage of verification that corresponds to the neural network of boats (96.66%), 4.44% corresponds to patterns verified as false, this means that the output-layer neuron is zero.

Table 2. Percentages of Identification for each sound source.

Sound Sources	Testing Patterns	% identification Training Patterns	% identification Testing Patterns	Total
Airplane's Sound	30	100%	78%	89%
Motorcycle's Sound	30	100%	76%	88%
Car's Sound	30	100%	79%	89.5%
Boat's Sound	30	100%	83%	91.5%
Global %		100%	79%	89.5%

Some LPC-Cepstral coefficients are illustrated in Fig. 3 to 6 and here is demonstrated the difference between sound-sources and the similarity between sounds that come from the same source. Those differences facilitate to the neural network the verification and identification processes.

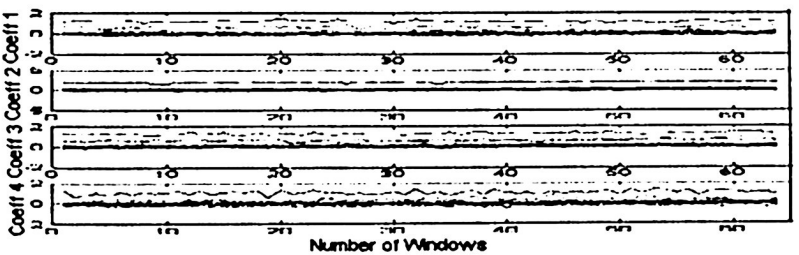


Fig. 3 LPC-Cepstral Coefficients from airplane engine.

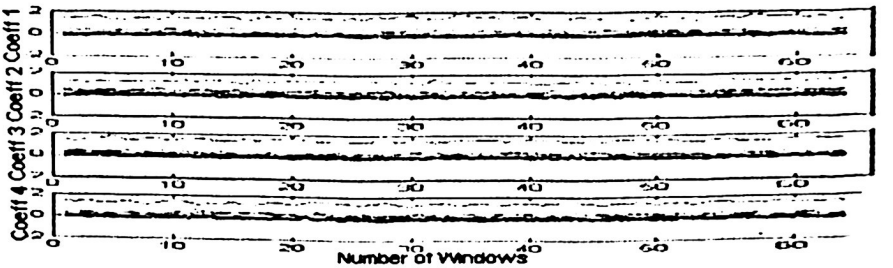


Fig. 4 LPC-Cepstral Coefficients from boats impeller.

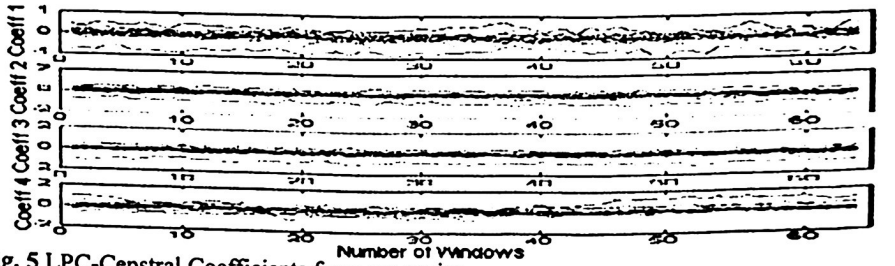


Fig. 5 LPC-Cepstral Coefficients from car engine.

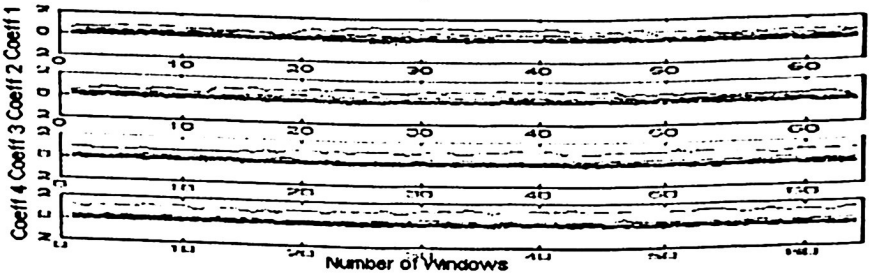


Fig. 6 LPC-Cepstral Coefficients from motorcycle engine.

4. Conclusions.

In this paper was proposed an environmental sounds recognition system based in the LPC Cepstral coefficients feature extraction, after this was computed the DFT magnitude this coefficients matrix. With this matrix an artificial neural network backpropagation trained. The verification and identification percentage were acceptable, 90.42%

89.5% although the number of feature vectors was small; specifically two feature vectors were used. The lowest percentages were obtained for seemed sound-sources, as cars, motorcycles and airplanes. This system seems to be good for some practices applications.

References.

[1] Goldhor, R. S., "Recognition of Environmental Sounds", Proceedings of ICASSP, Vol. 1, pp.149-152, 1993.

[2] Martin, K., "Sound-source recognition: Atheory and computational model", Ph.D. Tesis. *MIT Media Lab, 1999.*

[3] Yuya Hattori, Kazushi. Ishihara, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno, "Repeat Recognition for environmental sounds", in proc. of IEEE International Workshop on Robot and Human Interaction (ROMAN 2004), pp. 83-88, Kurashiki, Sep. 2004.

[4] Yasuhiro Ota, Bogdan M. Wilamoski, "Identifying Cutting Sound Characteristics in Machine Tool Industry with a Neural Network",

[5] Haykin Simon, "Neural Networks A Comprehensive Foundation", edited by Marcia Horton Bayani Mendoza de Leon, Prentice Hall,1999.

[6] T. Kitamura & E. Hayahara, "Word Recognition Using a Two-Dimensional Mel-Cepstrum in Noisy Environments", paper PPP6 presented at the 2nd Joint Meeting of the ASA and ASJ, Hawaii, 1998.

Fuzzy De-noise of ECG signals with wavelet techniques

I. S. Razo Zapata, L.E. Ramos Velasco and J. W. Vilanova

Center for Research on Systems and Information Technologies, Autonomous University of Hi
Carr. Pachuca – Tulancingo, km. 4.5 C.P. 42084, Pachuca, Hgo. MEXICO

email:{ri096373,lramos,julio}@uaeh.reduaeh.mx

Abstract. This work describes an adaptive fuzzy scheme for noise suppression of electrocardiograms (ECG). The Wavelet Transform (WT) is used for analysis and synthesis of signals in the noise cancellation process. A Takagi-Sugeno-Kang system carries out the diffuse adjustment of the threshold operator involved in the suppression of noise.

Key words: Fuzzy de-noise, threshold, TSK, electrocardiograms, wavelets.

1. Introduction

Electrocardiogram (ECG) is a graphical register of the electrical potentials produced cardiac tissue, which ones are located between a frequency range from 1 to 100 Hz. the synchronization mechanism is shown on Figure 1.

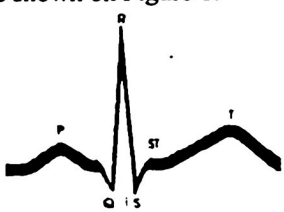


Fig. 1 Electrocardiogram elemental signal.

The bents presents in the drawing of an ECG represent different states of the heart ing a beat. The *P wave* and the *QRS complex* indicates contraction of the atriums ventricles respectively, the *ST segment* indicate the time that transcurse since a ventri contraction ends to a starting rest period previous to the ventricles start to contract selves for the next beat, and finally the *T wave* indicate the rest period of the ventri [5].

The micro-electrical signal analysis has been strongly used to determine different kinds of medical malaises and diseases such as muscular fatigue, cardiac dysfunctions, etc. Y thanks to the advance of this area, actually it is possible to realize medical diagnostics.

A motivation behind the implementation of transforms in signal processing is that for some applications signals in this domain offers better information in analysis that in the time domain [13].

Applications of the wavelet transform in signal processing in biomedical engineering include a great variety of tasks. In cardiology, analysis with wavelets is applied in electrocardiograms [1, 3, 7, 18].

Utilization of Fuzzy Logic provides machinery for carrying out approximate reasoning processes when available information is uncertain, incomplete, imprecise, or vague. The success of this methodology has been demonstrated in a variety of fields, such as control of systems, experts systems [17].

The aim in this work is to achieve a fuzzy de-noise in signals coming from electrocardiograms; implementing the WT and a threshold operator, with the purpose to obtain a signal free of noise.

The work is organized in the following way. Section 2 presents the scheme used to achieve the de-noise of the ECG signal, section 3 shows the methodology used in the elaboration of the work, section 4 shows the results obtained in simulation, and finally on section 5 are located the accomplished conclusions from results shown during work development.

Notation

This preliminary section fixes the mathematic notation used in the paper.

t	Continues time variable,	$t \in R$
n	Discrete time variable,	$n \in Z$
χ	Hilber pace,	
$\langle , \rangle$	Inner product, $\langle , \rangle : \chi, \chi \to C$	
l_2	$l_2 = \left\{ X[n] \mid \sum_{n=-\infty}^{\infty} \mid X[n] \mid^2 \right\} < \infty$	
L_2	$L_2 = \left\{ f[t] \mid \int_{-\infty}^{\infty} f(t) \mid^2 dt \right\} < \infty$	

2. Fuzzy - Wavelet scheme

In this section the fuzzy – wavelet scheme used for noise suppression is shown. Firstly summary of basic notions on wavelet transform, and later the fuzzy model used is presented.

2.1 Wavelet Transform (WT)

The continuous wavelet transform of a function $f(t)$ in a Hilbert space X , is defined follows:

$$CWT_f(a,b) = \langle \psi_{a,b}(t), f(t) \rangle, f(t) \in L_2(R) \quad (1$$

Where

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) \quad (2)$$

Is the basic function called mother wavelet, which one depends of the parameters a, where both are used to transfer and to climb respectively.

For the discrete case the wavelet transform for a discrete signal $x[n]$ is given by

$$DWT_f(a,b) = \langle \psi_{a,b}[n], x[n] \rangle = \sum_{-\infty}^{\infty} x[n] \psi_{a,b}[n], x[n] \in l_2(R) \quad (3$$

with sampled time, T and $n = kT$.

A general model for noise suppression using wavelets is depicted in Figure 2[14].

Fig. 2. Model for noise suppression.

In Figure 2, the first processing applied to the signal consists in an analysis based wavelets; with this analysis the signal is decomposed in Low and High frequencies, threshold operator acts in high frequencies and in this point noise suppression is formed. The threshold operator is fixed by a fuzzy system; finally a wavelet synthesis carried out for to yield a signal free of noise.

The analysis and synthesis process are shown in Figure 3; in the which one can obse that the signal is decomposed in high and low frequencies by one couple of filters wavelet coefficients, after it a downsampling and the threshold operator are applied. nally a subsampling and the synthesis process are performed with the dual filters $\overline{H}$ [14].

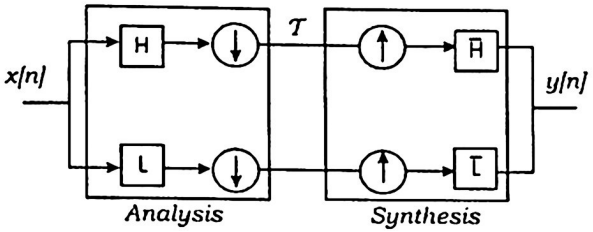


Fig. 3. Analysis and Synthesis in the wavelet scheme.

2.2 Fuzzy Model

The fuzzy model proposed by Takagi-Sugeno-Kang is described by fuzzy rules IF() and IF() .. THEN, each one is a lineal relationship between inputs and outputs. This set of rules is expressed under the form:

$$R_j = IF\ x\ is\ A_j \quad THEN \quad y = f_j(x)$$

Here A_j are linguistic terms, x is the input linguistic variable, while $y = (y_1, \ldots, y_j, \ldots y_{max})$ is the output variable. The value of the input linguistic variable may be crisp or fuzzy. If the value of the input variable is a crisp number then the variable x is called a singleton. The TSK system is used to achieve adjusts of the threshold operator dynamically, over each decomposition in the wavelet scheme.

3. Methodology

In this section two kinds of signals were used; reference signal and noisy signal. The reference signal is ECG signal without noise, the noisy signal is filtered with the fuzzy – wavelet scheme to approximate it to the reference signal.

The carried out process is illustrated in the following steps:

- Discrete Wavelet Transform (DWT) compute.
- Determination of the noise (entropy, energy).
- Filtering process with threshold operator.
- Inverse Discrete Wavelet Transform (IDWT) compute.
- Comparison between the reference letter and the IDWT, with the purpose of calculating the reconstruction quality.

The DWT was computed with the analysis wavelet presented in Figure 3, the ECG signal was decomposed in low frequencies (approaches) and high frequencies (details), after a subsampling process was applied, and as result, a coarse version of the original signal was obtained.By empirical approaches was determinate two decomposition levels for the analysis process and the wavelet of analysis was the wavelet of Daubechies 10.

In order to determinate the levels of noise presented in the ECG signal, two parameters were considered entropy and energy. Entropy and energy were calculated from details (high frequencies) in the signal. High entropy indicates noise in the signal and if this energy is not too large, noise has a relatively small influence on the important large signal coefficients. These observations suggest that small coefficients should be replaced by zero, because they are dominated by noise and carry only a small amount of information. Both parameters are used to adjust the threshold operator.

Noise suppression is accomplished with the threshold operator shown in Figure 2, this operator attenuates or keeps components in signal.

The threshold operator is fixed with the TSK system, which depends of energy and entropy in the signal; both parameters are represented by five membership functions shown in Figure 4. Besides twenty five (25) IF ... THEN rules are generated. The minimum is used as *t-norm* in the reasoning process, to determinate the best threshold.

The set of fuzzy rules is shown in Table 1, the rules has been built of smaller to bigger, below to up, and left to right. In this way, **IF** the *energy* is *smaller* and the *entropy* is *smaller* **THEN** the threshold operator should be pondered by 0.01.

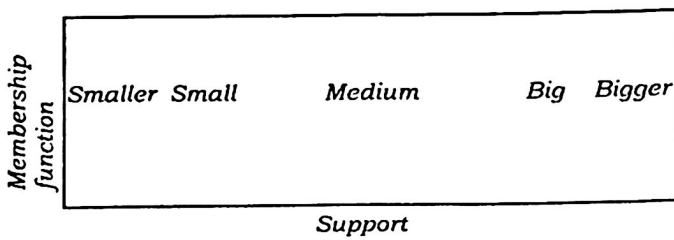


Fig. 4 Membership functions for energy and entropy.

In order to apply this method, it is necessary to consider that the used signals are long play, like for example the Holter signals. In this case, the methods applied can fail because throughout the signal the noise level will change. A possible solution is to divide the signal in a series of intervals, and to process each one of them separately.

Table 1. Set of fuzzy rules.

0.10	0.15	0.30	0.45	0.50
0.15	0.30	0.45	0.50	0.55
0.30	0.45	0.50	0.55	0.70
0.45	0.50	0.55	0.70	0.85
0.50	0.55	0.70	0.85	1.00

3.1 Threshold Operator

Thresholding is performed so as to zero out small magnitude wavelet coefficients and retain the value of large magnitude coefficients. The general soft threshold operator is defined as [14]:

$$(F_\delta c)_n = \begin{cases} c[n], & |c[n]| > \delta, \\ 0, & otherwise \end{cases} \tag{4}$$

Where $c[n]$ are the details (high frequencies). The threshold δ proposed is given by

$$\delta = \sqrt{2\log(N)\bar{\sigma}} \tag{5}$$

where N is the level of decomposition and $\bar{\sigma}$ estimation of the noise variance

$$\bar{\sigma} = median(|c[n]|)/0.6745 \tag{6}$$

The calculation of inverse discrete transformed wavelet (IDWT) for the reconstruction of the signal is made firstly by means of the process of synthesis shown in Figure 3, consistent in an upsampling, followed of a convolution with the filters wavelet L and H.

Finally a comparison between the reference signal and the filtered signal is made to determine the effectiveness of the filtrate. This comparison is made taking the measurement from the mean square error:

$$E(e) = \frac{1}{N}\sum_{n=0}^{N}(e)^2 \tag{7}$$

Where e is the difference between the reference signal and the filtered signal, i.e. $e[n] = d[n] - y[n]$

Table 2. De-noise schemes.

De-noise Scheme	Error
Soft threshold	0.02835
Fuzzy threshold	0.02640

4 Lab's results

For the implementation of these schemes were used a set of ECG signals that previously were obtained from the PhysioNet's network database[1]. Comparison between the soft threshold and the fuzzy threshold was made, and the results are shown on Table 2. The fuzzy threshold has relatively a better performance, with the reference signal.

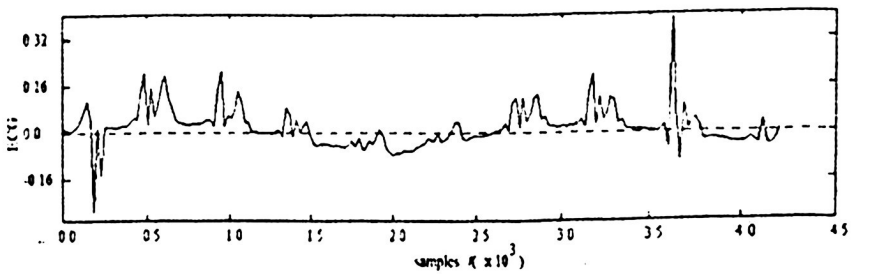


Fig. 5 Filtered signal with soft threshold.

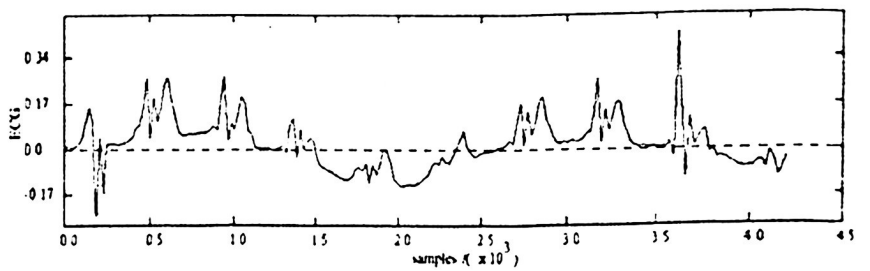


Fig. 6. Filtered signal with fuzzy threshold.

[1] www.physionet.org

Signal obtained by both schemes are shown in Figures 5 and 6. Figure 5 represent the filtered signal with the soft threshold, and Figure 7 represents the filtered signal with fuzzy threshold.

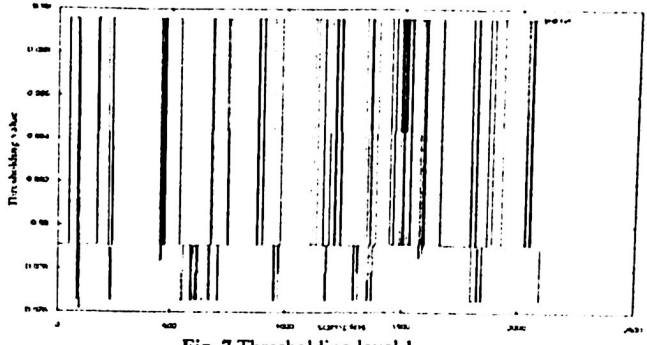


Fig. 7 Thresholding level 1.

The behavior presented by the threshold operator in the first level of decomposition appears in Figure 7, where it is possible to observe that its value is changing according to the values of entropy and energy during time windows, these windows of time are of 11 samples. Besides is possible to observe that Figure 8 presents a time scale equal to half of the original signal, because the threshold operator acts on the coefficients of high frequency in each level of decomposition where every time the amount of samples is reduced to half.

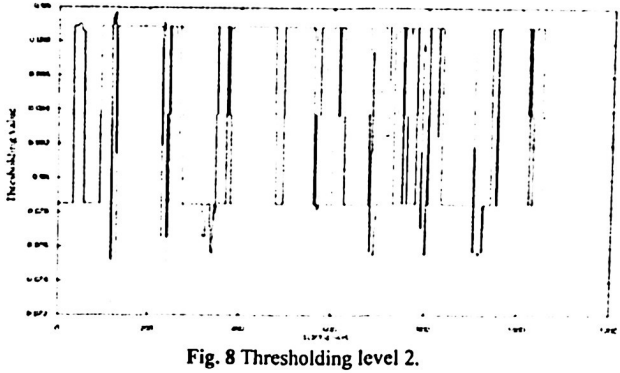


Fig. 8 Thresholding level 2.

Finally in Figure 9 the performance of the threshold operator for the second level of decomposition is shown, and in the same way its value is varying during the time.

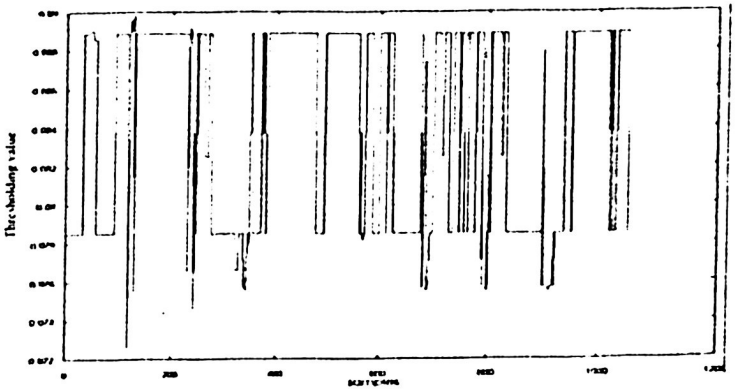


Fig. 9 Thresholding level 2.

5 Conclusions and Outlooks

In this work is proposed a scheme and algorithm of adapted filtering. With the properties of the wavelet transform, was possible to implement a combined digital filter with the threshold operator. In the other hand the implementation of a fuzzy system TSK was made, to adjust dynamically the threshold operator, which took in consideration physical properties from the signal, such as the entropy and the energy.

The performance of the scheme presented, show better results with respect to the classic scheme of noise suppression with wavelets (soft threshold)[3]. This performance can be improved, adjusting the set of fuzzy rules of TSK system, with more properties of ECG signals. In addition linguistic attributes of the signals can be used.

References

[1] Akay Metin, Mello Claudia, *Wavelets for Biomedical Signal Processing*, 19th International Conference, Chicago, 1997.

[2] Chassaing Rulph, *Digital Signal Processing, Laboratory Experiments Using and the TMS320C31 DSK*, John Wiley and Sons, Inc.

[3] Cuesta F. D., Nóvak D., Pérez C. J., Andréu G. G., Eck V., Sastre M. C., Llorca A. M., Reducción del ruido en señales electrocardiográficas mediante la transformada wavelet, República Checa.
[4] Daubechies I., Ten lectures on wavelets, SIAM, Philadelphia, PA, 1992.
[5] Dubin D., Electrocardiografía práctica, Mc Graw Hill, 1976.
[6] Haykin S., Adaptive filter theory, 2nd. Ed., Englewood Cliffs, NJ: Prentice Hall, 1991.
[7] Kanapathipillai M., Jouny I., Hamilton P., Adaptive wavelet representation and classification of ECG signals,'IEEE, 1994.
[8] Kehtarnavaz Nasse, Simsek Burc, *C6X Based Digital Signal Processing*, 1977.
[9] Kuo S. M., Lee B. H., Real-time digital signal processing, implementations, applications and experiments with the TMS320C55X, John Wiley and Sons Ltd., 2001.
[10] Lee J-W., Lee G-K., Design of an adaptive filter with a dynamic structure for ECG signal processing, International Journal of Control, Automation and Systems, 2005.
[11] Oppenheim Alan y Willsky Alan, *Señales y Sistemas*, Prentice Hall, 1983.
[12] Proakis John y Manolakis Dimitris, *Tratamiento Digital de Señales: Principios, algoritmos y aplicaciones*. Prentice Hall 1998.
[13] Rioul Olivier, Vetterli Martin, *Wavelets and Signal Proccessing, IEEE SP Magazine*, pp. 14-38, 1991.
[14] Theolis Anthony, Computational Signal Processing with Wavelets, Birkhäuser Boston, 1998.
[15] Thuillard Marc, *Fuzzy Logic In The Wavelet Framework*, Proc. Toolmet'2000 —Tool Environments and Development Methods for Intelligent Systems, April 13-14 2000.
[16] Thuillard Marc, *New Perspectives for the Integration of Wavelet Theory into Soft Computing*, Siemens Building Technologies.
[17] 17 TMS320C6x User's Guide, Digital Signal Processing Products, Texas Instruments, October 1994.
[18] Vetterli Martin, Wavelets and Subband Coding. Prentice Hall PTR, 1995.
[19] Walker E. A., Nguyen H.T., A First Course in Fuzzy Logic, Chapman \& Hall/CRC, 2000.
[20] Wiklund U., Akay M., Niklasson U., Short-term analysis of heart-rate variability by adapted wavelet transforms, IEEE in medicine and biology, 1997.
[21] Wiklund U., Akay M., Spectral analysis of bioelectric signals by adapted wavelet transforms, 2nd International Conference on Bioelectromagnetism, 1998.

Best basis selection for image compression using cellular automata transforms.

Juan E. Paz

Center for Studies on Electronics and Information Technologies, Central University of Las Villas, Zip code: 54830, Santa Clara, Cuba.
jpaz@ceeti.uclv.edu.cu

Abstract. Cellular Automata Transform is an alternative tool for image and signal processing like *Fourier* or *Wavelet* transforms. From the big number of transformation bases that can be generated using this method is possible to find the one more accurate for the problem in question that when using previous transforms. The present paper deals with an efficient basis selection criterion measured though the entropy of the transformed coefficients in an application like image compression.

1 Introduction

Cellular Automata are dynamical systems in which space and time are discrete. They were presented for the first time by *von Neumann* and *Ulam* [1] and had been progressively used in modeling a great variety of dynamic systems in diverse applications [2]. Like former transforms as *Fourier* or *Wavelet*, the Cellular Automata Transform (CAT) finds a practical application in digital image processing like compression, noise filtering or edge detection.

The theory behind this transform and its applications has been exposed by *von Neumann* and *Lafe* [1],[2]. According to them, there are a set of parameters involved in obtaining a single CAT basis. These are: the space dimension (one dimensional $D = 1$ or two dimensional $D = 2$), the size of the basis (N), the starting initial state, also called cellular automaton that can be presented as a vector in $D = 1$ or as a $N \times M$ matrix in $D = 2$.

Other parameters are the evolution time ($T > 0$), the number of states each cell can take ($K > 2$), the rule number according to which the initial state will evolve (RN), the number of neighbors considered in the evolution (m), the *Class* (=1 in case $T = N$ and Class = 2 in case $T > N$), and the *Type* of linear development for the obtained evolution in order to calculate the basis. See equation 4 for an example.

All parameters and their values for the experiments in this paper are listed in table I and II. In general, their amount and variability range allows obtaining a huge number of bases and, among these, the one which fits better to the problem in question than when using other transforms.

2 Experiments.

2.1 Obtaining the basis

If f_i is a discrete signal with $i = 1, \ldots N$ then a basis $A_{i,k}$ is search so that

$$f_i = \sum_k c_k A_{i,k} \qquad (1)$$

where c_k are the coefficients obtained from the inverse transformation;

$$c_k = \sum_i f_i B_{i,k} \qquad (2)$$

and $B_{i,k}$ is the inverse of $A_{i,k}$.

The construction of a basis starts with the initial state (*IS*) of the automata. In case ($D = 1$) it can be described by an N - element vector $a(i, t)$ with $t = 0$ and $i = 1,\ldots, N$, and in $D = 2$, by a N x M matrix which elements are $a(i, j, t)$ with $t = 0$, $i = 1,\ldots, N$ and $j = 1,\ldots, M$. They are both composed of cells linked to each other and each one taking a finite number of K possible states (values). Figure 1 shows an initial state with $D = 1$, $N = 4$ and $IS = [1\ 0\ 1\ 1]$.

Fig. 1 Cellular Automaton.

The future state of each cell at $t > 0$ will be determined by an evolution rule and by the present states of the automaton in a certain neighbourhood m. Where a_{it} is the i cell at state t. $a_{it} = a(i, t)$, i = space index, t = time index. The initial state will evolve T times to build a N x T matrix in 1-D case according to an specific evolution rule described by a rule called *Wolfram* rule. See equation 3. In *Class* =1 evolutions, the evolution time $T = N$ which returns square matrices. With these matrices and using equation 5, the transformation basis can be obtained.

The *Wolfram* vector W_j ($j = 0, 1, 2\ldots, 2^m$), is a weight vector obtained form the rule number (*RN*), the maximum state number of each cell and the number of neighbours to consider in evolution (m). Its precise expression can be found in [1]. Example: $W = [0\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 1]$, is the *Wolfram* vector for $RN = 53$, $K = 2$ and $m = 3$. The evolution of cell a_{it} at time $t+1$, for 3 neighbours, is described by the following rule:

$$a_{it+1} = (W_{a_{i-1}} + W_1 a_{i1} + W_2 a_{i2} + W_3 a_{i-1} a_{i1} + W_4 a_{i1} a_{i2} + W_5 a_{i-1} a_{i2} + W_6 a_{i-1} a_{i1} a_{i2} + W_7)^{a_{it}} \bmod K \qquad (3)$$

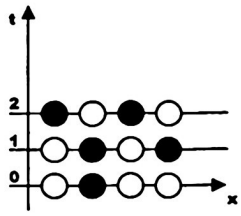


Fig. 2 Evolution of initial state.

The evolution leads to a N x T matrix E used to form the desired basis according to some lineal combination:

$$A(i,j) = \alpha + \beta \cdot E(i,j) \qquad (4)$$

where α , $\beta \in \Re$ are real number introduced by the user.

In 2-D case, the basis can be obtained as a result of linear evolutions of two dimensional initial states or as an inner product of one dimensional basis already obtained:

$$B(i,j,k,l) = A(i,j)*A(k,l) \qquad (5)$$

All bases used in this work were obtained using the second method. *Gramm-Schmidt* procedure was performed over those bases that result not orthogonal after an orthogonality test [5].

2.2 Ordering the basis

The orthonormal bases were placed ordered in a set according to its performance measured in the entropy values of the coefficients obtained with each of them. According to the relation between *Shannon* entropy [6] and compression ratio of a data set, only when entropy based algorithms are used, bases among the first positions will report the best results in compression purposes. In this particular case bases of dimension 4x4x4x4 ($N = 4$) and dimension 8x8x8x8 ($N = 8$) were studied.

Equation 7 shows an example of a basis of size 2x2x2x2 ($N = 2$),

$$A(i,j,k,l) = \begin{bmatrix} a_{ij11} & a_{ij12} \\ a_{ij21} & a_{ij22} \end{bmatrix}, \quad a_{ij11} = \begin{bmatrix} a_{1111} & a_{1211} \\ a_{2111} & a_{2211} \end{bmatrix}, \quad i,j,k,l = 1,2. \qquad (7)$$

When dealing with cellular automata bases is common to obtain the same basis using different set of parameters which justifies their widespread use in other applications like encryption for example. For compression purposes on the other hand, only different bases are required and among them, the one with better performance. So, in order to have only those different to each other, only the new and different to those already in, are added to the set.

2. 3 Bases classification.

The process is done transforming a set of real images with every basis in the set and obtaining the entropy values of their coefficients transformed. The bases are ordered in the set according to the transformed images coefficients entropy values. So is possible to select among the first ones, the best for compression purposes. In this case a lossless arithmetic compression algorithm has been used.

Table 1. Parameters for obtaining basis $N = 4$ (left) y $N = 8$ (right)

Basis	RN	IS				Basis	RN	IS							
1	28	1	0	1	1		11	0	1	1	1	1	0	1	0
2	10	0	0	0	1	2	15	0	1	1	0	1	1	1	1
3	15	0	0	0	1	3	15	0	1	1	1	1	0	1	1
4	21	0	0	0	1	4	11	0	0	0	0	1	1	0	0
5	8	0	0	0	1	5	15	1	0	0	0	0	1	0	0
6	130	0	0	1	0	6	15	1	0	0	1	0	0	0	0
7	16	0	0	1	0	7	1	1	1	0	0	0	0	0	0
8	16	0	0	1	0	8	43	1	1	1	0	0	1	1	1
9	252	0	0	1	0	9	43	1	1	1	1	0	0	1	1
10	59	0	0	1	0	10	15	1	1	1	1	0	1	1	0
11	173	0	0	1	1	11	11	0	0	0	1	1	0	0	0
12	230	0	1	1	0	12	15	0	0	0	1	1	1	0	1
13	74	0	1	1	0	13	14	0	0	1	0	1	0	1	1
14	188	1	0	0	1	14	15	0	1	0	1	1	1	0	0

Table 2. Other parameters

N	D	IS	RN	Class	T	Type	K	m	α	β
variable	2	variable	variable	1	N	2	2	3	-1.0	2

But first one has to be sure that there is a different performance between them no matter what images they are processing. Ten real images (256 x 256, 8bpp) were transformed with two different sets of bases with $N = 4$ and $N = 8$, and the entropy values of the resulting coefficients were calculated.

Table 3. Bases with $N = 4$ (Left) and $N = 8$ (Right). Entropy and *PSNR* after reducing the amount of necessary bits from 10 to 6 ($N = 4$) and from 14 to 8 ($N = 8$).

Basis (EN)	Entropy [bpp]	PSNR [dB]
28	1.81	46.-
10	3.89	56.1
15	4.08	52.3
21*	3.98	56.6
78	1.93	46.5
130	3.94	56.1
16	3.94	56.1
167*	2.96	45.4
252	4.08	51.9
59	1.88	46.4
173	1.81	46.-
230	1.69	43.-
71	1.83	46.9
188	1.83	46.9

Basis (IS)	Entropy [bpp]	PSNR [dB]
122	4.86	51.1
111	5.35	51.8
123	5.23	51.4
12	1.04	27.3
132	2.26	35.9
144	1.45	30.8
192	1.08	27.6
231	4.86	51.1
243	5.12	51.5
246	5.35	51.8
24	1.39	29.-
29	1.40	30.3
43	5.11	51.5
92	2.26	35.9

The bases in the paper had been obtained using the parameters in tables I and II. The initial states are one dimensional but using the equation 5 two dimensional bases can be obtained from one-dimensional ones. *Type* = 2 means a linear expression like the one in equation 4, $K = 2$ is the maximum number of states of each cell; also [0, 1].

In figure 3 the lines does not cross each other. This means that there is an independent behaviour or different performance of each basis for any image. The expression used for entropy is *Shannon* entropy:

$$E = -\sum_{i}^{N} p_i \log_2(p_i), \quad i = 1,\ldots, N \quad [bpp]. \qquad (7)$$

where p_i is the probability for one specific value in the coefficients and N is the number of coefficients.

The basis with $N = 8$ showed similar behaviour that those with $N = 4$. It has been shown that their performance is different from each other and independent for each image.

3. Results and discussion.

In order to represent coefficients obtained with bases of $N = 4$, 10 *bpp* (bits per pixel) were needed, (2 *bpp* more than in the original image), that is 131072 bits more then the original image, while in transforming the images with bases of $N = 8$, 14 *bpp* were needed.

Table 4. Entropy and compression ratios for bases with $N = 4$ (left) and $N = 8$ (right)

#	RN	Entropy [bpp]	TC [bpp]	PSNR [dB]
8	167*	0.81	0.79	23.2
12	230	1.03	0.92	23.9
11	173	1.06	0.80	25.5
1	28	1.07	0.80	25.5
10	59	1.1	0.88	25.4
14	188	1.17	0.93	26.4
13	74	1.17	0.94	26.4
5	78	1.25	1.03	25.4
2	10	2.81	2.0	34.8
7	16	2.81	2.02	35.0
6	130	2.81	2.06	35.0
4	217	2.86	1.99	35.7
3	15	2.92	1.78	29.8
9	252	2.95	1.87	29.6

#	IS	Entropy [bpp]	TC [bpp]	PSNR [dB]
4	12*	0.43	0.40	21.5
7	192*	0.50	0.47	21.1
11	24*	0.73	0.59	22.8
12	29*	0.74	0.70	22.9
6	144*	0.79	0.75	22.8
14	92*	1.32	1.22	27.6
5	132*	1.32	1.22	27.6
8	231	1.94	1.27	31.0
1	122	1.94	1.29	31.0
9	243	2.09	1.41	30.7
13	43	2.09	1.42	30.7
3	123	2.17	1.47	29.7
2	111	2.25	1.58	30.1
10	246	2.25	1.58	30.1

The standard deviation between original and reconstructed images after rounding the coefficients was only 0.28 units in all cases, i.e. to a *PSNR* of 60 units. This means that the simple rounding of the coefficients does not introduce errors in the reconstructed image.

The amount of necessary *bpp* was reduced to 6 in bases with $N = 4$ (except in basis with $RN = 167$ where 10 *bpp* were needed). On the other hand in bases with $N = 8$, this quantity was reduced to 10 and 8 *bpp* in some cases from 14 *bpp* originally necessary to represent the transformed coefficients. These reductions brought considerable entropy decrease of the coefficients. A non-uniform coder was also introduced who reduced more the coefficients entropy values while maintaining the *PSNR* in acceptable values i.e. with very small loss in the reconstructed image. See table III.

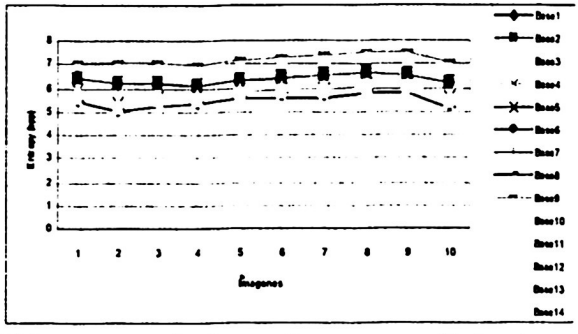


Fig. 3 Cellular automata transform bases performance measured in the entropy values of the coefficients of 10 different images. Each basis shows a stable behaviour in comparison to the others, therefore is possible to achieve better results with a certain basis no matter what the input image is. Bases with N = 8 show similar behaviour.

The ordered basis according to the mean entropy value, the *PSNR* values and the mean compression ratio over the set images are shown in table IV. The minimal loss in the reconstructed images is due to the presence of quantization.

The entropy values were calculated as the mean value for each basis of all image transformed coefficients using expression (7). Compression ratio has been calculated as the quotient of necessary bits to represent the original image and the amount of bits necessary to represent the codified image.

The *Peak Signal-to-Noise Ratio (PSNR)* was calculated using expression

$$PSNR = 10 \cdot \log_{10}\left(\frac{x_{pp}^2}{MSE}\right) \quad [dB] \,, \qquad (8)$$

where $x_{pp} = 255$,

$$MSE = \frac{1}{N} \cdot \sum_{i=1}^{N} (x_i - x'_i)^2 \qquad (10)$$

is the Mean Square Error and x' is the reconstructed image.

As explained before, the number of basis that can be obtained varying the parameters is a huge number. In this paper we had limit this number to a few bases in order to prove the relation of entropy and compression ratio when using entropy based coders.

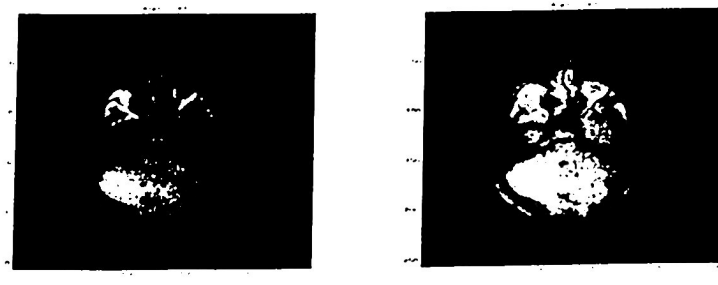


Fig. 4 a) Image a) Magnetic resonance image (T1slice), 256×256 pixels and 8 bpp, b) reconstructed image after being compacted: *PSNR*= 25.4 dB, TC = 0.39 bpp (CR = 20.2 times).

The entropy based coders use the probability of occurrence of a value in the transformed coefficients, while the energy based coders take account of the coefficients magnitude. Therefore when using energy based coders or combining these with entropy based coders, the criterion exposed here will not be effective in selecting the best basis for compression purposes.

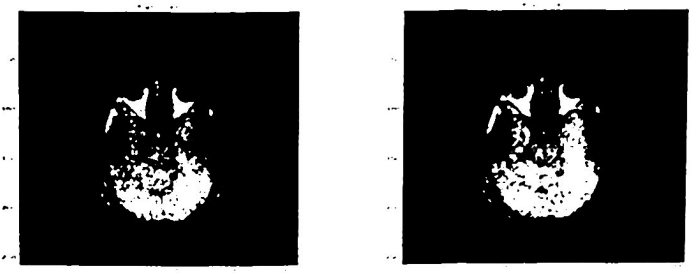


Fig. 5 a) Magnetic resonance image (T1slice). 256×256 pixels and 8 bpp.
b) reconstructed image after being compacted *PSNR*= 27.1 dB,
TC = 0.77 bpp (CR = 10.3

Figures 4 and 5 show two images compressed using base named (*IS* = 12) in table IV. The images are 256x256 pixels and 8 *bpp*. The compression rates obtained are between 10 and 20 times, keeping the *PSNR* bigger then 25 *dB*, what is acceptable for processing medical images.

References

[1] J. von Neumann, Theory of Self-Reproducing Automata (edited and completed by Arthur Burks), University of Illinois Press, 1966.

[2] O. Lafe, Cellular Automata Transforms: Theory and Applications in Multimedia Compression, Encryption, and Modeling, Kluwer Academic Publishers, 2000, ISBN 0-7923-7857-1.

[3] R.R. Coifman, M.V. Wickerhauser, "Entropy-Based Algorithms for Best Basis Selection". *IEEE Trans. on Inf. Theory*, Vol.38, No.2, 1992.

[4] I.H. Witten,, R.M. Neal, J.G. Cleary, "Arithmetic coding for data compression", *Communications of the ACM*, Vol.30, No.6, June 1987, pp. 520 - 540.

[5] G. Arfken, "Gram-Schmidt Orthogonalization", in *Mathematical Methods for Physicists*, 3rd ed,. Orlando, FL: Academic Press, pp. 516-520, 1985.

[6] C.E. Shannon, "A mathematical theory of communication",. *Bell System Technical Journal*, Vol. 27, 1948.

Wavelet Domain Order Statistics Filters for Image Denoising

Jesús Martínez-Valdes, Francisco Javier Gallegos Funes, Volodymyr Ponomaryov

Section of Senior Studies and Research of Superior School of Mechanical and Electrical Engineering Higher School of the National Polytechnic Institute, Mexico D.F., Mexico.
jesusipnmex@yahoo.com.mx, fgallegosf@ipn.mx

Abstract. This paper shows the performance results to apply the analysis wavelet with the order statistics in the task of the suppression of impulsive noise in color images. The proposed filtering scheme in defined as two filters in the wavelet domain to conform the structure of a general filter that can be modified in some headings, the first filter is based on redundancy of approaches, the second one is the wavelet domain iterative center weighted median algorithm. With the structure of the proposed filter different implementations for the estimation of the noisy sample are already carried out using different order statistics algorithms that by their good performance can be beneficial in image processing applications.

1 Introduction

Many different classes of filters have been proposed for removing noise from images [1, 2]. They are classified into several categories depending on specific applications. The order statistics filters are designed to suppress noise of different nature, they can remove impulsive noise and guarantee detail preservation [1, 2]. By other hand, the filters based in the wavelet domain provide a better performance in terms of noise suppression in comparison with different filters in the spice domain [3, 4].

In this paper we proposed a filter that works in the wavelet domain and to use different order statistics algorithms in its filtering scheme.

The proposed filter is conformed by two filters that carry out the impulsive noise suppression in the wavelet domain. The first filter based on redundancy of approaches [5] smoothes the LF of the noisy image by means of double convolution operation (first of decomposition and after reconstruction) between the wavelet coefficients and the samples of the corrupted image, the second one is based wavelet domain iterative center weighted median filter [6] that provides an analysis of the histograms of the wavelet coefficients of several pairs of images (original and corrupted) through different scales and carries out an improved estimation of the variance field of the noisy wavelet coefficients of the image and with aid of the estimator of the minimum mean square error finally obtains the filtered wavelet coefficients of the approaches and details of the image.

The structure of the proposed filter is designed so that it can be modified in the sections of the first and second detection of the noisy sample in addition to the estimation algorithm, all contained in the wavelet domain iterative center weighted median filter [6] to increase its robustness and to improve its performance in the task of the impulsive noise suppression. We introduce in the proposed filter the algorithms of the SD-ROM, Tri-state Median, Adaptive Center Weighed Median, Multi-stage Median, FIR Median Hybrid, and MM-KNN Filters [7-11], and adapt them to work in the wavelet domain into the two detection blocks proposed in this paper.

2 Wavelet Domain Order Statistics Filtering Scheme

The structure of the proposed filter is constituted by two filters, the filter based on redundancy of approaches [5] and the wavelet domain iterative center weighted median (ICWMF) filter [6] as it is shown in the Fig. 1. For each color component of the noisy image is necessary to apply all the steps contained by this structure. This technique apply up to 5 scaling levels for the details and only 1 scaling level for the approaches, it obeys to the differences that can be found between them and their importance at the moment of the image reconstruction. Other operations are indicated to make clearer the wavelet analysis that it is carried out in this paper. We modify this structure in the block of the ICWMF. For that reason, the expressions used by the ICWMF to calculate the improved estimation of the variance field of the noisy wavelet coefficients will be required to indicate when and where different proposed filtering algorithms will take place into it to improve the performance of the proposed filter.

The first stage of the ICWMF [6] that detects if a sample contains noise or not is:

$$\hat{\sigma}^2_{g^s}(k)=\begin{cases}\tilde{\sigma}^2_{g^s}(k) & \text{if } \lambda_s \geq \lambda_{sh} \\ med^*_{c_w}(\tilde{\sigma}^2_{g^s}(j)) & \text{if } \lambda_s < \lambda_{sh}\end{cases} \quad (1)$$

where $\tilde{\sigma}^2_{g^s}$ is the variance field estimated previously, k is central sample in the filter window, j is one of the N sample contained into the window, λ_s is the standard deviation of the preliminary estimate of the signal coefficients variances $\tilde{\sigma}^2_s(k)$ in each scale and λ_{sh} is the discriminating threshold defined as [6]:

$$\lambda_{sh} = \frac{\sum_s \lambda_s 2^{-s}}{\sum_s 2^{-s}} \quad (2)$$

where s is the scale used in the wavelet analysis and 2^{-s} is the weighting function.

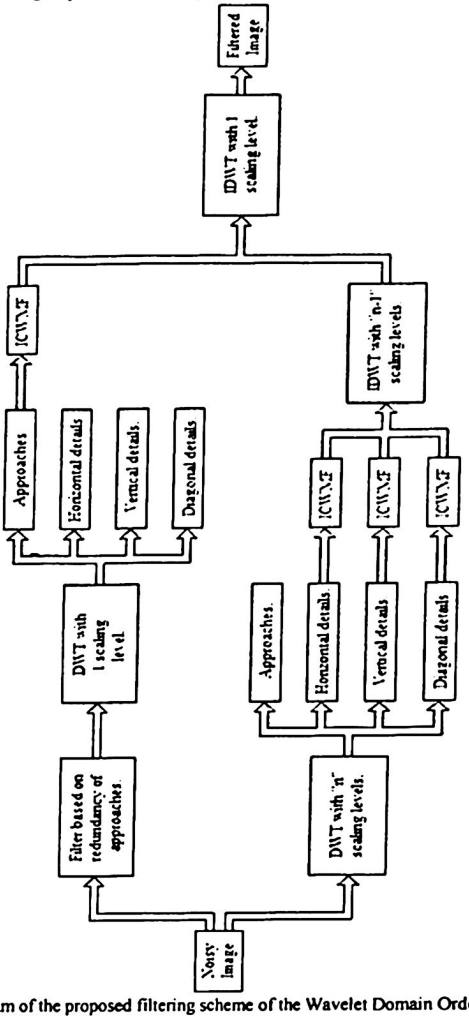


Fig. 1. Block diagram of the proposed filtering scheme of the Wavelet Domain Order Statistics Filter.

2.1 Wavelet Domain Signal Dependent Rank-Ordered Mean Filter

Consider a 3x3 filter window size with a central sample $\tilde{\sigma}^2_{g'}(m)=\tilde{\sigma}^2_{g'}(k)$ where $m=[m_1,m_2]$ is the location of the sample in the image, and a vector $\varphi(m)$ which contains the neighbor pixels of $\tilde{\sigma}^2_{g'}(m)$ into the window [1],

$$\begin{aligned}\varphi(m) &= [\varphi_1(m), \varphi_2(m), \varphi_3(m), \varphi_4(m), \varphi_5(m), \varphi_6(m), \varphi_7(m), \varphi_8(m)] \\ \varphi(m) &= [\tilde{\sigma}^2_{g_1}(m_1-1, m_2-1), \tilde{\sigma}^2_{g_1}(m_1-1, m_2), \tilde{\sigma}^2_{g_1}(m_1-1, m_2+1), \\ &\quad \tilde{\sigma}^2_{g_1}(m_1, m_2-1), \qquad\qquad \tilde{\sigma}^2_{g_1}(m_1, m_2+1), \\ &\quad \tilde{\sigma}^2_{g_1}(m_1+1, m_2-1), \tilde{\sigma}^2_{g_1}(m_1+1, m_2), \tilde{\sigma}^2_{g_1}(m_1+1, m_2+1)]\end{aligned} \tag{3}$$

The samples of the previous vector can be ordered to obtain the next vector [1]:

$$v(m) = [v_1(m), v_2(m), ..., v_8(m)] \tag{4}$$

where $v_1(m), v_2(m), ..., v_8(m)$ are the rank ordered elements of $\varphi(m)$ that satisfy the condition $v_1(m) \le v_2(m) \le ... \le v_8(m)$.

Then, the Rank Ordered Mean is defined as follows [1],

$$rom(m) = \frac{v_4(m) + v_5(m)}{2} \tag{5}$$

and the vector that contains the rank ordered differences $rod(m)$ is [1]:

$$rod(m) = [rod_1(m), rod_2(m), rod_3(m), rod_4(m)] \tag{6}$$

where $rod_i(m) = \begin{cases} v_i(m) - \tilde{\sigma}^2_{g_1}(m) & \text{for } \tilde{\sigma}^2_{g_1}(n) \le rom(m) \\ \tilde{\sigma}^2_{g_1}(m) - v_{9-i}(m) & \text{for } \tilde{\sigma}^2_{g_1}(n) > rom(m) \end{cases}$, and i=1,...,4.

Finally, the SD-ROM algorithm detects to $\tilde{\sigma}^2_{g_1}(m)$ as a noisy sample if any of the following conditions is true [1]:

$$rod_i(m) > U_i \tag{7}$$

where i=1,...,4, U_1, U_2, U_3 y U_4 are threshold, $U_1 < U_2 < U_3 < U_4$, and $U_1 = 8$, $U_2 = 20$, $U_3 = 40$, $U_4 = 50$ [1].

2.2 Wavelet Domain Tri-State Median Filter

The Tri-State Median (TSM) Filter joints the Median Filter to the Center Weighted Median (CWM) Filter into a noise detector that decides if a sample is corrupted or not. The TSM Filter algorithm is defined as [7]:

$$\theta_{TSM} = \begin{cases} \tilde{\sigma}^2_{g_1} & U \ge d_1 \\ \theta_{CWM} & d_2 \le U < d_1 \\ \theta_{Med} & U < d_2 \end{cases} \tag{8}$$

where θ_{CWM} y θ_{Med} are the values of CWM and median filters respectively, $\tilde{\sigma}^2_{g_5}$ is the value of the filtering center window, $d_1 = \left|\tilde{\sigma}^2_{g_5} - \theta_{Med}\right|$, $d_2 = \left|\tilde{\sigma}^2_{g_5} - \theta_{CWM}\right|$ with $d_2 \le d_1$, U is a selected threshold depending on the image and the noise level, a rank between 10 and 30 is defined [4] based on different simulations, in this case takes an intermediate value of $U = 20$.

2.3 Wavelet Domain Adaptive Center Weighed Median Filter.

The Adaptive Center Weighed Median (ACWM) Filter uses an adaptive operator which forms estimations based on the differences between the central sample of the filter window and the values from the CWM Filter with variations of its central weight. The ACWM Filter is defined as [8]:

$$\theta_{ACWM} = \begin{cases} \theta^1_{CWM} & d_k > U_k \\ \tilde{\sigma}^2_{g_5} & \text{otherwise} \end{cases} \tag{9}$$

where $d_k = \left|\theta^m_{CWM} - \tilde{\sigma}^2_{g_5}\right|$ are the differences between the Variable CWM Filter and the central sample in the analysis window, $\theta^m_{CWM} = med\left\{\left[\tilde{\sigma}^2_{g_5}\right]_1, m \Diamond \tilde{\sigma}^2_{g_5}\right\}$ is the Variable CWM Filter being m its weight and $\left[\tilde{\sigma}^2_{g_5}\right]_1$ are the samples contained in the filter window, $U_k = s \cdot MAD + \vartheta_k$ are the filter thresholds and $MAD = med\left\{\left[\tilde{\sigma}^2_{g_5}\right]_1 - \theta^1_{CWM}\right\}$ is the median absolute deviations from median, the parameter s gives necessary robustness to the filter and varies between $0 \le s \le 0.6$. The optimal values for these parameters are $s = 0.3$ and $\vartheta_k = [\vartheta_0, \vartheta_1, \vartheta_2, \vartheta_3] = [40, 25, 10, 5]$.

2.4 Wavelet Domain Median M-type K-Nearest Neighbor Filter

The algorithm of Median M-type KNN (MM-KNN) filter can be written as [9, 10]:

$$\hat{e}^{(q)}_{MMKNN}(i,j) = med\left\{g^{(q)}(i+m, j+n)\right\} \tag{10}$$

where $g^{(q)}(i+m, j+n)$ it is a set of K_c samples with weight according to the used function $\psi(\tilde{\sigma}^2_{g_5})$ required to carry out a comparison with the estimation of the previous step $\hat{e}^{(q-1)}_{MMKNN}(i,j)$. The initial estimator is $\hat{e}^{(0)}_{MMKNN}(i,j) = \tilde{\sigma}^2_{g_5}(i,j)$, $\tilde{\sigma}^2_{g_5}(i,j)$ is the central value of the filtering window, $\hat{e}^{(q)}_{MMKNN}(i,j)$ is the estimation in the iteration q.

The parameter K_c reflects the local data activity and the impulse presence [10],

$$K_c(i,j) = \left[K_{min} + aS(\tilde{\sigma}^2_{g_5}(i,j))\right] \le K_{max} \tag{11}$$

where a controls the detail detection, K_{min} is the minimum number of nearest neighbors to remove the noise, and K_{max} is the maximum number of neighbors used for the detection of edges and fine details. The optimal values for these parameters are $a=2$ and $K_{min}=5$, and $S(\tilde{\sigma}^2_{g_1}(i,j))$ is the impulsive detector defined as [9, 10]:

$$S(\tilde{\sigma}^2_{g_1}(i,j)) = \frac{med\left\{\left|\tilde{\sigma}^2_{g_1}(i,j) - \tilde{\sigma}^2_{g_1}(i+m, j+n)\right|\right\}}{MAD\left\{\tilde{\sigma}^2_{g_1}(i,j)\right\}} + 0.5\frac{MAD\left\{\tilde{\sigma}^2_{g_1}(i,j)\right\}}{med\left\{\tilde{\sigma}^2_{g_1}(i+k, j+l)\right\}} \qquad (12)$$

where $MAD_n = med\left\{\left|(\tilde{\sigma}^2_{g_1})_i - M_n\right|\right\}$ is the median of absolute deviations from median, $M_n = med\left\{(\tilde{\sigma}^2_{g_1})_i\right\}$, $(\tilde{\sigma}^2_{g_1})_i$ is the sample of the window that goes from left to right and of above to down, and M_n is the median of samples in the filtering window.

We also use the simple influence function in the MM-KNN filter [9],

$$\psi_{cut(r)}(\tilde{\sigma}^2_{g_1}) = \begin{cases} \tilde{\sigma}^2_{g_1}, & \left|\tilde{\sigma}^2_{g_1}\right| \le r \\ 0, & \left|\tilde{\sigma}^2_{g_1}\right| > r \end{cases} \qquad (13)$$

where r is a parameter between 0 to 256.

2.5 Wavelet Domain Multi-Stage Median Filter

The proposed algorithms mentioned before were applied to the proposed filter as a first detection block, but the following two algorithms were applied as a second detection block due that these algorithms only constitute the part of estimation of the noisy sample value (only if the sample was detected of this way) and the proposed filter can continue operating in all its sections in the same way. For this reason it is necessary to present the expression for the second detection block contained in the proposed filter structure [6]:

$$med'_{cw}(\tilde{\sigma}^2_{g_1}(j)) = \begin{cases} med(\tilde{\sigma}^2_{g_1}(j)) & \text{if } \tilde{\sigma}^2_{g_1}(k) \le \gamma\sigma^2_n \\ \tilde{\sigma}^2_{g_1}(k) + med(\tilde{\sigma}^2_{g_1}(j) - \tilde{\sigma}^2_{g_1}(k)) & \text{if } \tilde{\sigma}^2_{g_1}(k) > \gamma\sigma^2_n \end{cases} \qquad (14)$$

The proposed filter uses the median algorithm represented as $med(\tilde{\sigma}^2_{g_1}(j))$ to estimate the value of the central sample in a filter window if the sample is detected as noisy, and it is possible to use other estimation algorithms such as the Multi-Stage Median and FIR Median Hybrid (Filters that retain more information of the image.

The Multi-Stage Median (MSM) filter is based on the obtaining of the median of medians in such form [12, 11, 12]:

$$FMM = med\{h - med, v - med, d45 - med, d135 - med\} \qquad (15)$$

where the parameters of filter are displayed in the Fig. 2.

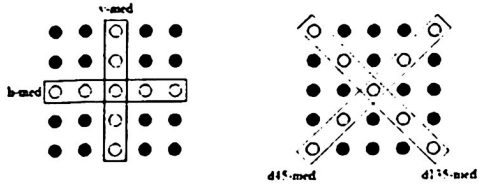


Fig. 2. Median Filters *h-med, v-med, d45-med, d135-med.*

2.6 Wavelet Domain FIR Median Hybrid Filter

There exist different types of these filters which offer the possibility of choosing the number of sub-filters, its type, as well as the weights and the type of window. In this case, a FIR Median Hybrid (FIRMH) filter is defined by means of use the filter windows shown in the Fig. 3 and the following equations [2, 13]:

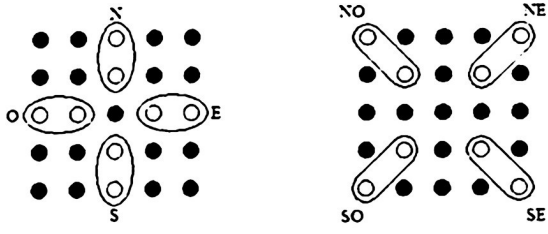


Fig. 3. Filter windows for the FIR Median Hybrid Filter.

$$F_1 = med\left\{Y_N, Y_E, Y_S, Y_O, \tilde{\sigma}^2_{g^3_{m+1,m+1}}\right\} \tag{16}$$

$$F_1 = med\left\{\frac{1}{m}\sum_{i=1}^{m}\tilde{\sigma}^2_{g^3_{i,m+1}}, \frac{1}{m}\sum_{i=m+2}^{z}\tilde{\sigma}^2_{g^3_{m+1,j}}, \frac{1}{m}\sum_{i=m+2}^{z}\tilde{\sigma}^2_{g^3_{i,m+1}}, \frac{1}{m}\sum_{i=1}^{m}\tilde{\sigma}^2_{g^3_{m+1,j}}, \tilde{\sigma}^2_{g^3_{m+1,m+1}}\right\} \tag{17}$$

$$F_1 = med\left\{\frac{1}{m}\sum_{i=1}^{m}\tilde{\sigma}^2_{g^3_{i,m+1}}, \frac{1}{m}\sum_{i=m+2}^{z}\tilde{\sigma}^2_{g^3_{m+1,j}}, \frac{1}{m}\sum_{i=m+2}^{z}\tilde{\sigma}^2_{g^3_{i,m+1}}, \frac{1}{m}\sum_{i=1}^{m}\tilde{\sigma}^2_{g^3_{m+1,j}}, \tilde{\sigma}^2_{g^3_{m+1,m+1}}\right\} \tag{18}$$

$$F_1 = med\left\{\frac{1}{m}\sum_{i=1}^{m}\tilde{\sigma}^2_{g^3_{i,m+1}}, \frac{1}{m}\sum_{i=m+2}^{z}\tilde{\sigma}^2_{g^3_{m+1,j}}, \frac{1}{m}\sum_{i=m+2}^{z}\tilde{\sigma}^2_{g^3_{i,m+1}}, \frac{1}{m}\sum_{i=1}^{m}\tilde{\sigma}^2_{g^3_{m+1,j}}, \tilde{\sigma}^2_{g^3_{m+1,m+1}}\right\} \tag{19}$$

$$F_2 = med\left\{Y_{NO}, Y_{SO}, Y_{SE}, Y_{NE}, \tilde{\sigma}^2_{g^3_{m+1,m+1}}\right\} \tag{20}$$

$$F_2 = med\left\{\frac{1}{m}\sum_{i=1}^{m}\tilde{\sigma}^2_{R^s_{i,j}}, \frac{1}{m}\sum_{i=m+2}^{m}\tilde{\sigma}^2_{R^s_{z-i+1,j}}, \frac{1}{m}\sum_{i=m+2}^{z}\tilde{\sigma}^2_{R^s_{i,j}}, \frac{1}{m}\sum_{i=m+2}^{z}\tilde{\sigma}^2_{R^s_{z-i+1,j}}, \tilde{\sigma}^2_{R^s_{m+1,m+1}}\right\} \quad (21)$$

$$F_3 = med\left\{F_1, F_2, \tilde{\sigma}^2_{R^s_{m+1,m+1}}\right\} \quad (22)$$

where z is the measurement in pixels of a side of the used square window in the analysis and m it is obtained from $m = \frac{z-1}{2}$, Y are simply average filters, F_1 and F_2 are median filters and F_3 is the result of FIR Median Hybrid Filter.

3 Simulation Results

We obtained from the simulation experiments the properties of the proposed filter by means of use the following criteria: the crossed correlation index (CCI) to evaluate the correlation between the original and filtered images, the peak signal noise relation (PSNR) and the mean square error (MSE) to evaluate the noise suppression, the mean absolute error (MAE) to evaluate the detail preservation, the mean chromaticity error (MCRE) to evaluate the chromaticity retention, and the normalized color difference (NCD) to quantify the perceptual error [1, 2]. In the simulations results that we present here, the image "Lena" was degraded with 20% of impulsive noise, the wavelet used for the analysis was the dB5, and with 3x3 filtering window.

The performance results obtained for the Wavelet Domain Order Statistics Filter (section 2) is shown in the Table 1.

Table 1. Performance results for the proposed filter.

Scale	CCI	PSNR(dB) Red channel	PSNR(dB) Green channel	PSNR(dB) Blue channel	MSE Red channel	MSE Green channel	MSE Blue channel
1	0.925930	29.759030 dB	31.189573 dB	31.602808 dB	[illegible]	[illegible]	[illegible]
2	0.971353	39.467616 dB	41.539996 dB	42.219814 dB	[illegible]	[illegible]	953.867441
3	0.984417	45.181795 dB	48.075334 dB	49.057513 dB	709.348935	531.125033	481.437204
4	0.987709	47.353714 dB	50.630914 dB	51.895372 dB	570.866701	411.346420	362.487470
5	0.988663	47.947656 dB	51.408423 dB	52.737680 dB	537.947796	380.880148	333.205385

Scale	MAE Red channel	MAE Green channel	MAE Blue channel	MCRE Red channel	MCRE in Green channel	MCRE Blue channel	NCD
1	34.941953	34.119675	34.309896	0.051001	0.044203	0.042414	0.480219
2	26.417234	24.171183	23.693334	0.019317	0.015701	0.014670	0.313019
3	20.736065	17.854867	17.026820	0.010909	0.008168	0.007404	0.227659
4	18.777056	15.732278	14.815044	0.008779	0.006326	0.005575	0.198971
5	18.215902	15.077929	14.170148	0.008273	0.005857	0.005124	0.191196

Tables 2 to 5 present the performance results to apply the proposed filter with only one detection block by means of use the SD-ROM, TSM, ACWM, MM-KNN algorithms, respectively.

The performance results for the use of a second detection block are shown in Tables 6 and 7 by means of use the MSM and FIRMH Filters, respectively. In these cases we use a 7x7 filtering window size.

One can see from the simulation results that the detection blocks (implementation of different filtering algorithms) provide good results in terms of noise suppression and detail preservation.

Finally, we can say that the proposed filters provide better properties in terms of CCI, PSNR, MSE, MAE, MCRE, and NCD in comparison with the performance of the thresholding Wavelet methods [3, 4] and Wavelet domain iterative center weighted median filter [6]. It can see in refs. [5, 14] where we demonstrate the better performance of the method of redundancy of approaches, and when we use it into the Wavelet domain iterative center weighted median filter in comparison with the filters described in [3, 4, 6]

Table 2. Performance results for the proposed filter using the SD-ROM Filter algorithm.

Scale	CCI	PSNR(dB) Red channel	PSNR(dB) Green channel	PSNR(dB) Blue channel	MSE Red channel	MSE Green channel	MSE Blue channel
1	0.925999	29.768870 dB	31.200992 dB	31.614584 dB	3313.161568	2871.033178	2754.713506
2	0.971379	39.476371 dB	41.549986 dB	42.231109 dB	1254.998935	1019.969956	952.810607
3	0.984424	45.184984 dB	48.080024 dB	49.062561 dB	709.122811	530.862115	481.194231
4	0.987710	47.354953 dB	50.632708 dB	51.896504 dB	570.795947	411.272604	362.446420
5	0.988663	47.947747 dB	51.400072 dB	52.737976 dB	537.942899	380.893521	333.195533
Scale	MAE Red channel	MAE Green channel	MAE Blue channel	MCRE Red channel	MCRE Green channel	MCRE Blue channel	NCD
1	34.921154	34.096479	34.284837	0.050951	0.044153	0.042364	0.479900
2	26.404142	24.158269	23.589541	0.019300	0.015686	0.014653	0.312853
3	20.732692	17.850251	17.022308	0.010905	0.008164	0.007400	0.227608
4	18.775680	15.730385	14.813994	0.008778	0.006325	0.005574	0.198958
5	18.215976	15.078284	14.169941	0.008273	0.005858	0.005124	0.191197

Table 3. Performance results for the proposed filter using the TSM Filter algorithm.

Scale	CCI	PSNR(dB) Red channel	PSNR(dB) Green channel	PSNR(dB) Blue channel	MSE Red channel	MSE Green channel	MSE Blue channel
1	0.925994	29.767953 dB	31.200629 dB	31.613487 dB	3313.405473	2871.139201	2755.015593
2	0.971377	39.475705 dB	41.549439 dB	42.230318 dB	1255.092470	1020.025769	952.886050
3	0.984423	45.184815 dB	48.079870 dB	49.062087 dB	709.134793	530.882189	481.217027
4	0.987710	47.354983 dB	50.632792 dB	51.896626 dB	570.794260	411.269157	362.442012
5	0.988663	47.947710 dB	51.400117 dB	52.737987 dB	537.944882	380.891820	333.195163
Scale	MAE Red channel	MAE Green channel	MAE Blue channel	MCRE Red channel	MCRE in Green channel	MCRE Blue channel	NCD
1	34.922722	34.096923	34.287249	0.050956	0.044154	0.042369	0.479918
2	26.405104	24.158935	23.590518	0.019302	0.015687	0.014654	0.312864
3	20.732959	17.850533	17.022530	0.010906	0.008164	0.007400	0.227611
4	18.775621	15.730340	14.813994	0.008778	0.006325	0.005574	0.198958
5	18.215947	15.078240	14.169926	0.008273	0.005858	0.005124	0.191197

Table 4. Performance results for the proposed filter using the ACWM Filter algorithm.

Scale	CCI	PSNR(dB) Red channel	PSNR(dB) Green channel	PSNR(dB) Blue channel	MSE Red channel	MSE Green channel	MSE Blue channel
1	0.925997	29.768585 dB	31.200745 dB	31.614228 dB	3313.196139	2871.104621	2754.811450
2	0.971378	39.476028 dB	41.549664 dB	42.230851 dB	1255.043408	1020.002855	952.835192
3	0.984424	45.185074 dB	48.080198 dB	49.062361 dB	709.116376	530.864734	481.203861
4	0.987710	47.354917 dB	50.632708 dB	51.896521 dB	570.798033	411.272604	362.445814
5	0.988663	47.947747 dB	51.400072 dB	52.737976 dB	537.942899	380.893521	333.195533
Scale	MAE Red channel	MAE Green channel	MAE Blue channel	MCRE Red channel	MCRE in Green channel	MCRE Blue channel	NCD
1	34.921967	34.097027	34.285769	0.050953	0.044154	0.042365	0.479911
2	26.404512	24.158920	23.589985	0.019301	0.015686	0.014653	0.312857
3	20.732559	17.850266	17.022382	0.010905	0.008164	0.007400	0.227609
4	18.775695	15.730385	14.814068	0.008778	0.006325	0.005574	0.198958
5	18.215976	15.078284	14.169941	0.008273	0.005858	0.005124	0.191197

Table 5. Performance results for the proposed filter using the MM-KNN Filter algorithm.

Scale	CCI	PSNR(dB) Red channel	PSNR(dB) Green channel	PSNR(dB) Blue channel	MSE Red channel	MSE Green channel	MSE Blue channel
1	0.925931	29.759522 dB	31.190093 dB	31.603201 dB	3316.200178	2872.883571	2756.315533
2	0.971354	39.467824 dB	41.540249 dB	42.220384 dB	1256.071982	1022.963586	953.833107
3	0.984417	45.181801 dB	48.075928 dB	49.057611 dB	709.348521	531.091364	481.432500
4	0.987709	47.353817 dB	50.631474 dB	51.895347 dB	570.860769	411.323573	362.488358
5	0.988663	47.947501 dB	51.400560 dB	52.738011 dB	537.956139	380.874941	333.194364
Scale	MAE Red channel	MAE Green channel	MAE Blue channel	MCRE Red channel	MCRE in Green channel	MCRE Blue channel	NCD
1	34.941124	34.118491	34.308979	0.051999	0.044201	0.042412	0.480207
2	26.417012	24.170621	23.603521	0.019317	0.015701	0.014669	0.313013
3	20.735917	17.854127	17.026790	0.010909	0.008167	0.007404	0.227653
4	18.776923	15.731746	14.815074	0.008779	0.006326	0.005575	0.198967
5	18.216050	15.077544	14.169926	0.008273	0.005857	0.005124	0.191195

Table 6. Performance results for the proposed filter using the MSM Filter estimation algorithm.

Scale	CCI	PSNR(dB) Red channel	PSNR(dB) Green channel	PSNR(dB) Blue channel	MSE Red channel	MSE Green channel	MSE Blue channel
1	0.925943	29.758344 dB	31.194552 dB	31.608770 dB	3316.590962	2872.883571	2756.315533
2	0.971380	39.473387 dB	41.552420 dB	42.233572 dB	1255.373417	1019.721731	952.575976
3	0.984427	45.181500 dB	48.085745 dB	49.066113 dB	709.369882	530.570355	481.023343
4	0.987714	47.351208 dB	50.639627 dB	51.902085 dB	571.069734	410.988151	362.244201
5	0.988668	47.944820 dB	51.408736 dB	52.744168 dB	538.100340	380.563669	332.989290
Scale	MAE Red channel	MAE Green channel	MAE Blue channel	MCRE Red channel	MCRE in Green channel	MCRE Blue channel	NCD
1	34.945281	34.113713	34.302722	0.051005	0.044181	0.042389	0.480189
2	26.410725	24.158269	23.589763	0.019306	0.015682	0.014649	0.312862
3	20.737751	17.845976	17.019172	0.010909	0.008159	0.007398	0.227575
4	18.780680	15.725163	14.809645	0.008781	0.006320	0.005571	0.198914
5	18.219778	15.071509	14.165562	0.008275	0.005853	0.005121	0.191141

Table 7. Performance results for the proposed filter using the FIRMH Filter estimation algorithm.

Scale	CCI	PSNR(dB) Red channel	PSNR(dB) Green channel	PSNR(dB) Blue channel	MSE Red channel	MSE Green channel	MSE Blue channel
1	0.925943	29.758344 dB	31.194552 dB	31.608770 dB	3316.590962	2872.883571	2756.315533
2	0.971380	39.473387 dB	41.552420 dB	42.233572 dB	1255.373417	1019.721731	952.575976
3	0.984427	45.181500 dB	48.085745 dB	49.066113 dB	709.369882	530.570355	481.023343
4	0.987714	47.351208 dB	50.639627 dB	51.902085 dB	571.069734	410.988151	362.244201
5	0.988668	47.944820 dB	51.408736 dB	52.744168 dB	538.100340	380.563669	332.989290
Scale	MAE Red channel	MAE Green channel	MAE Blue channel	MCRE Red channel	MCRE in Green channel	MCRE Blue channel	NCD
1	34.945281	34.113713	34.302722	0.051005	0.044181	0.042389	0.480189
2	26.410725	24.158269	23.589763	0.019306	0.015682	0.014649	0.312862
3	20.737751	17.845976	17.019172	0.010909	0.008159	0.007398	0.227575
4	18.780680	15.725163	14.809645	0.008781	0.006320	0.005571	0.198914
5	18.219778	15.071509	14.165562	0.008275	0.005853	0.005121	0.191141

4 Conclusions

We present different implementations of order statistics filters in wavelet domain. The proposed filters constitute a good tool to the impulsive noise filtering in color image applications. It is obviously that the use of different order statistics filters into the proposed block detection in the wavelet domain can increase the processing time.

Acknowledgment. To National Polytechnic Institute of Mexico for all the given facilities to carry out this paper.

References

1. S. K. Mitra and G. L. Sicuranza, Nonlinear Image Processing, Academic Press, San Diego, CA, 2001, pp. 111-133.
2. J. Astola and P. Kuosmanen, Fundamentals of Nonlinear Digital Filtering, CRC Press, Boca Raton-New York, 1997.
3. Z. Cai, T.H. Cheng, C. Lu, K.R. Subramanium, Efficient wavelet-based image denoising algorithm, Electron. Lett. 37(11) 683-685, 2001.
4. S.G. Chang, B. Yu, M. Vettereli, Adaptive wavelet thresholding for image denoising and compression, IEEE Trans. Image Process. 9 (9) 1532-1546,.2000.
5. J. Martínez Valdes, F. J. Gallegos Funes, M. A. Acevedo Mosqueda, Reducción de Ruido Impulsivo en Imágenes Digitales a Color Utilizando Wavelets, Memorias de IEEE 15 Reunión de Otoño de Comunicación y Exposición Industrial ROC&C 2004, Acapulco, México, Noviembre 2004.
6. S. M. Mahbubur Rahman, Md. Kamrul Hasan, Wavelet-domain iterative center weighted median filter for image denoising, Signal Processing, 83, 1001-1012, 2003.
7. T. Chen, K. Ma, L. Chen, Tri-State Median Filter for image denoising, IEEE Trans. Image Process. 8 (12), 1834-1838, 1999.
8. T. Chen and H. R. Wu, Adaptive impulse detection using center-weighted median filters, IEEE Signal Processing Letters, 8(1), 1-3, 2001.
9. F. Gallegos, V. Ponomaryov, O. Pogrebnyak y L. Niño de Rivera, Filtros Robustos RM-KNN con Diferentes Funciones de Influencia para Supresión de Ruido Impulsivo en Imágenes Digitales, Computación y Sistemas, Vol. 6, No. 3, pp. 183-195, 2003.
10. F. Gallegos, V. Ponomaryov, Real-time image filtering scheme based on robust estimators in presence of impulsive noise, Real Time Imaging, 8(2), 78-90, 2004.
11. G.R. Arce, M. P. McLoughlin, Theoretical analysis of the max/median filter, IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-35, 60-69, Jan. 1987.
12. J. Astola, P. Heinonen, Y. Neuvo, Linear median hybrid filters, IEEE Trans. Circuits and Systems, vol. CAS-36, 1430-1438, Nov. 1989.
13. P. Heinonen, Y. Neuvo, Smoothed median filters with FIR substructures, Proc. 1985 IEEE Int. Conf. Acoust. Speech and Signal Processing, Tampa, March 1985, 49-52.
14. J. Martinez Valdes, F. Gallegos-Funes, M. A. Acevedo-Mosqueda, "Filtro Mediano con Peso Central Iterativo en el Dominio Wavelet utilizando el Método de Redundancia en las Aproximaciones", 6to. Congreso de Computación, CORE 2005, Research on Computing Science 13, Advances in Computer Science in Mexico, pp. 41-49, IPN, CIC, Mexico, D. F., Mayo 2005.

Microcontroller-Based Distributed Serial Communication and PC-Based Supervision and Control for an Automated Greenhouse

Laura E. Muñoz H.[1], Luis A. Paredes S. [2], Julio Romero G.[3], América Saldaña S[4], J. Antonio Sánchez H.[5], Julio C. Ramos Fernández[6], Virgilio López Morales[7]

Center for Research on Systems and Information Technologies, ICBI-UAEH,
Carr. Pachuca Tulancingo Km. 4.5, C.U., C.P. 42084, Pachuca Hidalgo MEXICO.

e-mail : [1] lauris3112@gmail.com , {[2] albert8314, [4] ameri_kas}@hotmail.com, [6] jramos@uttt.edu.mx, {[5] his_eternal_majesty, [3] jediknight_fisto}@yahoo.com.mx, [7] virgilio@uaeh.edu.mx

Abstract. In order to have a better and bigger plants production, it is necessary to have a regulated micro-environment, in a greenhouse. Inside a hydroponics greenhouse it is possible to regulate the main physical variables (temperature, humidity irrigation cycles, etc.) as well as the variables involved in the rates of plants growing (nutrients, sun radiation, water, etc.). Then, a greenhouse must be constantly supervised in order to regulate the main physical variables which are involved in the production process. For instance, temperature and humidity can be regulated by opening or drawing the ventilation system or domes or by heating or circulating the air inside the greenhouse. These tasks should be supervised all days, and it demands constantly, a supervised distributed communication system. In this paper we proposed a distributed serial communication system based on microcontrollers to monitoring and supervising the process via Internet. With this system, it can be achieved the on-line supervision and control of the greenhouse operation and conditions meteorological, as well as the main variables (temperature, humidity irrigation cycles, etc.).

Keywords: Greenhouse, Virtual Instrumentation, Supervision, Microcontroller

1 Introduction

Greenhouses can provide an excellent controlled environment for plant production. The greenhouse should provide uniform lighting, heating, and water to all plants. For this reason is important to have a system for supervising all this variables in order to control them, thus it can cultivate plants in excellent conditions. The advantages that offer us the greenhouses production are enough to consider automating it. These advantages are: low-

costs productions, better control of pests, better quality, save of water and more than one crop per year.

Furthermore, the fast evolution of the Personal Computer (PC) in the last two decades generated a revolution in virtual instrumentation for test and measurement. Virtual instrumentation offers several benefits to engineers and scientists who require increased productivity, accuracy, and performance (National Instruments, 2005).

A virtual instrument consists of an industry-standard computer or workstation equipped with powerful application software, cost-effective hardware such as plug-in boards, and driver software, which together perform the functions of traditional instruments (National Instruments, 2005).

Our objective in this paper is to design a minimum system using the PIC16F877 microcontroller in order to create a distributed serial communication system and a resource network to control and supervise via a website. The main goal of the minimum system called Slave is the acquisition and digital conversion of analog signals, to be sent to another minimum system called Master. The Master sends these signals to the PC that achieves three main tasks: stores the acquired data in a database, keeps a Domain Name Server (DNS)[1] finally generates the user interface. The minimum systems control the actuators of the system and can achieve the acquisition of the physical variables. These variables are digitals and analogical signals, such as speed and wind direction, solar radiation, internal and external temperature of the greenhouse, internal and external humidity, rain detector, heating actuator and shadow opening (Figure 1).

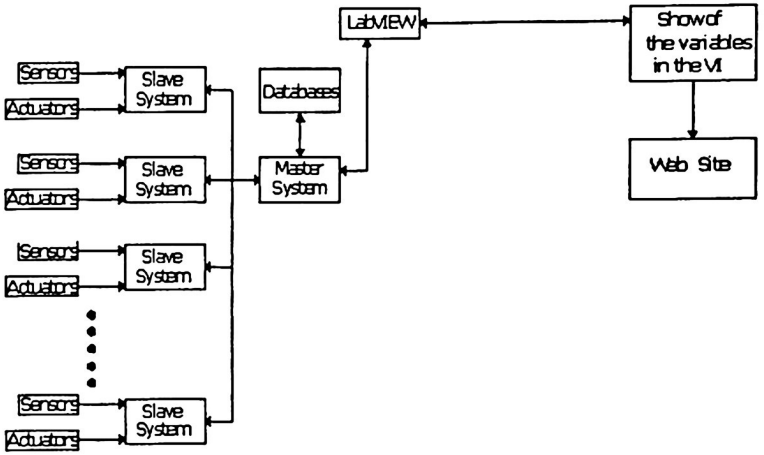


Fig. 1 Block Diagram

[1] Domain Name Server is an Internet service that translates *domain names* into IP addresses. We used this server to public our own page in the Web.

The slave minimum systems are interconnected by a RS485 protocol to the PC through Master-Slave architecture.

We use the RS485 Serial Communication Interface, since it allows us a multipoint communication (Table 2) and others relevant characteristics. Thus, it is necessary to realize a conversion from the RS232 Serial Communication Interface (shipped on PIC16F877) to the RS485 bus protocol.

Table 1. Comparison of RS-232, RS-422, and RS-485 Serial Communication Interfaces

Specifications	RS-232	RS 442	RS 485
Mode Operation	Single-Ended	Diffrential	Differential
Total Number of Drivers and Receivers on One Line (One driver active at time for RS-485 networks)	1 Driver 1 Receiver	1 Driver 10 Receivers	32 Drivers 32 Receiver
Maximum Cable Length	50ft (2500 pF)	4000 ft	4000 ft
Maximum Data Rate (40ft.4000ft for RS-422/RS-485)	20KB/s (by spec-can be higher)	10 Mbits/s	10 Mbits/s

2 System Overview

In this section, it is shown in detail the main components and devices used in our complete system. The system consists of two main parts: the PC connected to the Internet and the Master-Slave System (Figure 2).

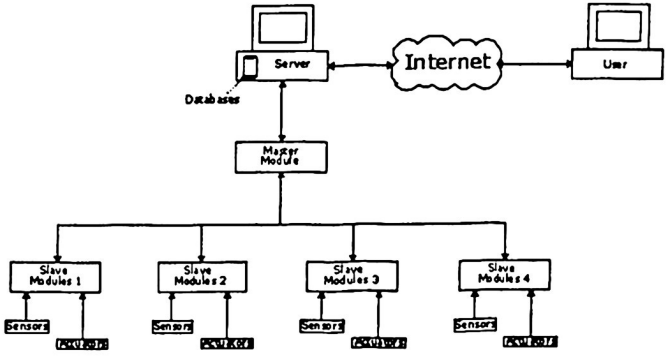


Fig. 2 General Structure

The physical variables are measured by the Sensors and then sent by the Slaves Module to the Master Module and then retransmitted to the PC.

The PC stores in a database the physical variables. In addition keeps the Web server, displaying the data via Internet.

3 Master-Slave system

The minimum system developed has eight analog inputs where the signals of the sensors are connected to be converter in digital signals using the Analog-Digital Converter Module. After the conversion the values are stored in a variable and it is ready to be sent to the master and then to the PC. The communication between devices is through the Addressable Universal Synchronous Asynchronous Receiver Transmitter (USART). In this case the USART[2] is configured as a full duplex asynchronous system and the baud rate is 9600.

Each Slave has a nine bits direction to be identified, thus the master can send the user request to the correct slave, and then it is useful to prevent errors between information transmission and request from the master to the slaves. The communication between master and PC is within eight bits since the LabVIEW only allows eight bits transmission.

When a request arrives to the correct slave, this receives code (from the master) which indicates the information that the slave has to transmit. This information comes from the temperature sensor, humidity sensor and solar radiation sensor. Thus the user can supervise the greenhouse and control these variables by sending control signals to the actuators connected to the slaves according to the control law programmed in LabVIEW.

Moreover the analog input module, the SCI module (transmission- receive) and the digital outputs module (to the actuators), the minimum system has the necessary implementations to be used in our system. These implementations are: the keyboard, the display, one free port and the Master Synchronous Serial Module. The display it is used to locally supervise some parameters in the same module instead of going necessarily to PC. The others shipped-on modules are not been used at this moment, but they will integrated in a future development works.

The linkage of these digital devices and components must be made by some software components. We have used some commercial packages and developed all the algorithms and VIs to have an open architecture to perform easily different control and supervision schemes.

---

[2] USART Asynchronous mode uses standard non-return-to zero (NRZ) format (one START bit, eight or nine data, bits, and one STOP bit). The USART transmits and receives the LSb first. The transmitter and receiver are functionally independent, but use the same data format and baud rate [4].

4. Software Features

In order to keep some compatibility among the different devices and software components and reliability of the proposed architecture, it was chosen some commercially software as described in the following.

4.1 LabView

LabView is a graphical programming language that uses icons instead of lines of text to create applications; furthermore, it allows achieving different forms of communication between a PC and external digital devices. This communication can be made by a serial or a parallel port, and all of this is programmed under an easy graphic user interface. Since the master and slaves, based on the microcontroller 16F877, have two serial ports, we have chosen a LabView based on serial communication interface.

The virtual instrumentation is achieved via virtual instruments (VI). This VI, defined by LabView, seems and operates as a physical instrument, such as an oscilloscope or a multi-meter. Furthermore, it can be used a VI in another VI, it is called a subVI.

In this work, each VI is used to manipulate functions for example the VI presents data on the screen or sends it to another files or PC's (Figure 3).

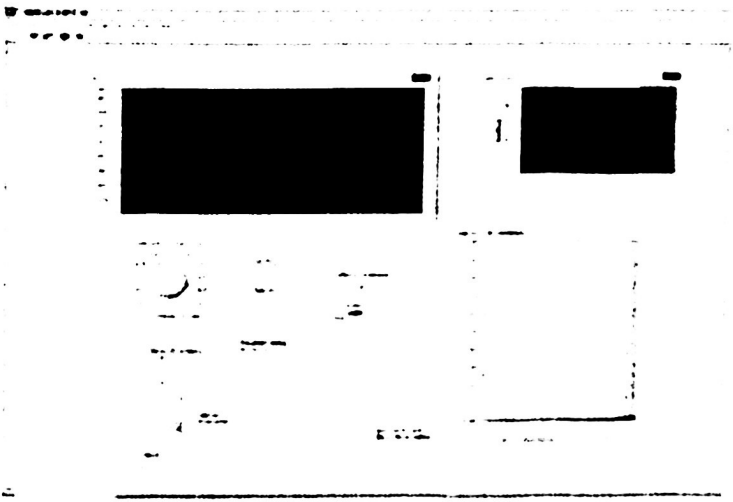


Fig. 3 LabView graphic interfaces

For the database, the data came from the Master and is directly saved in an Excel file due to the LabView input output format requirement. Thus, the created file is called for another program which manages databases (MySQL).

4.2 MySQL and PHP

MySQL is a compact database server ideal for small applications. In addition to supporting standard SQL[3] (ANSI), it compiles on a number of platforms and has different operating systems multithreading abilities on UNIX servers[4], providing a great performance.

This software is used to import information from an Excel file to a database admitted by PHP[5] software. PHP can access most any SQL or ODBC database. It can both read and write information in the database.

4.3 Dreamweaver

The visual editing features in Dreamweaver[6] allow us create some pages without writing a line of code and it can view all the site elements or assets and drag them from an easy-to-use panel directly into a document. We can streamline the development workflow by creating and editing images in Macromedia Fireworks or another graphics application, then importing them directly into Dreamweaver, or by adding Macromedia Flash objects.

How it was mentioned, Dreamweaver is only a tool that helps to the design of the Web page, and can be shown in a Web navigator in any kind of platform with Internet connection.

5 Website

The website (Figure 4) shows pictures for the crop plantation and the greenhouse construction architectonic, the different access to read the physical variables databases, and the LabView virtual instruments to supervise the different modules.

---

[3] The SQL part of MySQL stands for "Structured Query Language" - the most common standardized language used to access databases

[4] Red Hat 9 Linux is the leading platform for open source computing.

[5] (Personal Home Page) Hypertext Preprocessor

[6] Macromedia Dreamweaver MX 2004 is a professional HTML editor for designing, coding, and developing websites, web pages, and web applications. It allows us the control of hand-coding HTML or a visual editing environment

The website involves a DNS Server and a Dynamic Host Configuration Protocol (DHCP) due to the necessity of a DNS server to have a communication with Internet, since this convert the domain name to an IP address used in the TCP/IP protocol. The DHCP allows us the generation of an IP address automatic (to the user connected in the network) in a range of IP addresses, thus it is viable to have a multipoint network instead of a point to point network.

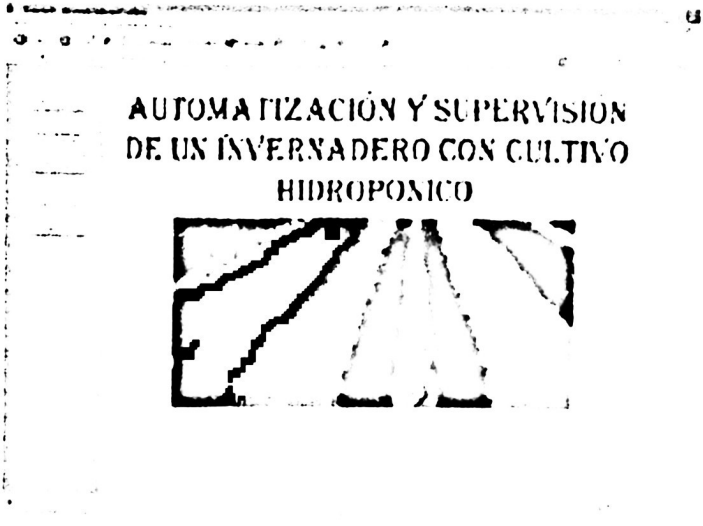


Fig. 4 Web Site

6 Conclusions

Based on the PIC 16F877 microcontroller, we have proposed a distributed serial communication system to supervise the external variables and control the internal variables in a greenhouse. The serial communication protocol used is RS485, since it allows us a multipoint communication and others relevant characteristics (number of receivers and cable length). Also some communication failure test has been carried out in order to check the robustness performance.

References

[1] LabView Basics II Course Manual, 2000, National Instrument, September 2000, Edition, Texas, USA.
[2] Using de Dreamweaver MX, 2004, Macromedia, September 2003, Ed. San Fancisco, USA.
[3] Red Hat Linux Getting Started Guide, 2003, Red Hat, Inc.
[4] LabView User Manual, 2003, National Instrument, April 2003, Edition, Texas, USA.
[5] MySQL Reference Manual, 2001.
[6] Serodio Carlos, Boaventura Cunha, J., Morais Raul, Couto Carlos, Monteiro Joao. A networked platform for agricultural management system. 2001. Computers and Electronics in Agricultural.
[7] Fuertes, J.M., Herrera, J., Arboleda, J.P., Heit, F., Casas, C., Company, J. Communication system for a distributed intelligent controller. 1999. Microprocessors and Microsystems
[8] Tipsuwan, Y., Chow Mo-Yuen. Control Methodologies in networked control system. 2003. 2003. Control Engineering Practice.
[9] Gieling, Th.H., Van Meurs, W. Th., Jansen, J. A computer network with SCADA and case tools for on-line process control in greenhouses. 1996. Pergamon.
[10] Boaventura Cunha, J., Moura Oliveira, J.P. Optimal management of greenhouses environments. 2003. EFITA 2003 Conference.

Design of a Nanosatellite Laboratory Model as a Proof of Concept for a Future University Space Mission

Dr. Esaú Vicente-Vivas[1], Jorge Prado-Molina[2], Dr. Salvador Landeros[3], Dr. Gianfranco Bisiacchi[4], Jorge Morales[5] and Arturo Rueda[6]

[1]Instituto de Ingeniería, UNAM, [2]Instituto de Geografía, UNAM, [3]Facultad de Ingeniería, UNAM, [4]Centro Tecnológico Aragón, FES Aragón, UNAM, Edo de México, [5] DEPFI-UNAM

Universidad Nacional Autónoma de México Cd. Universitaria Coyoacan, 04510, México DF.,

e-mail:evv@servidor.unam.mx, jprado@igiris.igeograf.unam.mx, diego@fi-b.unam.mx, jmorales@iingen.unam.mx, {bdruan, rusarturo}@yahoo.com

Abstract. A brief review of Microsatellite activities in Mexico and the world-wide technological tendencies in the field of small Satellites are presented in this paper. Afterwards, the proposal to design and develop a Nanosatelite (NS) laboratory model is presented. The NS model will evolve gradually towards a 3.5 Kgs Mexican Nanosatelital project, attractive in terms of development time as well as in manufacture and launching costs. Later, the preliminary design of NS subsystems and its current validation performed in laboratory are indicated. Finally, comments about the required steps to reach an operative model are also given.

Keywords: Nanosatellite, Instrumentation, laboratory model, satellite subsystems, satellite design.

1 The national experiences in the microsatellite field

Mexico has made two important efforts towards the development, launching, and operation experimental Microsatellites. They tried to call the attention of the country's political class and the society in general. Both efforts intended to form qualified human resources and to show the in-house development capacities in Science and Technology. In addition, the initiatives uncovered the opportunities and benefits to develop a national space program which might help to solve problems related with communications, national resources monitoring, disasters evaluation, and so on.

The first initiative was the UNAMSAT project, technically supported by the AMSAT International Organization to achieve the development of a fast mission. This project

incremented the added-value of the 17 Kgs, [1], by designing and constructing an astronomical payload intended to research the amount of dark matter existing in the universe. Unfortunately the mission could not reach enough operational results in space.

The second effort was the Satex project, which aimed the development of a fully designed and manufactured 50 Kg domestic Microsatellite, [2] and [3]. The project contemplated several payloads in communications, remote sensing, and fault-tolerant hardware and software, [4]. The ambitious goals, the difficulties to operate a project distributed in several universities and research institutions from all over the country, as well as the budget limitations, avoided, until now, the project to be completed at all.

It is important to highlight that both projects intended to enhance future national activities in the area of satellite technology. They expected to push activities towards projects of greater added-value to offer High-Tech solutions and services in the medium term. However, as Mexico has not long term technological policies in this field, the activities in this area have depended and will depend on the successes of small projects like the referred ones. Part of the problem has been magnified by the economic complications experienced by the country, which have taken to the reduction of financial support for Science and Technology activities. This, in turn, has increased the competition between research groups in the search for budgetary resources, and it has affected significantly emergent areas that do not even have great successes or results that allow them to attract important financial support, like in the case of satellite technology.

1.1 International trends in the small satellite area

In the last two decades the information technologies have evolved to such a degree that the capacities of a 50 kgs Microsatellite of the 90s can now be replaced with a 5 kgs Nanosatellite. This takes us to establish that in one decade the small satellite systems have managed to shrink their mass in an order of magnitude carrying out similar capabilities. Two outstanding project are the 6 kgs SNAP-1 first three-axis stabilized Nanosatellite developed by the University of Surrey, UK, [5], and the 3 kgs Quakesat Nanosatellite fabricated by the University of Stanford to investigate the early detection of earthquakes, [6]. However, many interesting technology validation Nanosatellites are now under development, [7].

It is important to notice that a good number of NS and Picosatellites (PS) projects around the world are taking advantage of the experiences acquired in the development of previous missions elsewhere in the world. In this way, many projects successfully build and operate very low cost satellites which are developed at Universities by academic staff in collaboration with undergraduate and postgraduate students, [8], [9], and [10].

This working scheme has allowed the University of Surrey (UoS), UK, and his local enterprise Surrey Satellite Limited (SSTL), to become the worldwide recognized international leaders in the small satellites arena, [11]. In this way, few of their major and successful developments are: Snap-1, the first three axis stabilized Nanosatellite; Topsat, the first high resolution remote sensing 120 Kgs satellite, [12]; and the world's first

first high resolution remote sensing 120 Kgs satellite, [12]; and the world's first satellite constellation for disaster monitoring, [13] and [14].

In addition, UoS is currently developing projects that will lead the field in the next years with projects like: Gemini, a low-cost geostationary Minisatellite platform based in SSTL's heritage of LEO technologies, [15]; constellations for Earthquake prediction; Synthetic Aperture Radar Minisatellites; Infrared Imaging Minisatellites; Microsatellites and Minisatellites for communications; and swarms of Nanosatellites, [16].

Nevertheless, universities and international organizations from other countries also work in a competitive way in the small satellite field such as: Space Systems Development Laboratory from the University of Stanford, USA; University of Toronto Institute for Aerospace Studies from Canada; Moscow Aviation Institute from Russia; Stutgart University from Germany; Indian Space Research Organization from India; Japan; China; Brazil, among others.

2 The beginnings of the Nanosatellite project

Participants of the Satex project motivated by the successfully results obtained in the developing of on-board instrumentation and operational software for the mission, took the opportunity to transfer the generated know-how to the development of a small sized Nanosatellite of about 3.5 Kgs of mass, figure 1.

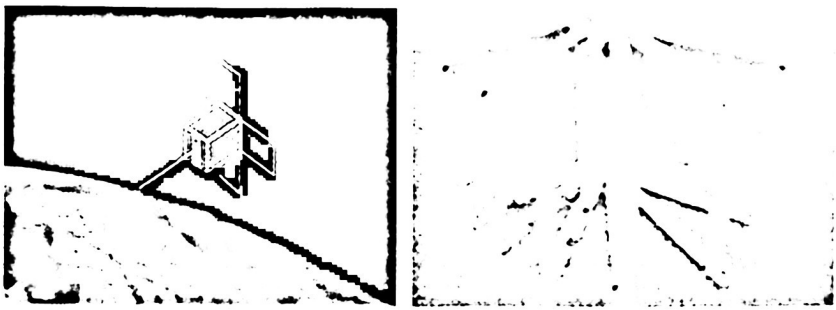


Fig. 1 a) Artistic view of the Nanosatellite platform,
b)Nanosatellite structural subsystem composed of aluminium frames

The initiative started in March 2004 through a postgraduate lecture imparted at the School of Engineering (DEPFI) from the National Autonomous University of Mexico (UNAM). The lecture is still going on under the title "Analysis and Design of Nanosatel-

lites and Picosatellites". The initial subjects consisted in the study and analysis of worldwide important papers in the fields of Microsatellites, NS and PS. Followed by the presentation and discussion of the main ideas to construct a Nanosatellite vehicle based on the Satex mission experiences. Afterwards, the lecture was oriented to the Nanosatellite subsystem analysis, parts selection, design, and evaluation of projected solutions. In some cases digital simulation of circuits was performed with the support of commercial software tools. A key issue was the validation in laboratory of some proposed electronic solutions. In our case, validation was performed when possible, accordingly with parts availability. In addition, the class was requested to write a final report expected to facilitate the insertion of newcomers to the project through the continuity of the lecture at DEPFI, UNAM.

The design process employed in the project took advantage of international trends for both the Nanosatellite and Picosatellite fields, regarding the widely use of commercial-off-the-shelf (COTs) electronic parts, [17]. In order to protect them from the space environmental conditions, electronic and physical shielding was considered. It was also adopted the international trend to extensively employ automotive as well as personal communications electronic parts, because they share important characteristics with space qualified components. Besides, COTs parts are easier to obtain and cheaper when compared against military or space qualified parts. Among the good qualifications of automotive electronic parts are the followings: temperature range, shock tolerance, low power, small package and CMOS technology availability which is the best suited for space applications by its radiation tolerance when used in low earth orbit, [18].

Even though the main goal is to develop a Nanosatellite proof of concept design it was decided to include as much as possible good quality parts and protections to ease the transition to a Nanosatellite flight prototype.

3 The nanosatellite project

Under this scenario, the Nanosatellite proof of concept design constitutes an effort that assimilates some of the domestic experiences in order to promote a new Mexican satellite initiative. The last takes advantage of new available technologies to offer a very light satellite platform with competitive capabilities compared to those from Microsatellites developed in the 90s.

Additional project goals towards the generation of an attractive Nanosatellite platform are the reduction of: development time (1 year in average), manufacture cost ($ 60,000 USD, excluding the payload cost), and launching cost accordingly with the vehicle mass. This is a very important issue to balance the technological goals of the project with the objectives from the potential supporters (economical, political, etc.).

The subsystems inherited from the Satex mission are the followings: a flight computer, a structural subsystem reconverted to be used as Nanosatellite structure, sensors (mag-

netic field, temperature, current, and voltage), as well as software for both satellite operations and Earth Station, [2]. It must be highlighted that the development of the operations software for small satellites is a very time consuming activity, characterized by long and continuous working sessions. In this way, according with comments from experts in the satellite field, software development for a small satellite implies about the 70 % of the workload for the whole project, [19]. So, the possibility to use or to adapt available software can importantly reduce the project workload as well as the mission development time.

3.1 The Nanosatellite Platform

The Nanosatellite platform includes all the necessary support subsystems for the satellite. In other words, the platform allows the space vehicle to provide the service for which it was made-up. For that reason, the Nanosatellite platform is composed by: structural subsystem, power subsystem, communications subsystem, flight computer, telemetry sensors, as well as room and resources (energy, communications, and automation capabilities) to lodge different payloads.

Once the platform becomes integrated with payloads, operative software, and the terrestrial segment, a complete satellite system is obtained.

3.2 The Nanosatellite laboratory model

The main objective of the NS proof of concept is to attract the attention of both potential financial supporters and academic authorities. The goal will be to continue the growth of the laboratory model in order to generate a NS engineering model. This is the reason why the current laboratory model does not include all of the NS subsystems. However, it contains few of the harder to accomplish satellite subsystems such as: the satellite operations software, earth station software, on-board computer, and part of the power subsystem. On the other hand, the paper design of the NS model is more advanced than the NS laboratory model for obvious reasons.

3.2.1 Structural Subsystem

The Nanosatellite structure constitutes an inheritance from the Satex project, where it was employed as the container of a reconfigurable flight computer. The structural subsystem, apart from offering physical protection to the NS subsystems, will connect the NS to the launching system through a mechanical interface. Figure 1b) shows the NS structure formed by aluminium frames designed to be connected one after each other in a piggy-back fashion.

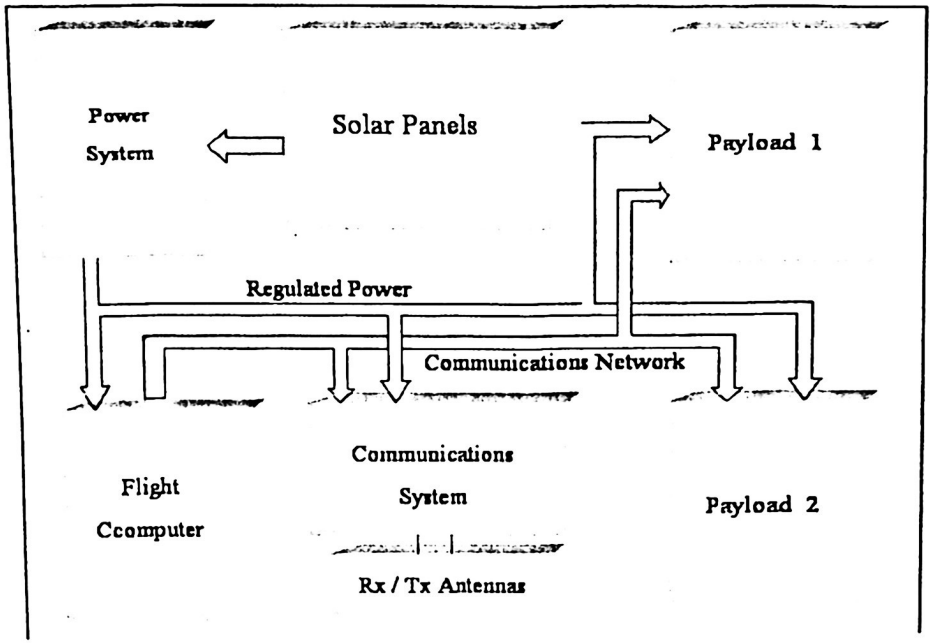


Fig. 2 Block diagram of NS subsystems

The first NS frame will be dedicated to the power subsystem along with telemetry sensors. The second will contain the communications subsystem, and the third frame is assigned to the flight computer, whereas the remaining frames can be assigned to hold satellite payloads (payload frames can be replaced both in amount or size according with mission needs). Once the NS is assembled its body dimensions can be 18x13x10 cm, depending on the payload needs.

Once the satellite body is assembled, four of the main NS faces will be covered with solar panels attached with stainless steel screws. The two remaining NS walls are let empty, whereas one of them is employed to accommodate the mechanical interface with the launching system.

Considering that one of the most important restrictions in satellites is its available power, the NS design considers the incorporation of 4 deployable double-sided solar wings. The wings will have a spring type mechanism that in orbital conditions will hold the panels in an orthogonal position with respect to the satellite walls. In addition, the spring mechanism will allow the panels to be parallel folded to the satellite walls with the application of a handy compression to the panels. The solar wings will be held in that

closed position by means of a nylon string. Once in orbit, the flight computer will send an order to burn the nylon string generating the automatic deployment of solar wings to the position shown in figure 1.

3.2.2 Power Subsystem

The NS power subsystem is formed by following three modules: generation, battery charge regulation, and power distribution. The generation module will charge a set of military qualified Li+ batteries. The module is formed by the solar panels which will be fabricated in our facilities, starting either with a base of space aluminium or composite material. The Nanosatellite will contain twelve solar arrays, figure 2, four will be double sided deployable solar panels and four will be directly mounted to the satellite body. The solar panels will employ GaAs COTs solar cells from Boreal Laboratories Ltd., whose cells are designed and utilized by the space industry. Each cell has the following characteristics: dimensions of 2x2.5 cm, 0.5 V output, and 100 mAmp/cm^2 at the beginning of life. More detailed information regarding the electronics of the power generation module, the battery charging scheme, and the power distribution unit of this subsystem is found in [20].

3.2.3 Flight Computer

The flight computer (FC) is a heritage from the Satex project. The hardware and software for this subsystem was fully generated through several years of work at UNAM, figure 3

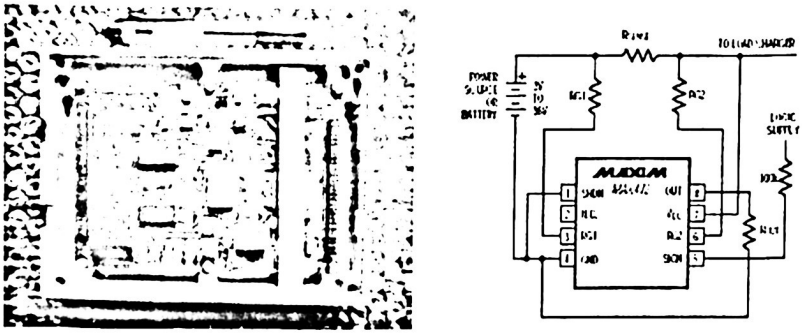


Fig. 3 a) NS flight computer installed in its aluminium frame,
b) Selected current sensor for the UN

As it is known, space applications have very severe restrictions of design; particularly in terms of weight, volume, and electrical available power. In addition, other common difficulties with all types of spacecrafts should be taken into account: vibration during the launching, the effects of radiation, [21], and vacuum, as well as the extreme temperatures, among others. All these constraints were considered during the development of the FC for both hardware and software.

The FC is a 5 volts stand-alone microcomputer based in the COT Industrial 16 bit RISC processor from Siemens. The processor has extended range of temperature and is latch-up protected with dedicated electronics. In addition, the FC memory is protected against data losses produced by Single Event Upsets (SEUs), in this case with the help of a military qualified error detection and correction (EDAC) unit, 29C516E from TEMIC.

The FC also contains the following peripherals: 64 Kb ROM for NS basic software, 1.2 Mb of SRAM protected by EDAC, 40 MHz military qualified oscillator, military qualified digital logic, watch dog timer, interrupt controller, three serial channels, 10 bits A/D conversion channels, 76 I/O lines, as well as a high quality multi-layer PCB.

3.2.3.1 Flight Software

The on-board already available software provides facilities for: up-loading new applications software, orbital management functions, telemetry acquisition, telemetry handling, telemetry packaging, telemetry transmission, command reception, command processing, mission reception, communications with ground, EDAC software, and watch-dog software, [22].

3.2.4 Sensors

The NS laboratory model will be equipped with telemetry sensors that will provide operative information regarding the satellite platform. It must be highlighted that one of the usual problems faced in satellite instrumentation is related with the amount of cables going and coming among the satellite equipment. This inconvenience is particularly present during the sensor wiring stage, because in traditional instrumentation there are at least two cables by sensor. If we take in to account that, by instance, a small Nanosatellite needs around 40 sensors, there would be required to place 80 cables all over the satellite PCBs. To avoid this problem NS employs 1-wire sensor technology, [23]. This allows saving weight and room in the platform. The 1-wire technology allows digital communications for data exchange among the flight computer and the sensors, therefore eliminating the insertion of noise in telemetry readings. Each 1-wire device contains a serial number formed by 64 bits for identification within a single 1-wire bus. Although, the connection among all the sensors can be done over one single 1-wire bus, the UN employs two

of them for clearness reasons. One of them will be in charged of all the temperature sensors and the other will be in charged of the voltage and current measurements.

Experiences from successful small satellite projects show that the expected internal temperature from an operative satellite in orbit is between 0 and 5 degrees Celsius. This value is obtained with the help of external heat protections of the satellite, as well with the internal heat dissipation facilities which force the physical contact among the satellite equipment.

The DS18S20Z 1-wire temperature sensor will be used aboard UN. It operates in a rank of temperatures from -55 to +125 degrees Celsius, with a 9 bits resolution. Each UN printed circuit board will enclose 6 temperature sensors, which will offer useful information to validate the satellite thermal model. Voltage measurements will be accomplished by the 1-wire DS2450, analogical to digital converter, which contains 4 channels of 16 bits.

Because the 1-wire currents sensors are not available, the UN will employ the analogical current sensor MAX472 along with the 1-wire DS2450, analogical to digital converter. The output voltage of the MAX472 is directly proportional to the measured current and its output current equation for a typical application is given by:

$$V_{out} = \frac{R_{sense} \cdot R_{out} \times I_{out}}{RG}$$

Where RSENSE needs to be selected in order to obtain a minimal voltage drop associated to the maximum current to be sensed.

3.2.5 Communications Subsystem

The communications system chosen for the NS vehicle will be based on the TEKK KS-900/960 data radio and the Bay Pac 9600 modem. Both of them modified for space flight. The equipments are attractive because they have successfully flown in the QuakeSat Mission, in 2004, [6]. In addition, the radio has the following characteristics: 2 Watts RF output, varactor controlled Direct FM modulation, -30 to +60 ˚C operation, 0.199 Kg weight and 7.5-12 V operation.

Nevertheless, the NS laboratory model has been initially communicated with the ground station software through wire lines to allow a quickly validation procedure. Later we will migrate to a low power wireless system to validate the NS software.

4 Current validation tests performed in laboratory

Some of the NS subsystems have undergone certain preliminary testing. Among them: the flight computer, FC operational software, Earth station software, and communications subsystem based on low power wireless electronics. However, few other subsystems are just about to be implemented either in breadboards or directly in PCBs (for those parts only available in surface mount package) for first round testing. Like in the cases of: power subsystem, sensors, solar panels, and payloads.

On the other hand, we have plans to perform few integration tests among available subsystems. Particularly, a campaign of preliminary tests between the flight computer, its operations software, and the Earth Station software has been planned. Those results will be soon published.

5 Concluding Remarks

This article has presented the efforts made towards the design of a NS laboratory model. The nanosatellite project has been fed with knowledge and equipments (hardware and software) developed in past domestic experiences in the space field. In this way, the project has been fully designed with extensive use of COTs parts tested in previous space missions and reported in international publications. However, the project also has important advances in the construction of NS subsystems which demand large development time, such as: the flight computer, flight software, Earth station operations software, etc. In the case of complementary subsystems they are just about to initiate preliminary operative tests. Therefore, in the short term we expect to have a basic laboratory model. The goal will be to use the NS model to make presentations in order to promote the project and search for financial support to construct the flight model. The financial support also would be used to prepare a pair of payloads, to make the NS qualification tests, as well as to finance its launching. By the way, the last will be economic, because the launching cost is related to the satellite mass, which in this case will be of approximately 3.5 kgs.

References

[1] Unamsat, http://space.skyrocket.de/doc_sdat/unamsat-a.htm, (1996).
[2] Satex, http://pumas.iingen.unam.mx/proyec/satex/, (2003).
[3] Vicente-Vivas E. et al., "La Convergencia de un proyecto multi-institucional para el desarrollo de un Microsatélite con tecnología Mexicana", Revista INGENIERÍA Investigación y Tecnología, ISSN 1405-7743, pp. 157-168, Vol III, No. 4, Octubre-Diciembre (2002).

[4] Vicente-VivasE. and García-Nocetti DF., "Reconfigurable Semi-virtual Computer Architecture for a Long Available Small Space Vehicle", 2001 IEEE Aerospace Conference Proceedings, Big Sky, Montana, USA, March (2001).
[5] Surrey, http://centaur.sstl.co.uk/datasheets/SNAP.pdf, (2005).
[6] Long M., et al., "A CubeSat Derived Design for a Unique Academic Research Mission in Earthquake Signature Detection", Proceedings of the 16th Annual/USU Conference on Small Satellites, Logan, Utah, August (2002).
[7] Lee R., et al., "Anomaly Detection Using The Emerald Nanosatellite On-Board Expert System", Proceedings of the 2004 IEEE Aerospace Conference, Big Sky MT (2004).
[8] Ohta A., et al., "The University of Hawaii CubeSat: An Undergraduate Student Satellite Project", Fifth University Space Systems Symposium, Honolulu, Hawaii, November (2002).
[9] Miyamoto K., et al., "Tokyo Institute of Technology Small Satellite Projects", 7th University Space Systems Symposium, Waikoloa, Hawaii, November (2004).
[10] Rankin D., et al., "The CANX-2 Nanosatellite: Expanding the Science Abilities of Nanosatellites", 55th International Astronautical Congress, Vancouver, Canada, October (2004).
[11] TelecomWeb, http://www.telecomweb.com/cgi/pub/via/via02010508.html, February (2005).
[12] Cawley S., "Topsat: Low Cost High Resolution Imagery from Space", (2003).
[13] Surrey, http://www.surrey.ac.uk/news/releases/03-0929launch.html, (2003).
[14] Space-Tech, http://www.space-technology.com/projects/mosaic/, (2005).
[15] Liddle D., et al., "A low-cost geostationary minisatellite platform", Acta Astronáutica Journal, Elsevier Science, Vol. 55, pp. 271-284, (2004).
[16] Sun W., et al., "Micro-Mini-Satellites for Affordable EO Constellations: RapidEye & DMC", 3rd IAA Symposium on Small Satellites for Earth Observation, Germany, April (2001).
[17] Masafumi Iai, "A PDA-Controlled Pico-Satellite, Cute-1.7, and its Radiation Protection" Conference Proceedings 18th AIAA/USU Conference on Small Satellites, Logan, Utah, Aug. (2004).
[18] Pisacane V. and Moore R., "Fundamentals of Space Systems", The Johns Hopkins University Applied Physics Laboratory, Oxford University Press (1994).
[19] Williams, C., USA Air Force, "Verbal communication", Weber State University, Ogden, Utah, November (1994). Winokur P.S., et al., "Radiation-Hardened Microelectronics for Space Applications", Radiation Phys. Chem. Vol 43, No. ½., Great Britain, (1994).
[20] Vicente-Vivas E. et al., "Software de Adquisición de Telemetría y Control de Operaciones para Microsatélites", 4a Conferencia Internacional en Control, Instrumentacion virtual y Sistemas Digitales "CICINDI 2002", Pachuca, Hidalgo, Agosto (2002b).
[21] Maxim, http://www.maxim-ic.com/1-Wire.cfm, (2005).
[22] Vicente-Vivas E., et al., "The design of a Nanosatellite Power Subsystem ", CICINDI'05 International Conference on Digital Systems Design, Mexico City, September (2005).

Preliminary Design of the Power Subsystem for a University Nanosatellite

Esaú Vicente-Vivas[1], Jorge Morales[2] Jorge Prado-Molina[3], Gianfranco Bisiacchi[4] and Salvador Landeros[5],

[1]Instituto de Ingeniería, UNAM, [2] DEPFI-UNAM, [3]Instituto de Geografía, [4]Centro Tecnológico Aragón, FES Aragón, UNAM, Edo de México, [5] UNAM, Facultad de Ingeniería, UNAM,

Universidad Nacional Autónoma de México Cd. Universitaria Coyoacan, 04510, México DF.,

email: evv@servidor.unam.mx, jmorales@iingen.unam.mx, bdruan@yahoo.com, jprado@igiris.igeograf.unam.mx, diego@fi-b.unam.mx

Abstract. Nowadays, NanoSats are a cheap and easy way to develop space technology. They represent the best cost-benefit relation in terms of project cost, mass, power budget, communications performance, and launching cost. Therefore, this kind of projects can be developed in countries with emergent economies for technology demonstration purposes, remote sensing, technology improvement, human resource development, etc. Considering this scenario, Mexico is developing a 3.5 Kg University Nanosat (UN) which attempts to be the first of a series of NanoSats designed and developed by academic staff, postgraduate students and undergraduate students. This work presents the UN power subsystem design proposal, which employs some automotive commercial-of-the-shelf (COTS) parts successfully proven in previous international space missions.

Keywords: Nanosatellite, power generation, power storage, power regulation and power distribution.

1 Introduction

To date, the small satellites field has worldwide intense activities, with a huge participation of universities from industrialized countries, [1] and [2]. Few of them started microsatellite research in the 80s with the support of their local space agencies and space industries that were eager both to generate qualified human resources in the area and to qualify new space products.

In this way, the initial participation of very few developing countries was started with government initiatives that anticipated a future form of industrial revolution. These coun-

tries had the vision to invest, follow, and encourage its space activities expecting that some day would obtain a feedback action to grow and reinforce their industry. Among them we found examples like: China, India, and Brazil, [3]; countries that nowadays have become regional space powers. Between them, China is a special case that now is competing with worldwide space leaders. In recent years, nations such as Mexico, Korea, Taiwan, Chile, Portugal, Pakistan, Argentina, South Africa, Thailand, Singapore, Malaysia, Algeria, Nigeria, and Turkey have developed Microsatellite projects in partnership with experienced and worldwide recognized institutions like AMSAT, [4], and Surrey Satellite Technology Limited, [5]. The motivation of these countries is to stretch the technological gap in this important field which comprises a gate to develop other information technologies areas.

Until now, Mexico has accomplished two microsatellital missions. The first one, Unamsat A (17 Kg), was launched on March 28th, 1995, and his twin, Unamsat B, was dispatched on September 1996. There is a third effort, Satex (55 Kg), whose lack of financial support and organizational problems have avoided until now the completion of the mission. Nevertheless, a sound success is still required to obtain a substantial support in order to establish an enhanced and continuous satellite program.

Under this scenario, the UN project will assimilate the experiences and the know-how generated in the Satex project to design and manufacture a 3.5 Kg compact satellite bus capable to be adapted to different missions. In addition, the project will employ state of the art automotive electronics protected for use in space. The idea is to build a cheaper and better Nanosat to open the access of satellite technology to other local universities and institutions. At the same time the UN will offer enough on-board capabilities to admit different types of payloads. Additional project goals are to generate an attractive Nanosatellite platform in terms of development time (1 year in average), manufacture cost (60 thousand dollars excluding the payload cost) as well as in launching costs accordingly with the vehicle mass.

2 The University Nanosatellite

The prism shaped Nanosat has four solar panels attached to its body. In addition, four double sided solar wings were added to increase its power generation, figure 1. Six solar wings will be 18x9 cm while the other six will be 18x12.9 cm. The UN subsystems will reside in printed circuits which will be mounted in aluminium frames as well, figure 2. Therefore, the Nanosatellite platform will be composed of: power subsystem, telemetry sensors, structural subsystem, communications subsystem, flight computer, as well as room and resources (energy, communications and automation capabilities) to accommodate different payloads. Thus, the UN platform will contain all the necessary support subsystems for the satellite and its payloads.

The subsystems inherited from the Satex mission are: a flight computer, a structural subsystem reconverted to be used as Nanosatellite structure, sensors (magnetic field, temperature, current and voltage), as well as software for both satellite operations and Earth Station, [6]. In this way, the electronics modules and the frames will be assembled in tandem to form the satellite body, figure 1b. Once assembled its dimensions will be 18x9x12.9 cm.

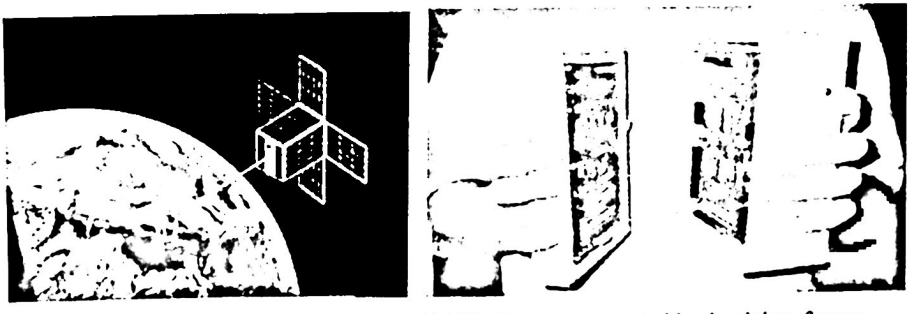


Fig. 1 a) The University Nanosatellite b) NS subsystems mounted in aluminium frames

2.1 The UN Power Subsystem

The Nanosat power subsystem will contain four different modules: power generation, power storage, power regulation, and power distribution. These subsystems will allow the Nanosatellite to generate the electrical energy required to operate its electrical equipments. Following sections describe the proposed preliminary design of UN power system. This is design will evolve in the short term into a proof of concept model and a later phase will be advocated to the construction of a flight model. This approach has been practiced before in the development of military small satellite power systems, [13] and [14]. Thinking about a the continuity of space activities in México, it will be very important continue the work not only in the small satellite field, but also in high efficient state of the art power systems, [15], because they will allow the placement of more sophisticated payloads in terms of communications, remote sensing, and so on. This particular research is being very successful in the microsatellite field, [16].

3 Power Generation Module

The UN power subsystem will generate its power from 12 solar panels connected in parallel fashion, figure 2. Four of them are attached to the satellite body and four of them

will be double sided deployable wings. The last configuration will allow the satellite to generate enough electrical power without high requirements of stabilization. This solar panel distribution has flown successfully by the University of Stanford in the QuakeSat Mission, [7]. Each solar panel will contain inexpensive GaAs solar cells with cover glass specially developed for the space industry. The cells are 20x25 mm, 0.204 mm thick from Boreal Laboratories Ltd, obtained as leftovers from companies that build solar panels for commercial satellites. Its electrical output is 100 mA/cm^2 at 0.5V. Then, the chosen 5 cm^2 solar cell presents an output current of 500 mA and an output voltage of 0.5 V. The cell output voltage is actually very sensitive to the temperature at which the cell is operating at. In general, the solar cells and series strings are reported to lose about 0.24% in voltage for each 1 °C increase in operating temperature. This aspect is expected to be studied in more detail in order to complete the UN power system design. Besides, validation tests in laboratory will support the final design. Each panel has a label indication to know its location in the satellite. PC label indicates a body panel and PD designates a deploying panel. For deploying panels, the letter T indicates a rear panel, while the letter F specifies a front panel. In addition letters N, E, S, and O refer to north, east, south, and west panels, respectively. In this way designators PDTN, PDTE, PDTS and PDTO are related with rear deploying panels north, east, south and west, respectively. Whereas PDFN, PDFE, PDFS and PDFO specify front deploying panels north, east, south and west, respectively.

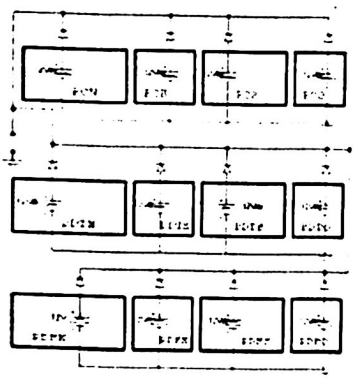
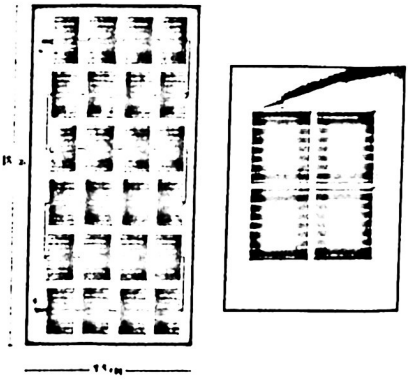


Fig. 2 a) Solar panels distribution for the University Nanosatellite.



b) University Satellite solar array schematic composed by 24 cells arranged in series and photograph of solar cells to be employed.

Figure 2b shows how a panel will look like with solar cells mounted on its surface. In addition, figure 2 presents a picture of the solar cell to be employed. Each panel contains 24, 20x25 mm, solar cells arranged in one single string. Therefore, the non regulated

cell, to the positive terminal of the next cell in the string, with both cells being in the same solar cell row, d) Turn-around: connects the negative terminals of one solar cell, to the positive terminal of the next cell in the string, with one cell being in the next row of solar cell. The interconnections have stress-relief areas to accommodate the expansion and contraction of the cells due to temperature changes. They will be first soldered to the solar cells. A number of this subassembly will be then assembled to make a string, conforming to the cell layout of their respective panels. The assembling processes will employ a number of jigs that will be custom machined for laying out our specific type of solar cell.

The panel structures might be made of space rated Aluminium-6061. Some microsatellite panel structures are made of 0.25 inch thick solid Aluminium, others of 0.5 inch thick Aluminium honeycomb, while others are made of 2 parts: the solar panel containing the cells which is made out of 0.063 inch Aluminium sheet, and a structural support panel which is made out of 0.5 inch Aluminium honeycomb. The more convenient choice for the UN is being studied by now.

The outer UN Aluminium surfaces that will hold the solar cells will be overlaid with dielectric material, for electrical insulating purpose. Domestic facilities to accomplish this process are also investigated.

Once the solar panels become ready, the cells will be bonded to the FM73 (epoxy) covered solar panel surface using a controlled volatility RTV (silicon compound) that we expect to obtain from international suppliers. The RTV would adhere to the area between the two negative terminal solder spots on the back of the cell. Once determined the layout of the cells on a panel, the areas will be located on two 10-mils Mylar sheet. These areas will then cut out, forming masks. One of the mask is used to held the prearranged cells, while the other one is used for masking the areas on the FM73 covered solar panel that were not to be glued.

The Mylar used for this masking process should have been 20-mils thick. This will ensure that enough RTV will be present for optimum bonding between the cell and the F-77 (adhesive) covered solar panel surface. In case we do not have access to 20-mils Mylar, we might use 10-mils Mylar, which has also been used in microsatellites that underwent successfully shake tests.

4 Power Storage

The UN generated power is stored in four Li+ batteries, connected in an array of two parallel strings, each one containing two batteries in series. The batteries are Tadiran AA-sized lithium (model TLM-155HP), with a capacity of 5A of continuous current at 4.0V, a discharge capacity of 550 mAh, and an operating temperature range of -40°C to 85°C; figure 2b. Their typical applications include military and aerospace systems, due to its high power, long life and, extended storage, [8].

cell, to the positive terminal of the next cell in the string, with both cells being in the same solar cell row, d) Turn-around: connects the negative terminals of one solar cell, to the positive terminal of the next cell in the string, with one cell being in the next row of solar cell. The interconnections have stress-relief areas to accommodate the expansion and contraction of the cells due to temperature changes. They will be first soldered to the solar cells. A number of this subassembly will be then assembled to make a string, conforming to the cell layout of their respective panels. The assembling processes will employ a number of jigs that will be custom machined for laying out our specific type of solar cell.

The panel structures might be made of space rated Aluminium-6061. Some microsatellite panel structures are made of 0.25 inch thick solid Aluminium, others of 0.5 inch thick Aluminium honeycomb, while others are made of 2 parts: the solar panel containing the cells which is made out of 0.063 inch Aluminium sheet, and a structural support panel which is made out of 0.5 inch Aluminium honeycomb. The more convenient choice for the UN is being studied by now.

The outer UN Aluminium surfaces that will hold the solar cells will be overlaid with dielectric material, for electrical insulating purpose. Domestic facilities to accomplish this process are also investigated.

Once the solar panels become ready, the cells will be bonded to the FM73 (epoxy) covered solar panel surface using a controlled volatility RTV (silicon compound) that we expect to obtain from international suppliers. The RTV would adhere to the area between the two negative terminal solder spots on the back of the cell. Once determined the layout of the cells on a panel, the areas will be located on two 10-mils Mylar sheet. These areas will then cut out, forming masks. One of the mask is used to held the prearranged cells, while the other one is used for masking the areas on the FM73 covered solar panel that were not to be glued.

The Mylar used for this masking process should have been 20-mils thick. This will ensure that enough RTV will be present for optimum bonding between the cell and the F-77 (adhesive) covered solar panel surface. In case we do not have access to 20-mils Mylar, we might use 10-mils Mylar, which has also been used in microsatellites that underwent successfully shake tests.

4 Power Storage

The UN generated power is stored in four Li+ batteries, connected in an array of two parallel strings, each one containing two batteries in series. The batteries are Tadiran AA-sized lithium (model TLM-155HP), with a capacity of 5A of continuous current at 4.0V, a discharge capacity of 550 mAh, and an operating temperature range of -40°C to 85°C; figure 2b. Their typical applications include military and aerospace systems, due to its high power, long life and, extended storage, [8].

The battery voltage will vary depending on the charge condition of the battery. It will range from 3.36 Volts at 80% discharged condition to 4.2 Volts, per battery, at fully charged condition. The 80% discharged condition is considered to be the minimum power level. Going beyond this point will put the nanosat at risk of not being able to recharge itself. The solar panels will be producing an average of 12 Watts of power. Since the capacity of the battery is 550 mAH, there will be no danger of over-charging the batteries. Therefore, charge limiters are not used, which also helps in simplifying the circuits. However, this scheme is still under analysis to verify its operation. In addition, laboratory tests are being planed for design validation purposes.

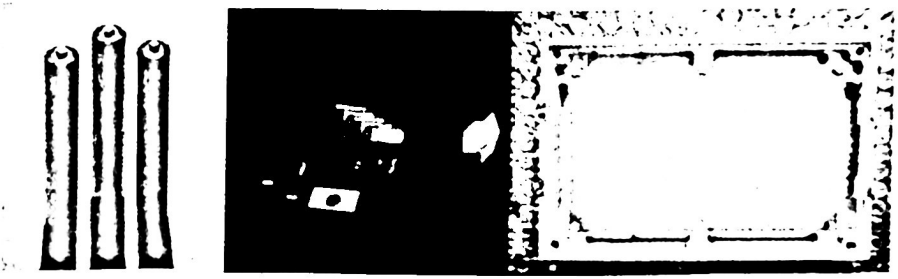


Fig. 3 NS subsystems will be mounted in aluminium frames batteries

It was decided to use one battery system, with no backup. An extra battery would increase weight considerably. Also, given the high reliability and performance of the type of battery selected, one battery pack should be more than sufficient to provide continuous power over the entire orbit and over the satellite's mission life.

4.1 Battery Pack

The batteries will be confined in a battery pack housed in an individual box, figure 5. They will be potted by RTV566 compound, made by General Electric, to provide adequate thermal mass for ensuring the proper operating temperatures. The box will have few temperature sensors to monitor the battery's temperature.

4.2 Battery Charging

The battery charging control process will be carried out by a MAX1873, which allows 2 batteries to be charged at the same time, [9] and [10]. This means that there are two Max1873 to control the charge of the four batteries. The COTS component was chosen for the UN because it was successfully validated in space orbit by the Quakesat mission

[7]. The Max 1873 datasheets contain a typical application circuit. In our case, this circuit was modified to fit the Nanosat power source characteristics (solar panels arrays); being 12 V and 0.5 A the best operative case. The most important modifications were the value of the inductor L1, the battery current sense resistor Rcsb, and the P-channel MOSFET switch selection, figure 6. For Rcsb selection [9] it is considered that Icgh takes its maximum value, which in our case is 0.5 A. The next equation was employed for Rcsb calculation:

$$Rcsb = 2.0\ V / Icgh = 4.0\ \Omega$$

The inductor value must be selected to obtain a reasonable ripple current. The greater the inductance, the lower the ripple current [9]. Typically, an inductor is chosen with a ripple current between 30% to 50% of DC average charging current. The next equation was used for inductor calculation:

$$L = \left[V_{BATT}\left(V_{DCIN} - V_{BATT}\right)\right] / \left(V_{DCIN(MAX)} \times f_{sw} \times I_{CHG} \times LIR\right)$$

Where:
LIR = ratio of ripple current, in our case 50 %
$V_{DCIN(MAX)}$ = 12 V
V_{BATT} = 8.4 V
Fsw = switching frequency, nominally 300 kHz

In this case we obtained L = 38.04μH, however, a commercial inductance value of 35 μ H fulfils the requirements. At last, the MOSFET switch must be selected to meet the efficiency or power dissipation requirements of the charging circuit. The charger was selected to meet the power dissipation. In general for MOSFETS, the worst case power dissipation due to on-resistance (P_R) occurs at the maximum duty cycle, where the operating conditions are: minimum source voltage and maximum battery voltage [9]. This can be approximated by the following equation:

$$P_R = \left(V_{BATT(MAX)} / V_{DCIN(MIN)}\right) \times R_{DS(ON)} \times I_{CHG}^{2}$$

For the chosen MOSFET, IRFR9014, R_{DON} = 500 mΩ, then P_R = 116.66 mW. This power dissipation is minimal for the IRFR9014 considering its maximum power capability of 25 W.

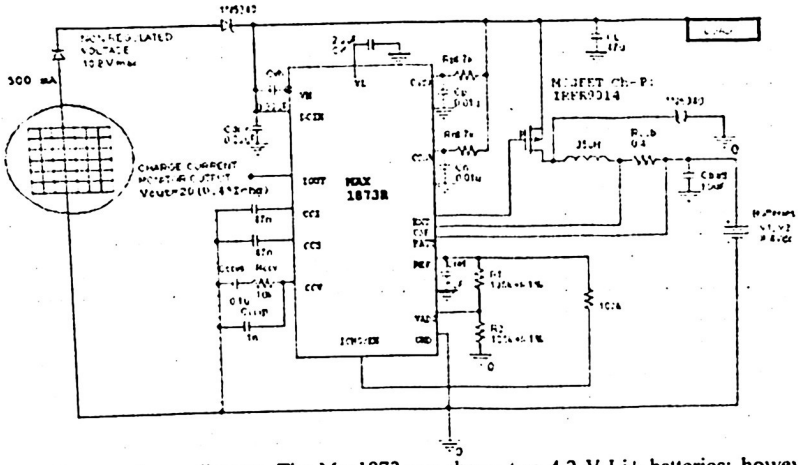


Fig. 6 Battery charge diagram. The Max1873 can charge two 4.2 V Li+ batteries; however, the maximum charge current is limited to 500 mA.

4.3 Unregulated Line

Unregulated DC power will be supplied to the Communication Subsystem. This involves direct connection from the charge regulator input terminals to the power input terminals at the Communication Subsystem. The unregulated voltage level will vary depending on the charge condition of the battery.

5 Power Regulation

The required regulated voltages for the UN model will be +5V and -5V. The former will be used by the flight computer, telemetry sensors, and probably for payload energizing. The last is planed to be used with power stabilization sensors (magnetometer and fine sun sensors), payloads as well as a polarization voltage for the Latch-up sensor placed in the flight computer. Regarding the communications system, it will be energized with the unregulated voltage obtained directly from the batteries.

In this way, the COT DFC6U5S5 device was chosen for power regulation in the UN. The 77 % efficiency DFC6U5S5 can provide a regulated voltage with up to 6 watts of output power, which means more than 1 Amp of output available current, [11]. This current is enough for the UN equipment requirements. In addition, the device accepts a wide

3.5 to 16 Volt input range allowing the operation from the UN batteries. Moreover, its output is electrically isolated, thereby allowing the output to be configured as a negative or a positive output voltage. Besides, its isolated and filtered output allows usage in low noise circuits. Therefore this device will allow to obtain the +5V and -5V power supplies required for the Nanosatellite model. Besides, few of them may be used in case the satellite loads demand more current than that provided by the device.

6 Power Distribution

Once the required regulated voltages are available in the UN platform, they have to be controlled directly from the flight computer according with satellite mission needs specified from Earth. In other words, the flight computer should be able to switch on or switch off any satellite equipment to carry out specific satellite tasks. To accomplish this power distribution scheme, the UN laboratory model will employ MOSFET IRFF130 from International Rectifier, which has a drive capacity of 8 Amp, 100V, 0.18 Ω, [12]. This device will drive any one of the UN loads without problem. However, several of them will be necessary to control every one of the controllable satellite modules (communications transmitter, sensors, actuators and payloads). On the other hand, other automotive COTS devices are being considered for the UN engineering model. We would like to include a surface mount device, with good driving capabilities and with less voltage restrictions to be commuted.

7 Concluding Remarks

The paper has presented the preliminary design of the power subsystem for the University Nanosat laboratory model which is expected to weight about 3.5 Kg. In order to shrink the manufacturing cost of the UN, the exposed design makes use of commercial-off-the shelf components that have been successfully qualified in previous small satellite missions. Moreover, the projected design employs good quality cheap solar cells, which will allow the generation module to be composed of 12 solar panels with two different sizes. However, each panel will hold the same amount of solar cells and therefore the same electrical characteristics. In this way the module will provide in average an estimated power of 12 Watt, at 12 V and 500 mA. This will be enough to energize the flight computer, telemetry sensors and communications equipment and a couple of payloads. Regarding the power storage module, it will contain 4 Li+ military spec batteries and his voltage charger will consist of COTS automotive electronics. The power regulation as well as the power distribution modules will also be implemented with COTS electronics.

It is expected in the short term to perform laboratory tests in order to validate the proposed design for the UN power system. In this way a modular and cheap Nanosatellite

laboratory model might evolve into an engineering model. Even more, in the medium term the UN laboratory model might evolve into a flight model if financial support is obtained. In this sense a publicity campaign is expected to be deployed with the help of the Nanosatellite proof of concept that will contain the power system presented in this paper.

References

[1] Klimov S.I. et al., "Aerospace Education Program Realization by Means of the Microsatellite ",Acta Astronautica Journal, Vol.56,iss.1-2[Special Issue], pp. 301-306, Elsevier Science, 2005AcAau..56..301K, January (2005).

[2] Kitts, C., "Three Project-Based Approaches to Spacecraft Design Education," In Proceedings of the 1999 IEEE Aerospace Conference, Snowmass, CO, March (1999).

[3] Power and Interest News Report, "The Coming World Realignment", http://www.pinr.com/report.php, June 20 (2005).

[4] A Brief History of Amateur Satellites, http://www.amsat.org, (2004).

[5] Surrey Satellite Technology Limited, (2005) , http://www.spaceref.com

[6] Proyecto Satex en la UNAM, http://pumas.iingen.unam.mx/proyec/satex/ , (2003).

[7] Long, M., et al., "A Cubesat Derived Design for a Unique Academic Research Mission in Earthquake Signature Detection". Ref. SSC02-IX-6. 16th Annual/USU Conference on Small Satellites, August (2002).

[8] Tadiran Batteries, http://www.tadiran.com/hipower.php, (2005).

[9] Maxim, MAX1873 simple current limited switch-mode Li+ charger controller. Maxim Datasheet, http://www.maxim-ic.com (2005).

[10] Dallas Semiconductor, Evaluation Kit for the MAX1873 Kit, http://www.maxim-ic.com, (2005).

[11] DFC6U5S5 power regulator specs.

[12] http://www.digchip.com/datasheets/parts/datasheet/385/DFC6U5S5.php

[13] IRFF130 Mosfet Data Sheets, http://www.alldatasheet.co.kr

[14] Malone P. et al., "Developing an Inflatable Solar Array", "In NASA/DOD Flight Experiments Technical Interchange Meeting Proceedings, 15p (SEE N93-28699 11-12), Washington, (1992).

[15] Lichodziejewski et al., "Inflatable Rigidizable Solar Array for Small Satellites", 44th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Norfolk, Virginia, April, (2003).

[16] Spacedaily, Tech Space, Invention Promises To Change The Way Small Satellites Are Powered, El Segundo CA, Nov. (2001) http://www.spacedaily.com

[17] Redel F. and Lichodziejewski D. "*Power-Scalable Inflation-Deployed Solar Arrays*", 45th AIAA/ASME/ASCE/ AHS/ASC Structures, Structural Dynamics & Materials Conference, Palm Springs California, April, (2004).

Design and implementation of a Robotic Arm Able to Play "TIC-TAC-TOE" controlled by a FPGA

Norma Elva Chávez Rodríguez[1], Fernando Arámbula Cosío[2], Joel Esquivel Villar[1], Miguel Ángel Valdes[1]

[1] Departamento de Ingeniería en computación, Facultad de Ingeniería UNAM, MÉXICO
[2] Laboratorio de Análisis de Imágenes y Visualización, Centro de Ciencias Aplicadas y Desarrollo Tecnológico UNAM, MÉXICO

e-mail :norma@fi-b.unam.mx

Abstract. This paper show a field programmable gate array (FPGA) based control of a robotic arm able to play TIC-TAC-TOE. The architecture of the control scheme is simple, and thus facilitates realization of the proposed digital controller. The designed controller has been implemented using the SPARTAN-3 STARTER KIT from Xilinx, Inc. Our "IC" can be used as a microprocessor in applications of robotic arm control. Due to the high-speed nature of FPGAs, the sampling frequency of the IC can be raised to values that cannot be reached using a conventional digital controller based on a microcontroller. The strength of this paper is the implementation of a 9-choice algorithm that makes the robotic arm win the game. "VHDL" language and Synthesis tool (view logic) was used to provide the FPGA with the information of the moves and the algorithm to select the most appropriate move.

Keywords: VHDL, digital design, Robotic Arm, FPGA

1 Introduction

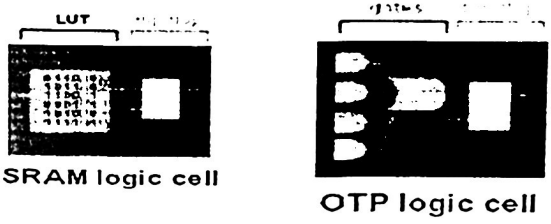
Robots have become important due a wide range of applications from manufacturing and surgery to the handling of hazardous materials. One of our robotic arm's functions is to move to a specific location or along a determined path. Moves can be performed in nine different ways thanks to the algorithm specially design to play wisely (intelligently?). A FPGA is used as control of the robotic arm with five motors and five joints to allow flexibility, five Axes of motion: Right / Left 350 degrees; Shoulder 120 degrees; Elbow 135 degrees; Wrist rotate CW & CCW 340 degrees; Gripper Open & Close 50 mm. Its dimensions are: Max Length Outwards = 360 mm Max Height Upwards = 510 mm Max Lifting Capability = 130g.

Field-Programmable Gate Arrays FPGAs are a special type of Application Specific Integrated Circuits ASICs which can be configured or reconfigured by the user instead of the manufacturer. One of the most important characteristics of FPGAs is fast circuit prototyping. A digital device of up to thousands of logic gates can be implemented and/or revised in days or even hours. In addition, FPGAs have a low cost of manufacturing, and have the property of being fully testable. FPGAs are gate-array-like devices, and they are typically used to implement multi-level logic functions. This paper shows that a FPGA can control directly a chopper driven brush DC motor and in addition generate a sequence of robotic arm commands.

The paper is organized as follows: Section 2 presents an overview of how and where programmable logic devices are used, including both "CPLD" & "FPGA" devices, and discusses the synthesis and implementation process for FPGAs. The design targets a Spartan-3 "FPGA". Section 3 takes the "VHDL" language through to a working physical device, Section 4 shows the sensor used in the board. Section 5 presents the implementation of our control "IC", finally, section 6 is dedicated to conclusions.

2 Overview of PLDs

By the late 70's, standard logic device were the rage and printed circuit, boards were loaded with them. Then someone asked the question: "What if we gave the designer the ability to implement different interconnections in a bigger device?" This would allow the designer to integrate many standard logic devices into one part. The two programmable planes provided any combination of AND and OR gates and sharing of AND terms across multiple OR's. This architecture was very flexible, but at the time due to wafer geometry's of 10um the input to output delay or propagation delay (Tpd) was high which made the devices relatively slow. Complex Programmable Logic Devices CPLD is the way to extend the density of the simple PLDs. The concept is to have a few PLD blocks or macrocells on a single device with general purpose interconnect in between. Simple logic paths can be implemented within a single block. More sophisticated logic will require multiple blocks and the use of the general purpose interconnect in between to make these connections. Field Programmable Gate Array FPGA is a regular structure of logic cells or modules and interconnect, which can be controlled completely. This means that changes to the circuit, its design and program can be done freely as required. There are two basic types of FPGAs: SRAM-based reprogrammable and OTP. These two types of FPGAs differ in the implementation of the logic cell and the mechanism used to make connections in the device. The dominant type of FPGA is SRAM-based and can be reprogrammed as often as you choose. In fact, an SRAM-FPGA is reprogrammed every time it's powered up, because the FPGA is really a fancy memory chip. That's why you need a serial PROM or system memory with every SRAM-FPGA.



a) b)

Fig. 1 a) SRAM based FPGA Logic Cell; b) OTP based FPGA Logic Cell.

In the SRAM logic cell, instead of conventional gates, an LUT determines the output based on the values of the inputs. (In the SRAM logic cell, Figure 1a, six different combinations of the four inputs determine the values of the output.) SRAM bits are also used to make connections. OTP FPGAs use anti-fuses (contrary to fuses, connections are made, not "blown," during programming) to make permanent connections in the chip. Thus, OTP-FPGAs do not require SPROM, or other means to download the program to FPGA. However, every time you make a change on design, you must throw away the chip. The OTP logic cell is very similar to PLDs, with dedicated gates and "flip-flops". The availability of CAD-software such as WebPACK ISE from Xilinx Inc. and Max+Plus II from Altera Inc. has made it much easier designing with programmable logic. Designs can be described easily and quickly using a description language such as ABEL, VHDL, Verilog-HDL, and AHDL or with a schematic capture tools.

3 Introduction to VHDL

VHDL is a language for describing digital electronic systems. It arose out of the United States Government's Very High Speed Integrated Circuits (VHSIC) program, initiated in 1980. In the course of this program, it became clear that there was a need for a standard language for describing the structure and function of integrated circuits (ICs). Hence the VHSIC Hardware Description Language (VHDL) was developed, and subsequently adopted as a standard by the Institute of Electrical and Electronic Engineers (IEEE) in the US. VHDL is designed to fill a number of needs in the design process. Firstly, it allows description of the structure of a design and its decomposition into sub-designs, and how those sub-designs are interconnected. Secondly, it allows the specification of the function of designs using familiar programming language forms. Thirdly, as a result, it allows a design to be simulated before being manufactured, so that designers can quickly compare alternatives and test for correctness without the delay and expense of hardware prototyping.

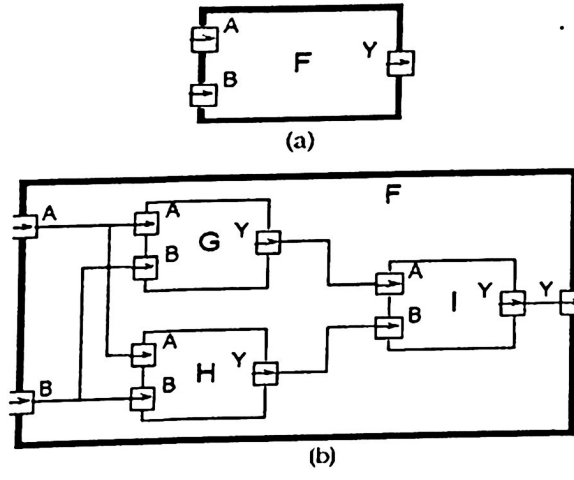


Fig.2 Example of a structural description.

3.1 Describing Structure

A digital electronic system can be described as a module with inputs and/or outputs. The electrical values on the outputs are some function of the values on the inputs. Figure2(a) shows an example of this view of a digital system. The module F has two inputs, A and B, and an output Y. Using VHDL terminology, we call the module F a design *entity*, and the inputs and outputs are called ports. One way of describing the function of a module is to describe how it is composed of sub-modules. Each of the sub-modules is an *instance* of some entity, and the ports of the instances are connected using *signals*. Figure 2(b) shows how the entity F might be composed of instances of entities G, H and I. This kind of description is called a structural description. Note that each of the entities G, H and I might also have a structural description.

3.2 Describing Behavior

In many cases, it is not appropriate to describe a module structurally. One such case is a module which is at the bottom of the hierarchy of some other structural description. For example, if you are designing a system using IC packages bought from an IC shop, you do not need to describe the internal structure of an IC. In such cases, a description of the

function performed by the module is required, without reference to its actual internal structure. Such a description is called a functional or behavioral description. To illustrate this, suppose that the function of the entity F in figure2-1(a) is the exclusive-or function. Then a behavioral description of F could be the Boolean function Y = A . B + A . B More complex behaviors cannot be described purely as a function of inputs. In systems with feedback, the outputs are also a function of time. VHDL solves this problem by allowing description of behavior in the form.

3.3 Discrete Event Time Model

Once the structure and behavior of a module have been specified, it is possible to simulate the module by executing its VHDL language description. This is done by simulating the passage of time in discrete steps. At some time, a module input may be stimulated by changing the value on an input port. The module reacts by running the code of its VHDL language description and scheduling new values to be placed on the signals connected to its output ports at some later time. This is called scheduling a transaction on that signal. If the new value is different from the previous value on the signal, an *event* occurs, and other modules with input ports connected to the signal may be activated. The simulation starts with an *initialization phase*, and then proceeds by repeating a two-stage *simulation cycle*. In the initialization phase, all signals are given initial values, the simulation time is set to zero, and each module's behaviors program is executed. This usually results in transactions being scheduled on output signals for some later time. In the first stage of a simulation cycle, the simulated time is advanced to the earliest time at which a transaction has been scheduled. All transactions scheduled for that time are executed, and this may cause events to occur on some signals. In the second stage, all modules which react to events occurring in the first stage have their behaviors program executed. These programs will usually schedule further transactions on their output signals. When all of the behaviors programs have finished executing, the simulation cycle is repeated. If there are no more scheduled transactions, the whole simulation is completed. The purpose of the simulation is to gather information about the changes in system state over time. This can be done by running the simulation under the control of a *simulation monitor*. The monitor allows signals and other state information to be viewed or stored in a trace file for later analysis. It may also allow interactive stepping of the simulation process, much like an interactive program debugger.

4 Control design and implementation

Due to limitation on the length of the paper the implementation of the VHDL code is not show, however we explain how the arm and interfaces work on the tic-tac-toe game.

4.1 The "Tic-Tac-Toe" Game

This game is played on a board with nine different holes, which are all of the same size and equipped with a pair of sensores, (The sensores are used to recognize if there is or there isn't a ball inside them, the sensores send a signal according to the presented condition to a "FPGA") 10 balls of the equivalent size are also used. The balls are divided in two group colors, 5 green balls, which are the balls that the robotic arm shall use, and 5 red balls, which are the balls that the Robotic's arm challenger shall employ.

The game is developed as a normal tic-tac-toe game, although the Robotic arm, will always have the first move. (This is so that the "FPGA" will be able to note a difference between the challenger's moves, and the Robotic's arm moves. What the "FPGA" does to recognize this difference, is that it separates and stores in a distinct place the moves that are an even number from the moves that are an odd number. The person will clearly always make the even moves, while the robotic arm will always make the odd ones.) This Robotic arm that plays "Tic-Tac-Toe", was programmed under an algorithm in "VHDL" language, which doesn't allow it to loose.

4.2 Power System

We use a FPGA as digital controller. The control of the robotic arm drives five DC motors, we used an L2930 driver, and each driver can control 2 motors. The figure 3 shows the control of two motors by driver.

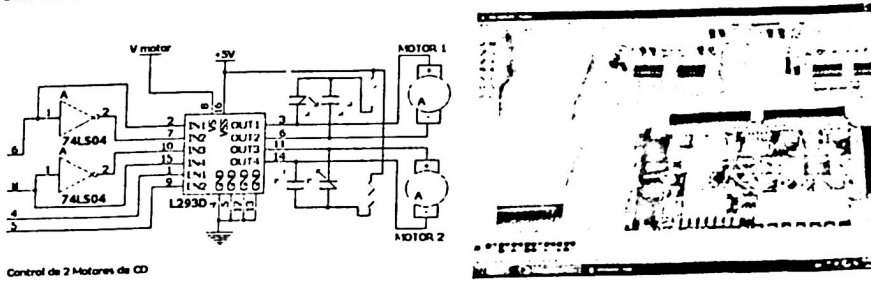


Fig. 3 The control of two DC motors

4.3 How the Controller Works

The robotic's arm control is in the FPGA. The FPGA has as an input the signals that come from the ultra red sensores. With these signals, the FPGA decides how the movement of the robotic's arm motors should be, according to the input, the FPGA sends signals to the motors that will make them move in the desired way. Between the FPGA and the motors, we have the interface of the drivers, which are used to boost the output power of the signals that the FPGA sends to the Robotic arm. See Figure 4.

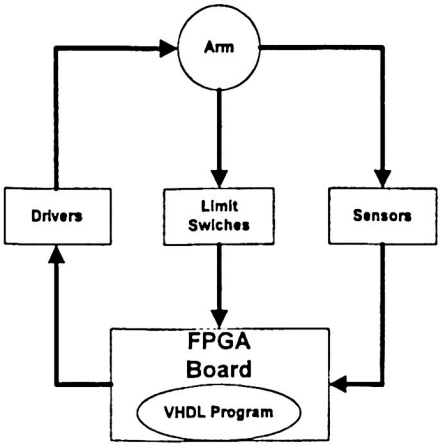


Fig. 4 Block Diagram of System Controller

5 RESULTS AND CONCLUSIONS

With all this, we created a robotic arm that is able to be an exciting tic tac toe challenger for any player who comes before it. The Robotic arm has good movements that allow it to present a high exactitude when placing the ball inside the hole. On the other hand, this Robotic arm never looses a game. It either ties or wins its matches. The longest time that a game with the Robotic arm can last is 15 minutes approximately. In the following Figures (5, 6), we can see the Robotic arm in action:

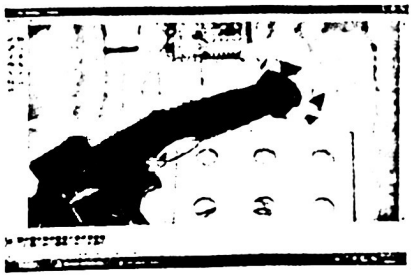


Fig. 5 The arm reaching for a ball to throw

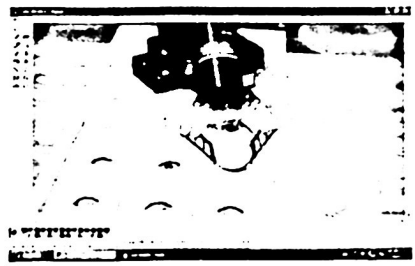


Fig. 6 The arm making the first move in the game.

References

[1] S. Brown, Z. Vranesic "Digital Logic with VHDL Design" McGraw Hill 2nd Edition.
[2] A. Lloris, A. Prieto, L. Parrilla "Sistemas digitales" Mc Graw Hill.
[3] J. Martin "Lenguajes formales y teoría de la computación" Mc Graw Hill , 3ª Edición
[4] D.Givone"Digital Principles and Design", Mc Graw Hill.. 2nd Edition
[5] A. Marcovitz "Introduction to Logic Design" ,Mc Graw Hill. 2nd Edition.
[6] "FPGA Starter Kit Manual" Book, 4ª Edition, Xilinxs .inc.
[7] "Getting Start" Manual, 5ª Edition Altera. inc.

Real Time Systems

REAL-TIME CONSTRAINTS FOR DIGITAL FILTERS TO PARAMETERS ESTIMATION ON MONOVARIABLE LINEAR STOCHASTIC SYSTEMS

J. J. Medel-Juárez[1] P. Guevara-López[2], D. Cruz-Pérez[1]

INSTITUTO POLITECNICO NACIONAL, MEXICO D. F., MEXICO.
CIITEC-IPN *[1,3], CICATA-IPN *[1,2,3], DEMS-IPN *[2]
{jjmedel1, pguevara2}@ipn.mx, dcruzp@yahoo.com3

Abstract The class of digital filters applied into PC and those interacts with dynamic processes and emit high quality output responses with time constraints and critical synchrony, will be described as RTDF (Real-time Digital Filters). The main RTDF properties will be described including a basic estimator example. The document is formed by the next sections: a) Digital Filter and Real-time Systems basically theories and main results of Real-time Digital Filtering., b) The RTDF theoretical results: considering local and global constraints., c) The RTDF implementation: considering a D. C. motor parameters estimator. In this part develops an extensive analysis of concurrent tasks: precedence constraints and times into a PC are considered.

Keywords: Digital Filter, Synchronization, Real-time, Task, Interval.

1 Introduction

Digital Filters are applied in industrial processes, control systems and monitoring systems [5], [7]. For example, chemical plants, manufacturing processes, airbags and fuel injection systems, voice analysis, data acquisition, medical applications, telecommunications, missile trajectories, etc. Digital Filters can't fail, in two senses: response quality and response time, or the processes can be crashed. For these reasons Digital Filters must be specified and implemented as *Real-time Digital Filters (RTDF)*. A RTDF can be implemented into embedded systems [4] using micro controllers, DSP's, etc. In addition, a RTDF can be implemented in digital computers with Real-time Operating Systems (RTOS).

2 Digital Filters

The filter concept, is commonly used to describe a physic device characteristic, using the information given by the *system* (to see: [13] and [9]); such that, the filter is applied to a set of data generated by a corrupted signal. To speak about the information, it means to consider a lot of states of the system combined with a lot of perturbations (these perturbations have two sources around to the system: a) External and b) internal. The noises or perturbations may arise from a variety of sources). The filter process is based on: a) Monitoring states (issued from the system), and b) Forecast or identify noises (originated from emitter, receptor or environment of system). The filter theory has two basic areas, which describe the principal properties about the essential system characteristics (inside and outside qualities of it) [13], [21]: *Identification* and *estimation.* In both cases, the filters use the information emitting by the system. Error functional *J(k),* (estimation or identification) must be converge to minimal value in finite time [13], [6]; the convergence criterion is defined by a difference between the filtering signal and original signal and applying on it, the second probability moment in this result [13], [21]. Examples of Digital Filters are: Wiener, Kalman [13] and Medel & Poznyak filter [22].

3 Real-time Systems (RTS)

Many authors define a Real-time Systems (RTS), (for example: Martin in [20], Burns & Wellings in [4], Heitmeyer & Mandrioli in [14], Gray in [11], De la Puente in [8], Stankovic in [26], in the others). And all these authors conclude in the same way: *Real-time System (RTS) is a system that suffice three conditions: a) interaction with physical world, b) correct responses, time constraints from physical world. RTS* must be synchronized with physical world and this velocity is relative, depending of dynamics of real system and the filter algorithm, and the computing system. Then, *RTS* may be fast or slow depending of real system dynamics; this obeys the criterions exposed in [23]. In a PC, the whole of all activities are processed by *Real-time tasks* [18].

4 State of Real-time Digital Filters

Chui & Chen exposing in [7] that Real-time Digital Filters (Kalman for example) can be implanted in *RTS* considering the high velocity computers and facility to express the filter in recursive form. Fredrik Gustafsson describing in [12] that FIR (*Finite Impulse Response*) Digital Filters as "applications to Real-time processing of standard signals". Papoulis & Bertran suggesting in [24] that the implementation of Digital Filter into PC, is through to consider the times measure of all tasks around the implementation of it into PC (considering the measurable: The A/D and D/A converters, processor operations, filter

algorithm and precedence constraints). But the concepts about Real-time Digital Filters never were cited: Jane Lui presenting an example of Kalman Filter in [18], but she didn't justify the Real-time characteristics of it, only said that it is a Real-time application. Baras in [3] described the Real-time Signal Processing only if it is expressed in recursive form.

5 Real-time Digital Filter (RTDF)

In base of these references, the characterization of *Real-time Digital Filters (RTDF)* is absolutely necessary. In this section the basic properties of these systems in real time sense are explained. If the Digital Filter selected has interaction with dynamical process, and has constraints in time, could be described as a RTDF, complied the characteristics of real time systems [But97].

Definition 1 (Real-Time Digital Filter). A RTDF is a Digital Filter with time constraints imposed by the dynamical process in the Nyquist sense [23], and:

1. *Receiving* and *giving* input and output responses, respectively, in synchronized form with respect to dynamics of the process. Inputs and outputs RTDF will be expressed in symbolic form as $\{u(k)\}_g$ and $\{y(k)\}_i$, with g, i, $k \in N^+$. [*],
2. *Giving correct responses set with respect to dynamical process:* The quality response is defined in local and global senses considered in [13], and must guarantee stable conditions defined in [2] and [6] and
3. *Express the RTDF in recursive form considering the concepts exposed by* Baras in [3] and Chui & Chen in [7], guarantee a minimal use of resources and memory, simulating the dynamics of the real process.

5.1 Implementation constraints of RTDF in a digital computer

When a RTDF is implemented in a computer with one processor, its different components use *concurrent Real-time tasks* (to see: [5] and [18]). The real-time task characteristics will be expressed in the following:

Definition 2 (RTDF: *local constraints*). The RTDF tasks have a lot of local constraints, imposed these by a lot of dynamical characteristics (in Nyquist sense [23]), with respect to real process. In Fig. 1, will be illustrate it.

---

* g, i, *represent a probably number of inputs and outputs respect in a concurrent system that evolution to k intervals.*

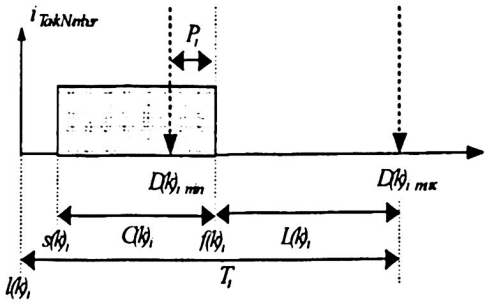


Fig. 1 RTDF as Real-time task

a) Arrival time. ($l(k)_i$) [5]: Is the time when task becomes ready for execution. $l(k)_i > 0$ $\forall k \in N^+$., *b) Computation time ($C(k)_i$)*: Is the total time in a *instance* (to see: [5]) of a task obtained by union of times used for atomic *activities*., *c) Deadline minimum relative ($D(k)_{i\ min}$)*: Is the minimum time in which a task should be finished[2]. It is a function of $l(k)_i$., d) *Deadline Maximum relative ($D(k)_{i\ Max}$)*: Is the maximum time which a task should be accomplished[3], bounded it by the right side with sample time. This deadline is a function too of time $l(k)_i$ and depend of system., e) *Start time ($s(k)_i$)*: Is a time when the task start with execution. This time depend of: Resource availability, latency times, context change, size of ready queue, etc. Start time is considered a stochastic variable [10] with a range named *jitter*,[18], and fulfill the condition $s(k)_i < D(k)_{i\ max}$., f) *Finishing Time ($f(k)_i$)*: Is the time when the task execution finished under interval time *k*. In mathematical sense $f(k)_i \in [ld(k)_{i\ min}\ LD(k)_{i\ max})$, with $ld(k)_{i\ min} := l(k)_i + D(k)_{i\ min}$ and $LD(k)_{i\ max} := l(k)_i + D(k)_{i\ max}$[4]., *g) Lateness ($L(k)_i$)*: Is the loss time between finishing time and final interval time *k*. This time is $L(k)_i := |LD(k)_{i\ max} - f(k)_i|$., h) *Premature time ($P(k)_i$)*: Is the gain in time generated by finishes task before maximum deadline. This time is defined by $P(k)_i := |ld(k)_{i\ min} - f(k)_i|$, where $f(k)_i \geq ld(k)_{i\ min}$ generate a premature task answer., i) *Sampling Period or Interaction time ($T(k)_i$)*: It is obtained by Sampling Criteria described by Nyquist in [23]. The RTDF evolution is bounded by Sample Period. The main characteristics of this are: a. $T(k)_i := 1/f(k)_{i\ sampling}$ with $T(0)i = T$, b. $\forall k\ \exists i\ t(k), \ni t(k+1)_i - t(k)_i = T(k)_i$ and, c. $\mu[l(k)_i, LD(k)_{i\ max}) := T(k)_i + \gamma_i$ where μ is a measurable function in the measure theory sense described in [1], and γ_i, represent a tiny time (*jitter*). If $[l(k)_i, LD(k)_{i\ max})$ is empty, i.e., $\mu[l(k)_i, LD(k)_{i\ max}) = 0$ *at the sense described in* [1].

---

[2] If the response is obtained before this lower deadline, it means that the system has a null task In the other hand, if the time is higher that dynamic interval defined by dynamic system period time.

[3] If response is obtained after this deadline, it is bad.

[4] If we sum $l(k)_i$ to $D(k)_{i\ min}$, $D(k)_{i\ max}$ obtain absolute deadlines ($ld(k)_{i\ min}$, $LD(k)_{i\ max}$) [Liu00].

Definition 3 (RTDF : *global constraints*). The RTDF whole tasks have a global deadline, considering that the infimum value in agreement criteria described in [13], could be closed with respect to supremum value allowed by dynamical process.

5.1 FDTR: Global performance

In this section will be described the RTDF global properties in agreement to convergence functional $\{J(m)\}$ when the infimum value tend to ε with $m>0$, and $m \in N^{+}$. The number m represent the interval when the RTDF converge, and ($m_i \uparrow m$, with $i = \overline{1,n}$).

Definition 4 (*Convergence: time t_{i_c}*). The time at which the RTDF converges, is expressed:

$$t_{i\_c} := f_i(k=m), \tag{1}$$

where m is the RTDF convergence interval and t_{i_c} has the condition:

$$d_{i\ c\ min} \le t_{i\ c} < d_i. \tag{2}$$

When d_i is a convergence deadline and and $d_{i\ c\_min}$ is a minimal convergence deadline imposed by physical world. Guarantying a response on time and synchronized with physical world. The shortest minimal convergence time is defined:

$$d_{i\ c\_min} := D(k)_{i\ min} \tag{3}$$

Theorem 1. *The convergence error in probability sense defined by J_m in [13] and [6]. has a value ε_i semi-positive defined, with respect to convergence time described in symbolic form by t_{i_c}.*

Proof. Suppose that ε_i is lower than zero ($\varepsilon_i<0$). The convergence error defined in probability sense is described by the second moment (see for example: [13]) i.e., $M\{\Delta_t \Delta_t^T\} \ge \rho_t$, with ρ_t positive semi defined, M represent the mathematical expectation operator and Δ_t is defined as difference between filtered value and real value (to see: [1]). Now, considering that the $\lim(M\{\Delta_t \Delta_t^T\}_{t \to di}) \to \varepsilon_i$, because the superior limit of ρ_t when $t \to t_{i\_c}$, is bounded by ε_i, and the inferior limit of ρ_t when $t \to t_{i\_c}$, is bounded by zero. Then $\varepsilon_i \ge 0$.

5.3 RTDF local behavior (for TILS)

All RTDF are stable if the parameters are bounded by the unitary circle for all k ([17], [25] and [6]):

$$\left\{a_e(k)\right\}_i \le 1,\ e = \overline{1,n} \tag{4}$$

The estimated parameters whole $\{a_e(k)\}$, represent the proper values of modeled system [15], and it is stable in discrete sense, when the values have been into a unitary circle [17],

[15], and [6]. Outside of unitary circle the response is unstable, and the filter has a bad construction with respect to [6] and [13].

Theorem 2 (Relative maximal deadline $D(k)_{i\_max}$). *A RTDF fulfill the follows condition:*

$$2f_{max}(D(k)_{i\_max}-D(k)_{i\_min}) < 1, \tag{5}$$

f_{i_max} is the dynamical process maximal frequency.

Proof. *Considering the Nyquist criterion [23], the follows condition is true:*

$$f(k)_{i\_muestreo} \geq 2f(k)_{i\_max}, \tag{6}$$

$$2T(k)f(k)_{i\_max} \leq 1, \tag{7}$$

and also, considering to Definition 2 section i, subsections a,b and c; the relative deadline differences is lower than sample time, i.e.:

$$T(k)_i > D(k)_{i\_max} - D(k)_{i\_min}. \tag{8}$$

Using transitivity in inequalities (7) and (8), is clear how to obtain the inequality (6).

5.4 RTDF Computational times and Deadlines

In concurrent sense, the total computational time is defined by the sum of all computing times of tasks.

Generally a RTDF that is implanted in a digital computer with a one processor, the whole of the tasks around of filter will be scheduled in concurrent form:

$$C(k)_i = C(k)_x + C(k)_y + C(k)_a + C(k)_J + C(k)_{au} + C(k)_{ay} + C(k)_{aye} \tag{9}$$

where: $C(k)_x$: Computation time of state equation algorithm., $C(k)_y$: Computation time of observable signal equation algorithm., $C(k)_a$: Computation time of estimator equation algorithm., $C(k)_J$: Computation time of convergence error equation algorithm., $C(k)_{au}$: Computation time of A/D conversion of input $u(t)$., $C(k)_{ay}$: Computation time of A/D conversion of output $y(t)$., $C(k)_{aye}$: Computation time of D/A conversion of estimated output.

To implement a RTDF into PC, required a tool set: *a)D.C. Motor, 20 V, 1 A, 1800 rpm, two poles, permanent field., b) Power unit A/D, D/A, 5 V input, 20 V output, 0.05 A input, 5 A output.*, c) *PC Pentium III 400 MHz, 64 MB RAM., d) ADC card PCL 818L, e) QNX® 4.24 Real-time Operating system., f) MicroPhoton Development Kit®.*

The RTDF ini agreement to (9), require creating next task: Xk: System states algorithm, C_{Xk}= 0.237 ms, Yk: Observable signal algorithm, C_{Yk}= 0.289 ms, Bk: Observable signal variance, C_{Bk}= 0.258 ms, Pk: Ricatti equation algorithm, C_{Pk}= 0.249 ms, Ak: Parameter estimator algorithm, C_{Ak}= 3.252 ms, Jk: Error functional, C_{Jk}= 0.245 ms, Au: A/D conversion of input, C_{Au}= 0.310 ms, Ay: A/D conversion of observable signal, C_{Ay}= 0.302 ms, Aye: D/A conversion of estimate signal, C_{Aye}= 0.314 ms, O: Parent task, C_O = 0.261 ms., $C(k)$ value is in agreement to (9), is: 5.7305 ms.

In an equivalent form in time, we obtain: $T(k)$= 20 ms, $I(k)$ = $T(k\text{-}1)k$ ms, $s(k)$= $I(k)$+ 0.0135 ms, $C(k)$= 5.7305 ms, $D(k)_{min}$= 2.5 ms, $D(k)_{max}$= 20 ms, $f(k)$= 5.7305 ms, $L(k)$= 14.269 ms, $P(k)$= 3.2305 ms. The convergence time proved experimentally is: m= 113 intervals, t_c=2.24 s, d= 3 s, t_c<d is complied. The time value is 0.41 (To see: [21]).

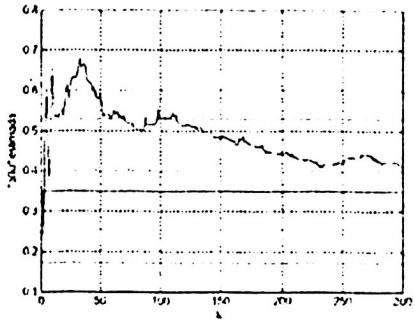


Fig. 2 RTDF: parameter estimator.

6 Conclusions

With base on comments of different authors cited before, a RTDF obey several conditions: Interaction with physical world, good responses, time constraints and filter ability expressed in recursive form, and if we need that a RTDF be critical, then it has to comply all time constraints in all cases and in each interval. Other important characteristic of RTDF is the ability to synchronizing it with dynamical process. A RTDF hasn't delay or pass the time constraints dictated by physical world, in another way a physical world is modify in not desired form. We observe that not all Digital Filters have characteristics of RTDF, is necessary to analyze many conditions: Physical world dynamic, digital filter structure, digital computer, operating system, in the others.

Bibliography

[1] Ash R. "Real Analysis and Probability", Ed. Academic Press, USA, 1970.
[2] Banach S., "The theory of Linear Operations", North-Holland Mathematical Library, 1991.
[3] Baras J. "Symbolic and numeric Real-time signal processing", Technical Report University of Maryland, USA 1999.
[4] Burns A., Wellings A. "Real-time systems and programming languages". University of York, Addison Wesley, 1997.
[5] Buttazzo G. "Hard real-time computing systems". Scuola Superiore S. Anna, Kluwer Academic Publishers, 1997.
[6] Caines P. "Linear Stochastic Systems". Ed. Wiley, Canada, 1986.
[7] Chui C., Chen G. "Kalman Filtering with Real-time Applications". Ed. Springer, USA 1999.
[8] De la Puente J. A. "Introducción a los Sistemas en Tiempo Real" Universidad Politécnica de Madrid, 2000.
[9] Grewal M. and Andrews A. Kalman filtering, theory and practice. Prentice Hall: Information and System Sciences Series. USA 1993.
[10] nedenko, B. V., the Theory of Probability, Ed. Chelsea Publising Company, USA, 1962.
[11] Gray D. "Introduction to the formal design of Real-Time Systems", Ed. Springer, UK1999.
[12] Gustafsson F. "Adaptive filtering and change detection". Ed. John Wiley & Sons, LTD, Linkoping University Sweden 2000.
[13] Haykin S. "Adaptive filter theory". Prentice Hall: Information and System Sciences Series, USA, 1991.
[14] Heitmeyer C., Mandrioli D. "Formal Methods for Real-Time Computing" Ed. Wiley, England 1996.
[15] Kailath T. "Linear Systems", Prentice Hall, USA 1980.
[16] Krten R. Getting started with QNX 4, PARSE Software Devices, and Canada 1998.
[17] Kuo B. C. "Sistemas de control automático". Prentice Hall 7ª edición, México, 1996.
[18] Liu J. "Real-time Systems" Ed. Prentice Hall USA 2000.
[19] Liu C., Layland J. "Scheduling algorithms for multiprogramming in hard-real-time environment". ACM, Vol. 20, No. 4, USA 1982.
[20] Martín J. "Design Real-time computer systems", Ed. Diana México 1980.
[21] Medel J. J. "Análisis de dos métodos de estimación para sistemas lineales estacionarios e invariantes en el tiempo" Computación y sistemas volumen 6 número 1, México 2002.
[22] Medel J. J . , & Poznyak A. S., Adaptive Tracking for DC- derivate motor Based on Matriz Forgetting, C y S, pp. 201-217, 2001.
[23] Nyquist H. Certain Topics in Telegraph Transmission Theory. AIEE Transactions, USA 1928.
[24] Papoulis A., Bertrán M. "Sistemas y circuitos", Ed. Marcombo, 1ª edición, España 1989.
[25] Rudín W. (1988). Análisis real y complejo. Universidad de Wisconsin, McGraw Hill.
[26] Stankovic J. "Real-time Computing". Technical Report, University of Massachusetts, 1992

MATRICIAL ESTIMATION OF START TIMES WITH STOCHASTIC BEHAVIOR OF PERIODIC REAL TIME TASKS IN A CONCURRENT SYSTEM

P. Guevara López [1,2], D. Cruz Pérez [2,3], J. J. Medel [3,2], A. Flores Rueda[3,2]

[1] DEMS, CICATA- I. P. N. [2] CICATA-I. P. N., [3] CIC, CIITEC, CICATA – I. P. N.,
Tel.57296000 Ext. 50427, Ext. 64344
(pguevara, jjmedel, aflores) @ipn mx, dcruzp@yahoo.com

Abstract. Using a multivariable dynamic model for describing startup times of real Time Tasks supposing the following considerations: The system is stationary, first order, with jitter and external perturbations bounded with a normal distribution without correlation that closely represent periodical behavior of real time tasks. To bear closer the task model in a concurrent system, systems internal dynamics are required, those are represented through the parameter matrix in function of output vectors in the regressive model as is for perturbations, but knowing their value is work for a multivariable estimator. Results of an example performed on a real-time platform are presented, considering periodic and concurrent tasks, an instrumental variable algorithm is used because of his good convergence time and his relatively easy implementation.

Keywords: Startup time, estimation, real time, periodic tasks, instrumental variable.

1 Introduction

Periodic tasks are usually found in several applications like airplanes and control process where uniform monitoring is required. Modeling them is not simple because each of those needs an adequate representation. A set of periodic concurrent tasks may be represented using a multivariable state model where in explicit way the model is function of the internal dynamic and previous states of the system's output information.
These matrix and perturbations are going to give the start times characteristics through system evolution. A parameter matrix estimator is needed in order to adjust the model for reconstruction, tracing and prediction in real time.

2 Start Times Model in PRTT (Periodic Real Time Tasks)

As seen in [1], [4], [5], [6], [10], [14] y [15], PRTT model is represented by a stochastic, stationary, first order and first grade type difference equation; Considering that external processor perturbations are not correlated and obey a normal distribution function.

Proposition 1. (Absolute Arrival Time for RTT). The vector of absolute arrival times L_k of an instance set with index k is described by:

$$L_k = L_{k-1} + \Pi_k . \tag{1}$$

Proposition 2. (Inter-arrival Time for RTT). The Inter-arrival time vector Π_k of an instance with index k of a RTT J_i set is describes as:

$$\Pi_k = A_k(\Pi_{k-1} - W_{k-1}) + U_k + W_k . \tag{2}$$

Where: *A_k is the system parameter matrix with unknown dynamics bounded in agreement with [7], [11], [12] y [14]; Π_k is the Inter-arrival times vector of instances with index k; W_k is the external processor perturbations vector, represented through random variables with Gaussian distribution; U_k is the Inter-arrival times vector of reference.*

Proposition 3. (Start Times for RTT). The starting times vector S_k of a feasible task with index k of a set of RTT J_i is described with:

$$S_k = S_{k-1} + \Pi_k + V_k - V_{k-1} . \tag{3}$$

Comment 1. If an inter-start times vector Π'_k is modeled it is just needed to add the *jitter* V_k as internal perturbation to the state equation such that:

$$\Pi'_k = X'_k + W_k , \tag{4}$$

$$X'_k = A_k X'_{k-1} + U_k + V_k , \tag{5}$$

$$\Pi'_k = A_k(\Pi'_{k-1} - W_{k-1}) + U_k + W_k + V_k . \tag{6}$$

Proposition 4. (Periodic Tasks in Real Time). A set of PRTT is that J_i where all their instances have inter-arrival times vectors Π_k close to a periodic vector T_k.

Para A_k $\quad \{a_{i,j,k}\} \subset [0,1]$ $\quad a_{i,j,k}$ constante $\forall\ i,j,k \in Z^+$

Para U_k $\quad \{u_{i,k} = T_i - a_{i,k}T_j\}$ $\quad u_{i,k}$ constant $\forall\ i,k \in Z^+$

3 Real Time Parameter estimator (RTPE)

In order to trace the parameter matrix dynamic of the concurrent PRTT a Real Time Estimator is needed, which is defined in agreement to [3], [7],[10], [14], [15], [16] as:

Definition 1. (Real Time Parameter Estimator RTPE). All RTPE is a digital filter with the following conditions:

a. Extraction and emission of observable information (input and outputs), where $\{u(k)_i \in \mathbf{U}(k)\}$ & $\{y(k)_j \in \mathbf{Y}(k)\}$, with $i, j, k \in \mathbf{Z}^+$, in criteria of [2], [3], [7], [8], [11], [12] y [13],
b. To give correct answers on the mater to the process considering some reestablished criteria as those shown in [2],[11], [12] y [13],
c. Being expressed recursively (see [3], [7], [10], [13], [17]),
d. Convergence value will be bounded in an infinite interval in which it will be varying the convergence value.
e. Matrix operation usage in agreement with process dynamic for each iteration, respecting process dynamics restrictions.

Proposition 5. (Convergence in all multivariables RTPE). All RTPE as a parameter estimator has an error functional bounded (see: [2],[7], [9], [16] y [17]), such that :

$$m^* = \operatorname{argmin} \inf_{k \geq m} P\{|\hat{a}_k - a| \leq \Delta\} = 1, \tag{7}$$

Where Δ is the error bound defined by noise variance, m is the convergence interval and m^* is the set of intervals where the RTPE has converged.

4 Estimation of Periodic Real Time Task Start Times with stochastic jitter using the Instrumental Variable technique.

As an example a set of 2 concurrent PRTT is analyzed. The set of relative arrival times Π_k is used as the observable signal and the parameter matrix A_k is going to be estimated, such that:

$$\begin{aligned} &E\{W_{k-1}\Pi_k^T\}=[0], \quad E\{W_k\Pi_k^T\}=\Theta_{w_k}^2, \quad E\{V_k\Pi_k^T\}=\Theta_{v_k}^2, \\ &E\{W_k(W_k)^T\}=\Theta_{w_k}^2, \quad E\{V_k(V_k)^T\}=\Theta_{v_k}^2, \; E\{V_k(W_k)^T\}=[0]. \end{aligned} \tag{8}$$

The estimator is expressed as:

$$\hat{A}_k := (\hat{A}_{k-1} B_{k-1} + \Pi_k Z_k^T) B_k^{-1}. \tag{9}$$

This representation obeys the definitions of a RTPE.
The estimation error is defined as in [2], [3], [7] y [9]:

$$\Delta_k = |\hat{A}_k - A|. \tag{10}$$

In agreement with [9] the estimation error is described as:

$$\Delta_k := ((-A\Theta_{w_k}^2 + \Theta_{v_k}^2)(I - A^2))(\Theta_{w_k}^2 (2A^2 + I + A^2) + \Theta_{v_k}^2 (I + 2A))^{-1}, \tag{11}$$

As the error functional in accordance with [9] y [15]:

$$J_k = E(\Delta_k (\Delta_k)^T). \tag{12}$$

The following data was considered as results of the estimation algorithm validation:

$$A=\begin{bmatrix}0.5 & 0.2\\ 0.3 & 0.4\end{bmatrix},\ \Theta^2_{v_k}=\begin{bmatrix}0.95 & 0.93\\ 0.97 & 0.94\end{bmatrix},\ \Theta^2_{w_k}=\begin{bmatrix}0.95 & 0.93\\ 0.97 & 0.94\end{bmatrix}. \tag{13}$$

For the experimental implementation of the RTPE the following was considered:

a. The maximal system deadline $\mathbf{D}_{k,max}$ is equal to period $\mathbf{T}_k$.
b. The Start time ($\mathbf{S}_k$- $\mathbf{L}_k$) was obtained such that $\mathbf{S}_k = \mathbf{L}_k + 0.0015$ ms.
c. The sampling period T_k (Real Time temporizator impulse) for task activation is 10 ms.
d. Minimal deadline $\mathbf{D}_{k\,min}$= 1 ms.
e. Convergence deadline of the PRTT set is: d= 4 s.

For an experiment of the RTPE Figure 1 was obtained with the following results:

m_{max}= 354 intervals, $t_{c\ max}$= 3,54 s. The convergence time is t_c =3.54 s.

a. m= [354, 213, 241, 318] intervals,
b. t_c= [3.54, 2.13, 2.41, 3.18] s,
c. d= 4 s. (400 intervals).

Figure1 y Figure 2 are just examples of a RTPE behavior used as a parameter estimator trough the instrumental variable technique.

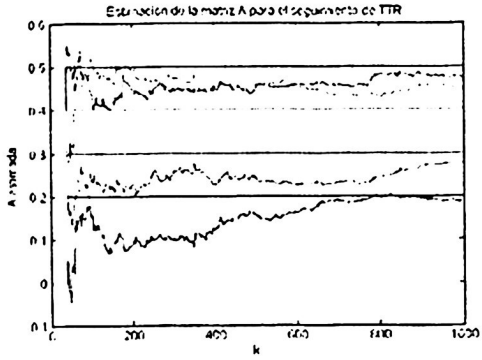


Fig.1 *Graphics of parameter matrix estimation "A" using the RTPE..*

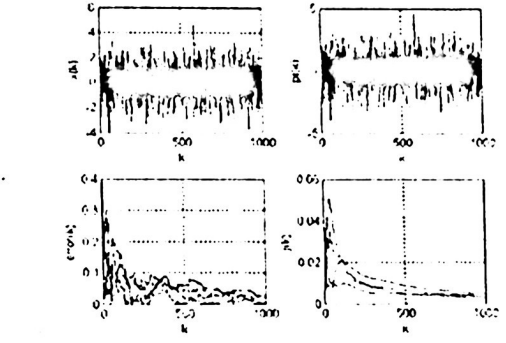


Fig. 2 *Internal state X_k, observable signal Π_k, estimation error and functional j_k for the RTPE.*

5 Conclusions

The presented Start times model considers internal dynamics of the system for Periodic Real Time Tasks in a concurrent system. Definitions were made for tasks depending on their arrival and start times and were characterized in accordance with his parameter or parameter matrix, his entrance and computing equipment's internal and external perturbations. The model showed to be capable of characterize several Real Time Task behaviors and obey characteristics mentioned by several authors. Important things to mention in this kind of real time filtering are: task characteristics, synchronicity, sampling periods and convergence time. For start time estimation, the convergence times where bounded by a Δ estimation error bound defined by the second moment of probability of the internal and external perturbations; convergence periods were acceptable. Instrumental Variable technique was used because of his good response time and his convergence time obeys Real Time System conditions, approaching to real parameters rapidly and in deadlines.

References

[1] Baras J. (1999), "*Symbolic and numeric Real-time signal processing*", Technical Report University of Maryland, USA. pp 226

[2] Caines P. (1986). "*Linear Stochastic Systems*". Ed. Wiley, Canada.

[3] Chui C., Chen G. (1999), "*Kalman Filtering with Real-time Applications*". Ed. Springer, USA.

[4] Cruz-Pérez D. (2004). "Modelo Dinámico para Tareas en Tiempo Real". Tesis de Maestría en Ciencias de la Computación, 7 de agosto de 2004, CIC-IPN, México D.F.

[5] Ecker K. H. (2000). *"Overview on Distributed Real-time Systems. Requirements, Analysis, Specification, Operating Systems, Case Studies"*. Institute of Informatik, Technical University of Clausthal, Germany.

[6] Guevara P., Medel J. J., Cruz D. (2004). *"Modelo Dinámico para una Tarea en Tiempo Real"*. Revista Computación y Sistemas, ISSN 1405-5546, Vol. VIII No. 1, México, Septiembre de 2004.

[7] Haykin S. (1991). *Adaptive filter theory*. Prentice Hall information and system sciences series.

[8] Kotel'nikov V. A. (1933). *"On the transmission capacity of "ether" and wire in electrocommunications "*. Izd, Red. Upr. RKKA (Moscow URSS) (Material for the first all-union conference on questions of communications), vol. 44, 1933.

[9] Medel J. J. (2002). *"Análisis de dos métodos de estimación para sistemas lineales estacionarios e invariantes en el tiempo con perturbaciones correlacionadas con el estado observable del tipo: Una entrada una salida"* Computación y sistemas volumen 6 número 1, México.

[10] Medel J., Guevara P., Flores A. *"RTMDF: Real-Time Multivariable Digital Filter"*. International IEEE Workshop Signal Processing 2003, Poznan Polonia, October2003.

[11] Nyquist, H. (1928). *Certain Topics in Telegraph Transmission Theory*. USA. AIEE Transactions.

[12] Shannon C. E. (1948), *"A mathematical theory of communication"*. Bell Syst. Tech. J. vol. 27, pp. 379-423, 623-656, July-Oct.

[13] Whittaker E. T. (1915), *"On the functions which are represented by the expansion of interpolation theory"*. In Proc. Roy. Soc. Edimburgh, vol. 35, pp. 181-194.

[14] Guevara P., Medel J.J., Cruz D. (2004). Modelo Dinámico para Tiempos de Arribo de una Tarea en Tiempo Real. Revista Computación y Sistemas, ISSN 1405-5546, Vol. VIII No. 1, pags. 190-209, México, Septiembre de 2004.

[15] J.J. Medel, P. Guevara, A. Poznyak. (2004). "Real-time Multivariable Digital Filter using Matrix Forgetting Factor and Instrumental Variable". Automatic Control and Computer Sciences Vol. 38, No. 1 pages 40-53, ISSN 0132-4160, February. 2004, (ISI), Latvia.

[16] J.J. Medel, P. Guevara, A. Flores. (2004). "Caracterización de Filtros Digitales en Tiempo Real para Computadoras Digitales". Revista Computación y Sistemas, ISSN 1405-5546, Vol. VII No. 3, México.

[17] Guevara-López P. (2004) "Resumen de Tesis Doctoral". Revista Computación y Sistemas, ISSN 1405-5546, Aceptado para publicación, Septiembre de 2004.

Control

Diagnosis of a nonlinear system with stabilizing control

J. C. Cruz-Victoria., R. Martínez-Guerra
CINVESTAV-IPN
A. P. 14-740, C. P. 07360,
México, D. F., MEXICO
e-mail: jvictoria(rguerra)@ctrl.cinvestav.mx

Abstract. In this paper is used the nonlinear systems diagnosis by means of differential algebraic techniques to obtain an estimate of the fault vector. Then, with these estimates of the fault components is obtained a stabilizing control law for the states. this result is illustrated by an example of a nonlinear system with control in one of its states, where the system is affected by the action of two additive faults. It will also be shown by means of numerical simulations that under this type of faults the state stays stable.
Keywords.- *Diagnosis, Stabilizing control law.*

1 Introduction

Systems diagnosis has been studied for more than three decades, see for instance [10]. In [11] a direct extension of the unknown input observer (UIO) results in linear systems to the nonlinear case was considered. An alternative to the nonlinear unknown input observer approach in nonlinear uncertain systems was proposed by Seliger and Frank [9], where the presence of modelling uncertainties is not taken into account, however, the reader is referred to the works of Diop and Martínez-Guerra [1], [2], where the presence of uncertainties is included using this methodology. For instance, [1], [2] presents an algebraic approach to solve the diagnosis problem. It consists on translating the solvability of the problem in terms of the algebraic observability of the variable which models the *fault*. The framework in which this paper is conceived is based essentially in the language of differential algebra. In [4], [5], [8], the methodologies employed for the observer design only include full order observers without considering uncertainty estimation, however, in this communication, the fault dynamics is considered as an uncertainty. In the proposed procedure, it is not necessary the construction of a full order observer, instead, a reduced order uncertainty observer is constructed using differential algebraic techniques applied to the fault estimation in the diagnosis problem.
The *main results* of this paper are: the estimation of the faults using differential algebraic techniques and the construction of a stabilizing control law for the

states depending on the fault estimates, this result is illustrated by an example of a nonlinear system with control in one of its states, where the system is affected by the action of two additive faults. It will also be shown by means of numerical simulations that under this type of faults the system stays stable.
The rest of this paper is organized as follows: in Section 2 some basic definitions on observability and systems diagnosability in a differential algebraic framework are introduced. Statement of the problem and the diagnosability condition are described in Section 3. In Section 4 an example with the application of the proposed methodology is shown. In Section 5, the construction of a reducer order uncertainty observer is described. In section 6, we show the numerical results. Finally, in Section 7 we will close the paper with some concluding remarks.

2 Basic Definitions

Before starting, some differential algebra definitions are introduced [3], [4], [6].

Definition 1. *A differential field extension L/k is given by two differential fields k and L, such that: i) k is a subfield of L, ii) the derivation of k is the restriction to k of the derivation of L.*
Example $\mathbb{Q}$,$\mathbb{R}$ *and* $\mathbf{C}$ *are differential field extensions, where* $\mathbb{Q} \subset \mathbb{R} \subset \mathbf{C}$.

Definition 2. *An element is said to be differentially algebraic with respect to the field k if it satisfies a differential algebraic equation with coefficients over k.*
Example $\mathbb{R}\langle e^{at}\rangle/\mathbb{R}$ *is a differential field extension* $\mathbb{R} \subset \mathbb{R}\langle e^{at}\rangle$, $x{=}e^{at}$ *is a solution of* $P(x){=}\dot{x}{-}ax{=}0$ *(a is a constant).*

Definition 3. *An element is said to be differentially transcendental over k, if and only if, it is not differentially algebraic over k.*

Definition 4. *A dynamics is a finitely generated differential algebraic extension* $G/k\langle u\rangle$ *(*$k\langle u,\xi\rangle$,$\xi \in G$*). Any element of G satisfies an algebraic differential equation with coefficients being rational functions over k in the elements of u and a finite number of their time derivatives.*
Example Let consider the input-output system $\ddot{y} + \omega^2 sen\,(y) = u$, *equivalent to the system:*

$$\Sigma_B \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\omega^2 sen(x_1) + u \\ y \ = x_1 \end{cases} \qquad (1)$$

System (1) is a dynamics of the form $\mathbb{R}\langle u,y\rangle/\mathbb{R}\langle u\rangle$ where $G = \mathbb{R}\langle u,y\rangle$, $y \in G$ and $k = \mathbb{R}$. Any solution of (1) satisfies the following algebraic differential equation:

$$\left(y^{(3)} - \dot{u}\right)^2 + \left(\dot{y}\left(y^{(2)} - u\right)\right)^2 = (\omega\dot{y})^2 .$$

Definition 5. *Let a subset $\{u,y\}$ of G in a dynamics $G/k\langle u\rangle$. An element in G is said to be algebraically observable with respect to $\{u,y\}$ if it is algebraic over $k\langle u,y\rangle$. Therefore, a state x is said to be algebraically observable if, and only if, it is algebraically observable with respect to $\{u,y\}$. A dynamics $G/k\langle u\rangle$, with output y in G is said to be algebraically observable if, and only if, all its states have this property.*

Example System Σ_B in (1) with output $y \in \mathbb{R}\langle u,y\rangle$ is algebraically observable, since x_1 and x_2 satisfies two algebraic differential polynomials with coefficients in $\mathbb{R}\langle u,y\rangle$, i.e.

$$x_1 - y = 0$$
$$x_2 - \dot{y} = 0.$$

3 Statement of the Problem

Let consider the class of nonlinear systems described by [2], [6]:

$$\begin{aligned} \dot{x}(t) &= A(x,\bar{u}) \\ y(t) &= h(x,u) \end{aligned} \tag{2}$$

Where $x = (x_1,...,x_n)^T \in R^n$ is a state vector, $\bar{u} = (u,f) = (u_1,...,u_{m-\mu}, f_1,...,f_\mu) \in R^{m-\mu} \times R^\mu$ where u is the stabilizing control (a known input vector) and f is an unknown fault vector, $y = (y_1,...,y_p) \in R^p$ is the output, A and h are assumed to be known analytical vector functions.
Then, the problem is to estimate the fault vector to obtain a stabilizing control depending on these fault estimates in order to stabilize the state vector.

Definition 6. *Given a fault f, it is called algebraically observable if each component f_i of the fault is algebraic over the differential field $k\langle u,y\rangle$.*

Definition 7. *An element $f \in k\langle \bar{u},y\rangle$ is said to be algebraically observable if f satisfies a differential algebraic equation with coefficients over $k\langle u,y\rangle$.*

Definition 8. *The class of nonlinear systems described by (2) is said diagnosable if it is possible to estimate the fault f from the system equations and the time histories of the data u and y, i. e., it is diagnosable if f is algebraically observable with respect to u and y.*

Remark It was already pointed out [2] that a diagnosable system need not to be observable, and vice versa. Indeed, the following system

$$\begin{cases} \dot{x}_1 = -x_1 + x_2, \\ \dot{x}_2 = x_2 + u + f, \\ y = x_2, \end{cases} \tag{3}$$

is diagnosable, i. e. $f = \dot{y} - y - u$, but it is not observable since x_1 is not observable with respect to u and y.

4 Example

In the following paragraphs is used the proposed methodology to obtain the diagnosability conditions of the fault components.
Consider the following nonlinear system

$$\begin{aligned} \dot{x}_1 &= -x_1 + f_1 x_2^3 + f_2 x_2 x_3 + u \\ \dot{x}_2 &= x_3 + f_1 \\ \dot{x}_3 &= -x_2^3 + f_2 \\ y_1 &= x_2 \\ y_2 &= x_3 \end{aligned} \tag{4}$$

where the control u can be naturally expressed as

$$u = -\hat{f}_1 x_2^3 - \hat{f}_2 x_2 x_3$$

and replacing u in $\dot{x}_1$ is obtained

$$\dot{x}_1 = -x_1 + \left(f_1 - \hat{f}_1\right) x_2^3 + \left(f_2 - \hat{f}_2\right) x_2 x_3 \tag{5}$$

It is observed that if the second and third term of the right of the equation (5) become equal to zero, then the state x_1 will be stable. With this purpose, it is necessary to obtain an estimate of the fault components and this is achieved imposing certain conditions, which will be presented in the observer synthesis. By replacing y_1 and y_2 in (4) it is not hard to obtain

$$\begin{aligned} \dot{y}_1 - y_2 - f_1 &= 0 \\ \dot{y}_2 + y_1^3 - f_2 &= 0 \end{aligned} \tag{6}$$

Then, system (4) is diagnosable and the fault components satisfies the following algebraic equations over $k\langle u, y\rangle$:

$$\begin{aligned} f_1 &= \dot{y}_1 - y_2 \\ f_2 &= \dot{y}_2 + y_1^3 \end{aligned} \tag{7}$$

where this equations are called the diagnosability conditions for f_1 and f_2.

5 Observer Synthesis

Once that we have the diagnosability conditions of the faults it is necessary to propose the construction of an observer, to obtain by means of numerical simulations the estimates of these faults, as follows.

Let consider system (2). The fault vector f is unknown and it be assimilated as a state with uncertain dynamics. Then, to estimate it the state vector is

extended to deal with the unknown fault vector. The new extended system is given by

$$\begin{aligned} \dot{x}(t) &= A(x, \bar{u}) \\ \dot{f} &= \Omega(x, \bar{u}) \\ y(t) &= h(x, u) \end{aligned} \tag{8}$$

where $\Omega(x, \bar{u})$ is a bounded uncertain function.

The following hypotheses are assumed:
H1: $\Omega(x, \bar{u})$ is bounded, i.e., $|\Omega(x, \bar{u})| \leq M$.
H2: $f(t)$ is algebraically observable over $k\langle u, y\rangle$.
H3: γ is a $\mathbb{C}^1$ real-valued function.

Next Lemma describes the construction of a proportional reduced order observer for (8).

Lemma 2 The system

$$\dot{\hat{f}} = K\left(f - \hat{f}\right) \tag{9}$$

is an asymptotic reduced order observer for system (8), where $\hat{f}$ denotes the estimate of f, f is given by its algebraic equation with coefficients in $k<u.y>$ and $K \in \mathbb{R}^+$ determines the desired convergence rate of the observer, if the following assumption is satisfied:

H4: $\left|e^{-\int K dt}\right| = 0$ with t_0 sufficiently large and $\limsup_{t \to t_0} \frac{\Delta f}{|K|} = 0$.

Sometimes, the output time derivatives (which are unknown), appear in the algebraic equation of the fault, then, it is necessary to use an auxiliary variable to avoid using them.

Corollary The dynamic system (9) along with

$$\dot{\gamma} = \psi(x, \bar{u}, \gamma), \text{ with } \gamma_0 = \gamma(0) \text{ and } \gamma \in \mathbb{C}^1 \tag{10}$$

constitute a proportional asymptotic reduced order fault observer for the system (8), where $\gamma \in \mathbb{C}^1$ is a change of variable which depends on the estimated fault $\hat{f}$, and the states variables.

Further details can be found in [7].

Remark It should be noted that f in (9) is obtained from the algebraic observability condition, that is to say, f is replaced in (9) by its algebraic equation in

$k\langle u, y\rangle$.

The performance of the reduced order observer estimator is shown by means of numerical simulations.

6 Numerical results

The following equation represents the uncertainty dynamics of the fault components:

$$\dot{f}(t) = \Omega(x, \bar{u})$$

In order to estimate the fault variable f, the following reduced order observer is proposed:

$$\begin{aligned} \dot{\hat{f}}_1 &= K_1\left(f_1 - \hat{f}_1\right) \\ \dot{\hat{f}}_2 &= K_2\left(f_2 - \hat{f}_2\right) \end{aligned} \tag{11}$$

then, from (7) and (11) is obtained:

$$\begin{aligned} \dot{\hat{f}}_1 &= K_1\left(\dot{y}_1 - y_2 - \hat{f}_1\right) \\ \dot{\hat{f}}_2 &= K_2\left(\dot{y}_2 + y_1^3 - \hat{f}_2\right) \end{aligned}$$

Note that $\dot{y}_1$ and $\dot{y}_2$ are not available. However, the following auxiliary variables allows to circumvent this problem. Define

$$\begin{aligned} \gamma_1 &= \hat{f}_1 - K_1 y_1 \\ \gamma_2 &= \hat{f}_2 - K_2 y_2 \end{aligned}$$

Then, the reduced order observer is given by

$$\begin{aligned} \dot{\gamma}_1 &= -K_1\left(y_2 + \gamma_1 + K_1 y_1\right) \\ \dot{\gamma}_2 &= K_2\left(y_1^3 - \gamma_2 - K_2 y_2\right) \\ \hat{f}_1 &= \gamma_1 + K_1 y_1 \\ \hat{f}_2 &= \gamma_2 + K_2 y_2 \end{aligned} \tag{12}$$

where $\gamma_1, \gamma_2 \in \mathbb{C}^1$

The simulation results are obtained with initial conditions γ_1=$\gamma_2 = 0$ and $K_1 = 5$, $K_2 = 2$. In figure 1 the estimates of the faults which converges to the current faults are shown. Figure 2 describes the numerical simulations corresponding to the state $x_1$without stabilizing control and the stable dynamic of state x_1 under the action of the stabilizing control law.

7 Concluding Remarks

In this paper was presented a stabilizing control for the states depending on the fault estimates, this result is illustrated by an example of a nonlinear system with control in one of its states, where the system is affected by the action of two additive faults. It is also shown by means of numerical simulations that under this type of faults the state x_1 remains stable.

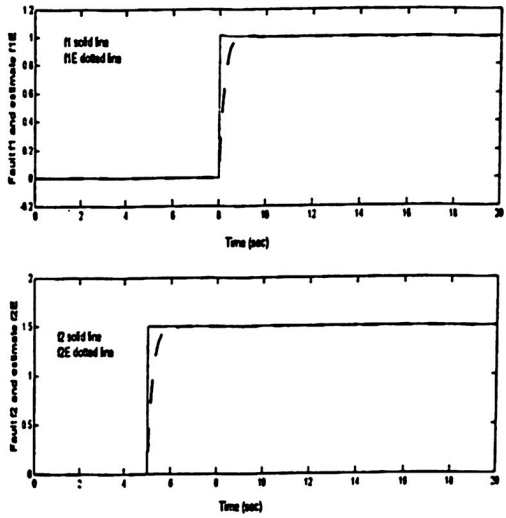


Fig. 1. True dynamics of the fault steps at times 5 and 8 seconds in solid line and its estimates in dotted line.

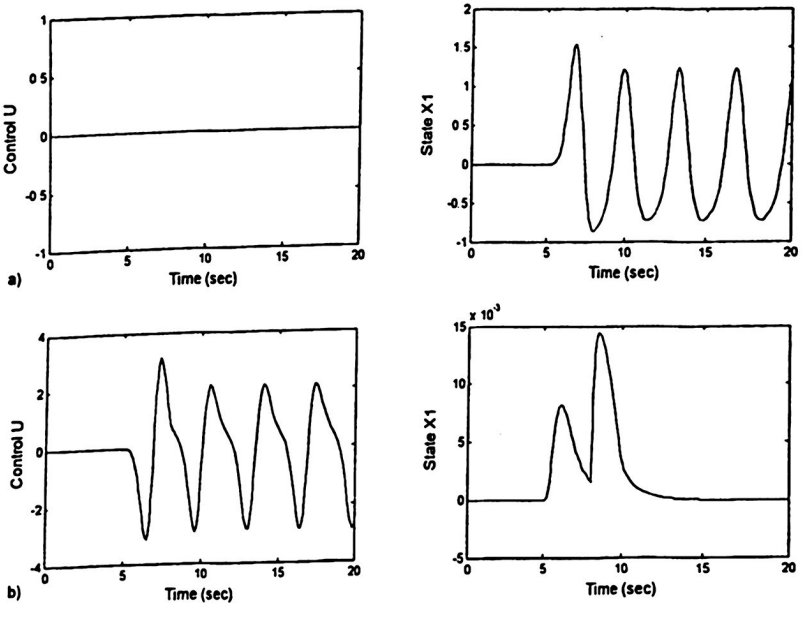


Fig. 2. a) Simulation results of state x_1 without control. b) Numerical results of the state x_1 under the action of the stabilizing control u.

References

1. Diop, S. and Martínez-Guerra, R. *An algebraic an data derivative information approach to nonlinear systems diagnosis.* Proceedings of the European Control Conference 2001,Porto, Portugal,ECC01, 2001, pp. 2334-2339.
2. Diop, S. and Martínez-Guerra, R. *On an algebraic and differential approach of nonlinear systems diagnosis.* In the Proceedings of the IEEE Conference of Decision and Control, CDC01, 2001, OrlandoFL, USA, pp. 585-589.
3. Fliess M., *A note on the invertibility of non-linear input-output differential systems* (1986), Systems and Control Letters, 8, 147-151.
4. Martínez-Guerra R. and De León-Morales J. *Nonlinear estimators: A differential algebraic approach.* Appl. Math. Lett., Vol. 9, No. 4, pp. 21-25, 1996.
5. Martínez-Guerra, R., Garrido, R. and Osorio-Mirón, A. *High-gain nonlinear observers for the fault detection problem: application to a biorreactor, in* IFAC Publications, Editorial Elsevier Sc. Ltd, Nonlinear Control Systems, Edits: Kurzhanski/Fradkov, Vol. 3, ISBN 0-08-043560-2, pp. 1567-1572, 2002.
6. Martínez-Guerra R, Diop S., Garrido R. and Osorio Mirón A. *Diagnosis of nonlinear systems using a reduced order fault observer: Application to a biorreactor.* Journées Franco-Mexicaines d'Automatique Appliquée, 12-14 Septembre 2001, IRCCyN, Nantes, France.
7. Martínez-Guerra, R., Mendoza-Camargo, J. Observers for a class of nondifferentially flat systems. IASTED Circuits, Signals & Systems (CSS2003), Cancún, México, Mayo, pp.67-72, 2003.
8. Martínez-Guerra, R., Ramírez Palacios, I. R. and Alvarado-Trejo, E. *On parametric and state estimation: application to a simple academic example.* Proc. IEEE 37th Conf. on Dec. and Control, pp. 764-765, 1998.
9. Seliger, R. and Frank, P. M. *Robust observer-based fault diagnosis in nonlinear uncertain systems. In issues of fault diagnosis for dynamic systems,* Eds. Patton, Frank, Clark, Springer, pp. 145-187, 2000.
10. Willsky, A. S. *A survey of design methods for failure detection in dynamic system.* Automatica, 12, pp. 601-611, 1976
11. Wünnenberg *Observer-based fault detection in dynamic system.* VDI-Fortschrittsber., VDI-Verlag, Reihe 8, Nr. 222, Düsseldorf, Germany, 1990.

Fault Diagnosis Methods for AC Induction Motors

Rodrigo López Cárdenas[1] and Luis Pastor Sánchez Fernández[2]

[1] Center for Computing Research
National Polytechnic Institute, México
rodrigo@correo.cic.ipn.mx
[2] lsanchez@cic.ipn.mx

Abstract. This paper presents a review and evaluation of developments in the field of diagnosis of electrical machines. It covers model-based techniques, knowledge-based techniques and signal techniques, organized in quantitative approach and qualitative approach. Exposes briefly drawbacks and benefits of principal methods and their combination. Then compares the principal methods and finally, suggests an hybrid method that can overcome their individual limitations and take advantage of particular merits.

1 Introduction

Motors are the workhorses or our industry. Safety, reliability, efficiency, and performance are some of the major concerns and needs for motor systems applications. They are essential components in most of today's manufacturing and production industries. The key for a successful motor operation are a quality motor, understanding of the application, choice of the proper maintenance of the motor [1] [2].

A fault is to be understood as a non permitted deviation of a characteristics property which leads to the inability to fulfill the intended purpose. This can be done by checking if particular measurable or unmeasurable estimated variable are within certain tolerance of the normal value. If this check is not passed, this leads to a fault message.The functions up to this point are usually called monitoring or fault detection, this is followed by a fault diagnosis: the fault is located and the cause of it is established [3].

2 Quantitative approach

Previous supervision of technical process was restricted to checking directly measurable variables for upward or downward transgression of fixed limits or trends. This could be automated by using simple limit-value monitors. Various faults in the process could be then detected, but only after the measurable output values had been effected considerably. The problem is orientate process faults with the

aid of the measurable input and output variables $U(t)$ and $Y(t)$ and mathematical models with nonmeasurable disturbance signals, nonmeasurable process parameters and partially measurable and partially nonmeasurable internal state variables [3].

2.1 Measurable signals

Measurable input signals $U(t)$ and output signals $Y(t)$ can be directly used to monitor changes in the process.

Limit and trend checking is a method well know and commonly used to check signal's limits $Y(t)$, is this excess a maximum value Y_{max} or fallen below of a minimum value Y_{min}. The normal state is:

$$Y_{min} < Y(t) < Y_{max} \tag{1}$$

This is referred to as an absolute value check. The limit check can also be applied on the trend $\dot{Y}(t)$ of the signal $Y(t)$. The normal state is:

$$\dot{Y}_{min} < \dot{Y}(t) < \dot{Y}_{max} \tag{2}$$

also a combination of absolute value and trend checking is possible.

If only limit checking is applied, the limits usually are set on safe side to allow sufficient time for counteractions. However, this can lead to false alarms if the variable return to the normal state without external actions. This disadvantage can be avoided, if the affected signals $Y(t)$ can be predicted. This also allows to predict the time of exceeding a threshold. In order to this, mathematical models have to be used [3].

2.2 Mathematical models

The task consist of the diagnosis of faults in a dynamical system by measuring the available input and output variables $U(t)$ and $Y(t)$. Process with parameters that can be linearized around one operating point are usually described by an ordinary differential equation

$$\begin{aligned} y(t) + a_1\dot{y}(t) + a_2\ddot{y}(t) + \ldots + a_n y^{(n)}(t) &= b_0 u(t) \\ +b_1\dot{u}(t) + b_2\ddot{u}(t) + \ldots + b_m u^{(m)}(t) & \end{aligned} \tag{3}$$

Additive faults at the input or output can be modeled by [4]

$$\begin{aligned} y(t) + a_1\dot{y}(t) + a_2\ddot{y}(t) + \ldots + a_n y^{(n)}(t) &= b_0 u(t) \\ +b_1\dot{u}(t) + b_2\ddot{u}(t) + \ldots + b_m u^{(m)}(t) + f_y + b_0 f_u & \end{aligned} \tag{4}$$

The process model parameters $\theta^T = [a_1 \ldots a_n \,|\, b_1 \ldots b_m]$ are more o less intricate relationships of several physical process coefficients, e.g. mass, speed, drag coefficient, viscosity, resistances. If a fault within the process changes one or several parameters by $\Delta\theta_j$ the output signal changes and the parameter estimation

indicates a change $\Delta\theta(t)$. Generally, the process parameters θ, depend on the physical process coefficients p. $\theta = f(p)$ via nonlinear algebraic equations. If the inverse of this relationship exists

$$p = f^{-1}(\theta) \tag{5}$$

then changes Δp_i of the process coefficients can be calculated. These changes in process coefficients are in many cases directly related to faults [3]. A necessary requirement of this procedure is, however, the existence of the inverse relationship Eq. 5.

The linear process can be described in state-space form as

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{6}$$

$$y(t) = Cx(t) \tag{7}$$

To reconstruct the states from measurable input and output signals a state observer can be used, the feedback matrix H must be selected properly to assure that the observer is stable, where $e(t)$ is the output error.

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + He(t) \tag{8}$$

$$e(t) = y(t) - C\hat{x}(t) \tag{9}$$

The process is now influenced by disturbances and faults as fallows [4]

$$\dot{x} = Ax(t) + Bu(t) + Fv(t) + Lf_L(t) \tag{10}$$

$$y(t) = Cx(t) + Nn(t) + Mf_M(t) \tag{11}$$

$v(t)$ and $n(t)$ represent the unmeasurable disturbances at the input and output, respectively, $f_L(t)$ denotes the input fault, acting through L on $x(t)$, $f_M(t)$ denotes the output fault acting through M as an output change $\Delta y(t)$.

For the state estimation error the following equations hold if the disturbances $v(t)$ and $n(t)$ are both zero

$$\dot{\tilde{x}} = [A - HC]\tilde{x}(t) + Lf_L(t) - HMf_M(t) \tag{12}$$

$$e(t) = C\tilde{x}(t) + Mf_M(t) \tag{13}$$

$F_L(t)$ and $f_M(t)$ are additive faults, because they influence $x(t)$ and $e(t)$ by a summation. Essentially, the residual $e(t)$ is the basis for different faults detection methods based on state estimation.

If faults appears as changes ΔA, ΔB or ΔC of the parameters the process behavior becomes

$$\dot{x} = [A + \Delta A]x(t) + [B + \Delta B]u(t) \tag{14}$$

$$y(t) = [C + \Delta C]x(t) \tag{15}$$

and the state estimation error

$$\dot{\tilde{x}} = [A - HC]\tilde{x}(t) + [\Delta A - H\Delta C'] + \Delta Bu(t) \tag{16}$$

$$e(t) = C\tilde{x}(t) + \Delta Cx(t) \tag{17}$$

The faults ΔA, ΔB and ΔC are multiplicative faults, because they influence $x(t)$ and $e(t)$ by a product with the variable $x(t)$ and $u(t)$. In this case the residual changes depends on parameter change as well as input and state variable change. By analyzing the information of faults included in the residuals series, faults can be detected and diagnoses. For more information see chapter two of [4].

The process parameter techniques try to monitor the process directly, based on ph. ical laws whereas the state variable techniques must assume the process parameters as known and try to monitor the signals. Of course, both techniques complement one another. Its disadvantage is that it requires an accurate motor model, one advantage of using a parameter estimation method is that a motor can be modeled accurately. According to the magnitude of coefficient changes, the degree of seriousness of the fault can be estimated. However, it may be difficult to get the relationship between the model parameters θ_j and the physical process coefficients p_i, moreover, when the state estimation method is used, the influence of large modeling error cannot be ignored.

2.3 Signal model

Methods based on signal analysis include, principally, vibration analysis and current analysis. Vibrations analysis is used to detect for example unbalance and bearings faults, whilst current analysis is used to sense rotor faults associated with broken rotor bars and mechanical unbalance [5]. These methods make use of signal models, such a spectrum, correlation function, Fourier transform, etc, to analyze the measured signal. A signal analysis is another source of information if changes of these signals are related to faults, then, features of motor operating conditions can be extracted and used for motor faults detection and diagnosis.

The main advantage of this kind of method is that accurate model is avoided. However, this kind of method only uses the output signals of motor, but no input signals, therefore, the relationship between input and output is not considered. Because of high cost of accurate sensing devices, this method is usually considered useful for large motor only [6].

3 Qualitative approach

Methods based on knowledge include expert system method, fuzzy logic and neural networks. An expert system can be built to detect and diagnose motor faults according to the experience accumulated by an engineer. An experienced engineer can usually detect and diagnose motor faults by observing the motor's operation performance, without knowing or understanding the exact system dynamic, unfortunately, engineer's experience is difficult to describe and be transmitted or automated [7].

Soft computing is considered as an emerging approach to intelligent computing, which parallels the human mind ability to reason and learn in circumstances with uncertainty and imprecision. In contrast with hard computing methods that only deal with precision, certainty and rigor, it is effective in acquiring imprecise

or sub-optimal, but economical and competitive solutions to real world problems. In general, soft computing methods consist of three essential paradigms: neural networks, fuzzy logic and genetic algorithms [8].

3.1 Neural networks-based methods

With artificial neural networks, human expertise can be partially imitated and that neural network can be trained to detect faults based solely on input-output examples without the need of mathematical models. The motivation of employing neural networks for motor fault diagnosis is due to their self-adaptation and nonlinear approximation abilities which can be used to extract the relationships between different input and output variables to diagnose and indicate possible faults. We emphasize that the learning procedure is usually guided by human experts.

The neural network performance depends on the chosen input variables which must be optimal fault indicators moreover, the diagnosis performance can be affected by problems like selection of neural network structure, over training or under training, and slow convergence speed. However, the critical shortcoming of neural network-based motor fault diagnosis is that qualitative and linguistic information from motor operator cannot be directly utilized or embedded in the neural network structure. Additionally, is difficult to interpret the input and output mapping of the training neural network into meaningful fault diagnosis rules [8] [9].

3.2 Fuzzy logic-based method

A system based on fuzzy logic allows the translation of heuristic and linguistic terms into numerical values via fuzzy rules and membership functions to approach the performance of the diagnostic system to the real world.

The major drawback of fuzzy logic is that not provide an exact solution to the problems (the solution are fuzzy in nature) moreover, the design of such system heavily depend on the intuitive experienced acquired from expert operators. The fuzzy membership functions and fuzzy rules cannot be guaranteed to be optimal in any sense. Furthermore, fuzzy logic systems lack the ability of self-learning which is compulsory in some highly demanding real-time fault diagnosis cases. The above drawbacks can be partly overcome by the fusion of neural networks and fuzzy logic techniques [7] [8] [9].

3.3 Diagnosis using neural-fuzzy techniques

To overcome the mentioned problems about neural network system and fuzzy logic system, we can build a hybrid neural/fuzzy system that take advantage of the feature of both technologies while minimizing their drawbacks. The idea behind the fusion of this two technologies is to use the learning ability of neural networks to implement and automate fuzzy systems. A possibility is to use inference systems like Adaptive Network based Fuzzy Inference System (ANFIS) or Fuzzy Adaptive Learning Control/decision Network (FALCON) [1] [10].

3.4 Genetic algorithms-based diagnosis

A Genetic Algorithm is a derivate-free and stochastic global optimization method inspired by the laws of natural selection and genetics, they use the concept of Darwin's theory of evolution, which is based on the rule of the survival of the fittest. These algorithms do not need functional derivate information to search for a set of parameters that minimize (or maximize) a given objective function [11]. Hence, it is attractive to employ genetic algorithms to optimize the parameters and structures of neural networks and fuzzy logic system instead of using back-propagation learning algorithm alone. Since genetic algorithm is only an auxiliary optimization method, it cannot be applied independently in practice. The combination of genetic algorithms with other motor fault diagnosis schemes has demonstrated enhanced performance in global and near-global minimum search. However, optimization with genetic algorithms often involves heavy computing, and is therefore quite time-consuming [8].

4 Comparison and evaluation of methods

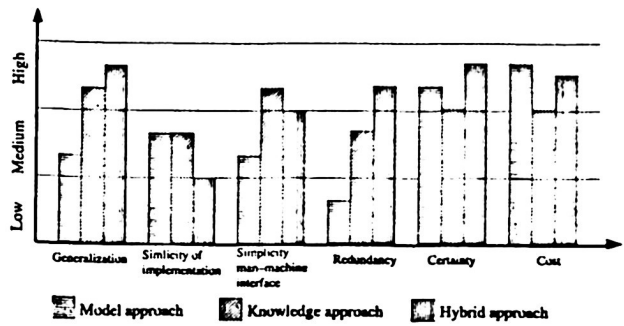


Fig. 1. Comparison of three principal methods

The Fig. 1 shows a comparative graphic of three principal methods: model-based approach, knowledge-based approach and hybrid-based approach with six topics: generalization, simplicity of implementation, simplicity of man-machine interface, redundancy, certainty and cost. Each topic evaluates each method in three ways with adjectives like low, medium and high.

Since self-adaptation, nonlinear capabilities and linguistic rules is common to knowledge-based method, this kind of methods are able to generalize deeper that mathematical model, because the operation of the mathematical model is just for the model for which it was created. Therefore, a mix of both techniques can improve the performance. To implement model-based approach and

knowledge-based approach an engineering designer or experienced personnel with high specialization is required, this can be a drawback in hybrid technique due to the mix of methods. Since is difficult to express or to analyzes raw data of many processes, the model technique is shorter that both others, at the other hand, express knowledge in crisp value is easiest, hybrid technique has intermediate point. Redundancy is the capacity of disposing repeated information, since the mix of methods is logic that hybrid method has the greatest redundancy. Obviously, if exists redundancy in diagnostic the certainty will grow supported by intrinsics certainty of mathematical model, because of that, the hybrid certainty is highest. Cost, according to the experts, is always below in knowledge techniques and, therefore in hybrid techniques than mathematical techniques.

5 Suggested method

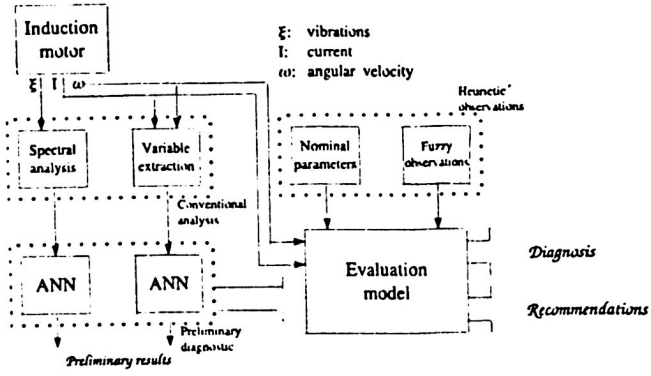


Fig. 2. Suggested scheme of induction machine diagnostic

Our schema, Fig. 2, consist of typical and more commonly used variables, like vibrations, angular velocity and current consumption, moreover, it makes use of nominal parameters, that is, manufacturer specifications and operator observations. This model is divided in two principal parts, the first one takes measurable signals and process them to extract unmeasurable and characteristics parameters by means of classical analysis and simulation models.

Firstly, analysis spectral of vibrations is introduced into an artificial neural network previously trained to detect anomalies in vibration signature. At parallel path, motor's angular velocity and current consumption are utilized by another artificial neural network previously trained with computer simulation data of well state motor and faults state motor, to detect possible faults conditions.

This double path, named preliminary diagnostic, provides diagnostic redundancy that can be used by evaluation model. The evaluation model can make use of preliminary results, direct measurements or both analysis. Moreover, to better inference and decision capability makes use of nominal parameters and fuzzy observations. This last block, could use knowledge base techniques, decision trees or neural/fuzzy inference systems.

At this moment, we are working on motor simulation model and searching the best inference architecture to recreate our model. Results will be published next.

6 Conclusion

In this paper we gave an overview of principal developments in the field of diagnosis of electrical machines. We reviewed model-based techniques, knowledge-based techniques and signal techniques as well as their combinations, advantages and disadvantages were discussed too. Finally we expose briefly an hybrid method that makes use of quantitative and qualitative techniques to overcome their individual limitations and take advantage of particular merits.

References

1. Altug, S., Chow, M.-Y., Trussell, H.J.: Fuzzy inference system implemented on neural architectures for motor fault detection and diagnosis. IEEE Transactions on Industrial Electronics **46** (1999) 1069-1078.
2. Chow, M.-Y.: Guest editorial special section on motor fault detection and diagnosis. IEEE Transactions on Industrial Electronics **47** (2000) 982-983.
3. Isermann, R.: Process fault detection based on modeling and estimation methods. - A Survey. Automatica. **20** (1984) 387-404.
4. Patton, R.J., Frank, P.M., Clark, R.N. (Eds.): Issues of fault diagnosis for dynamic systems. Springer, Great Britain 2000.
5. Benbouzid, M.H.: A review of induction motors signature analysis as a medium for faults detection. IEEE Transactions on Industrial Electronics **47** (2000) 984-993.
6. Liu, X.-Q., Zhang, H.-Y., Liu, J., Yang, J.: Fault detection and diagnosis of permanent-magnet DC motor based on parameter estimation and neural network. IEEE Transactions on Industrial Electronics **47** (2000) 1021-1030.
7. Chow, M.-Y.: Methodologies of using neural networks and fuzzy logic technologies for motor incipient faults detection. World Scientific, Singapore 1997.
8. Quiang, S., Gao, X.Z., Zhuang, X.: State-of-the-art in soft computing-based motor fault diagnosis. Control Applications, 2003. CCA2003. Proceedings of 2003 IEEE Conference on Control Applications. **1** (2003) 1381-1386 vol. 2.
9. Filippeti, F., Franceschini, G., Tassoni, C., Vas, P.: Recent developments of induction motor drives fault diagnosis using AI techniques. IEEE Transactions on Industrial Electronics **47** (2000) 994-1004.
10. Jang, J.-S. R.: ANFIS: adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man and Cybernetics **23** (1993) 665-684.
11. Jamshidi, M., Coelho, L.S., Krohling, R.A., Fleming, J.F.: Robust control system with genetic algorithms. CRC Press, USA 2003.

Generator of unidirectional irregular wave with linear motor and supervised neural control

Luis Pastor Sanchez Fernandez[1], Roberto Herrera Charles[2], Yovany Gonzalez Martinez[1]

[1]Center for Computing Research, National Polytechnic Institute
Av. Juan de Dios Batiz s/n casi esq. Miguel Othon de Mendizabal, Col. Nueva Industrial Vallejo.
CP 07738. Mexico City, Mexico
[2]Research and Development Center of Digital Technology
Tijuana, Baja California, C.P. 22510 Mexico
{lsanchez, yovanygm}@cic.ipn.mx, charles@cic.ipn.mx

Abstract. The coasts are affected mainly by the wave. The wave is a complex phenomenon and very important for the design of the structures that are used in the coastal zones and beaches. This paper presents a novel control system for the generation of unidirectional irregular waves, in project and research laboratories. The system uses digital signal processing, a feedforward neural network and a linear motor, as main elements. The research objective is to obtain a system of easy operation and greater efficiency. With the combined neural control, the transitory response disappears quickly. This behavior is interpreted, in marine hydraulics, as a fast calibration of experiments.

1 Introduction

The wave is a main element for to design coastal and maritime constructions [1], [2]. The phenomenon is very complex and its mathematical representation is difficult [3], [4], [5] [6], [7], [8], [9]. The mathematical models make possible to represent the sea disturbance and to calculate their effects, although in many cases, they need a calibration by means of physical modeling on reduced scale. In complex marine work designs, the physical modeling on reduced scale, is essential. This paper presents a system for the generation of spectral patterns of unidirectional irregular waves, in project and research laboratories. The system uses digital signal processing, a neural network and a linear motor, as main elements.

The research objective is to obtain a system of easy operation and greater efficiency with respect to traditional methods. The transitory response disappears quickly with the combined neural control [10], [11], [12]. This behavior is interpreted, in marine hydraulics, as a fast calibration. In addition, the spectral patterns of the generated wave will have small errors with respect to the spectral patterns of reference.

2 Technical Support and Schemes of Operation

Fig.1 presents the combined neural control for spectral patterns generation of irregular unidirectional wave, where:

S_T: Target spectrum; S_G: Generated spectrum.

The controlled process is a wave channel and the control final element is a generator formed by a linear motor and a paddle device (fig 2). A linear motor [13] is a type of electric motor; an induction motor in which the fixed stator and moving armature are straight and parallel to each other (rather than being circular and one inside the other as in an ordinary induction motor).

The maximum force offered by a linear motor is determined by its construction and is dependent on the position of the slider in the stator. The maximum force curve is symmetric to the centre of the movement range, the so-called Zero Position ZP. If the distance between the end of the stator and the end of the slider is equal to the Zero Position ZP of the motor, the slider is at the centre of its movement range. The Zero Position ZP can be found in the data sheet of each linear motor and is different for each motor. In the SS (shortened stroke) range, the slider's drive magnets are wholely inside the active part of the stator. This provides optimum force generation and a constant maximum force over the whole SS- stroke range. The more the slider moves away from the SS-stroke range, the fewer of its magnets are in the active part of the stator. This means that the maximum and effective forces are reduced linearly as the end of the stroke range S is approached.

A controller PI (proportional-integral) and an inverse neural network (INN) form the combined control.

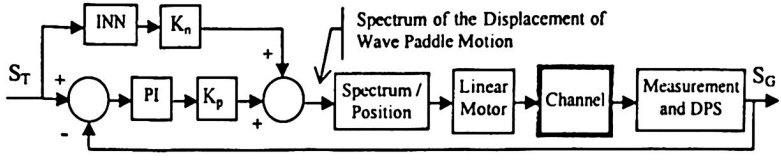


Fig. 1 Combined neural control for spectral patterns generation of irregular unidirectional wave

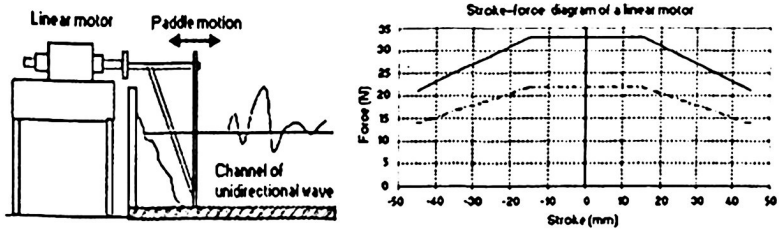


Fig. 2 Unidirectional wave channel and stroke–force diagram of a linear motor.

2.1 Supervised neurocontroller system design

The combined neural control also is called supervised neurocontroller and its general diagram is shown in Figure 3.

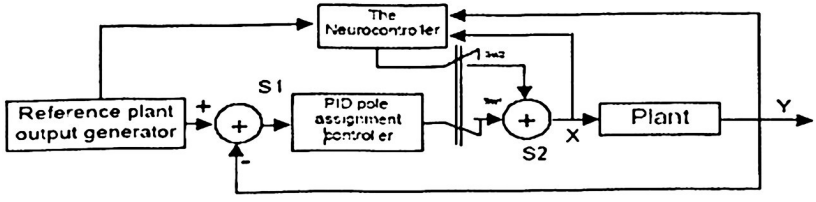


Fig. 3 Supervised neurocontroller system design.

The methodology of neurocontroller system design [15], [16] can be described briefly as learning of a neural network to act as a controller. The control signal generation by using the neurocontroller has the following steps:
1. Determining inverse model parameters which are used in control signal generation U_n which will be the output of this model under the effect of PID controller.
2. Learning of the neurocontroller to generate the control signal according to reference pattern which is obtained from the previous inverse model.
3. Learning the network until the RMS value come to the lowest value as far as possible (to minimize the resultant error).
4. Applying the control signal, generated from the neurocontroller, to the system.
5. Using the output of the system as a feedback input of the neurocontroller to generate the following control signal.
6. Repeating the above steps until the end of patterns.

3 Wave Generation Theory

Eq. (1) is basic for the spectral analysis of a registry of irregular wave in a fixed station, and this defines the function of spectral density $S(f)$ [2].

$$\sum_{f}^{f+df} \frac{1}{2} a_n^2 = S(f)df \tag{1}$$

This equation, nevertheless, contains an infinite number of amplitudes a_n of components of the waves, and this is not applicable to practical calculation. For the practical analysis, a wave registry of N points is acquired, with a constant sampling period: $\eta(\Delta t)$, $\eta(2\Delta t),\ldots\eta(N\Delta t)$. The analysis of harmonics of wave profile $\eta(t)$, the profile can be expressed as the well-known finite Fourier series [2], [18], Eq. (2)

$$\eta(t) = \frac{A_0}{2} + \sum_{k=1}^{N/2-1} (A_k \cos(\frac{2\pi k}{N} t_*) + B_k \sin(\frac{2\pi k}{N} t_*)) + \frac{A_{N/2}}{2} \cos(\pi t_*) \qquad (2)$$

$$t_* = t/\Delta t : t_* = 1,2,3,...N$$

The wave power spectrum can be generated by two general methods: first, in discreet form, with a series of Fourier and the components of power of each one of the harmonics. The second, in the continuous form, with the significant wave height and period and empirical equations of spectrum such as the Mitsuyasu [2], [9], Pierson and Moskowitz JONSWAP [2], etc., for example, the spectra of wind waves fully developed in the open sea, can be approximated by the following standard formulas:

$$S(f) = 0.257 H_{1/3}^2 T_{1/3}^{-4} f^{-5} \exp[-1.03(T_{1/3} f)^{-4}] \qquad (3)$$

$$S(f) = 0.205 H_{1/3}^2 T_{1/3}^{-4} f^{-5} \exp[-0.75(T_{1/3} f)^{-4}] \qquad (4)$$

where $H_{1/3}$: significant wave height; $T_{1/3}$: significant wave period; f : the frequency.

Fig 4. presents an example of sea spectrum. The dash-dot line is the result of fitting Eq. (4) with the values of the significant wave height and period of the record. Although some difference is observed between the actual and standard spectra, partly because of the shallow water effect in the wave record, which was taken at the depth of 11m, the standard spectrum describes the features of the actual spectrum quite well.

The wave generator of mechanical type is more useful and simple and it reproduces better the waveforms. The theory of displacement of the beater (paddle) and the characteristics of the generated waves are studied by several investigators [2], [3], [4], [9]. The desired wave power spectrum is multiplied by the transfer function of the wave generator, well known as equation of efficiency of the paddle. This transfer function is obtained when solving the differential equation for the free boundary conditions (see Eq. 5 and 6)

Piston type:

$$F(f,h) = \frac{H}{2e} = \frac{4\sinh^2(2ph/L)}{4ph/L + \sinh(4ph/L)} \qquad (5)$$

Flap type:

$$F(f,h) = \frac{H}{2e} = (\frac{4\sinh^2(2ph/L)}{4ph/L})(\frac{1 - \cosh(2ph/L) + (2ph/L)\sinh(2ph/L)}{4ph/L + \sinh(4ph/L)}) \qquad (6)$$

where H is the height of the produced wave in the channel; e is the amplitude of wave paddle at the mean water level; f stands for the wave frequency; L wavelength; h the depth of the water in front of the paddle in the channel. The Inverse Fourier Transform is applied to product of Eq. (3) or Eq. (4) and Eq (5) or Eq. (6) to obtain the wave signal in time domain. The Fig. 5 presents the process of the preparation of input signal to an irregular wave generator. The control systems, in general of open loop, need a relatively

great time for the calibration of each experiment in order to generate a wave spectral pattern (target spectrum).

4 FeedForward Neural Network

For identification and control of systems a FNN (FeedForward Neural Network) with three layers is sufficient. A hidden layer is sufficient to identify any continuous function [11], [12], [14], [17]. The input neurons are determined by the number of frequency bands which the spectrum is divided. The tests were made with 128 and 64 inputs. The best results were obtained with 64 (training error and epochs). Another input neuron is added for the different water levels in the channel. The hidden layer uses a sigmoid function. The output layer uses a lineal function. The number of neurons of the output layer is determined by the number of frequency bands, which the generated wave spectrum will be divided (the output neurons were taken equal to the number of input neurons).

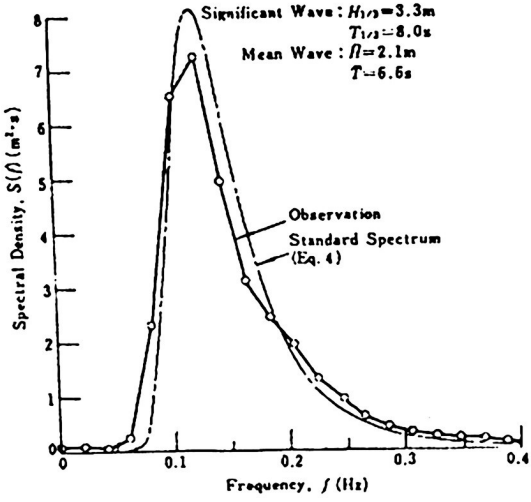


Fig. 4 Example of spectrum of sea waves

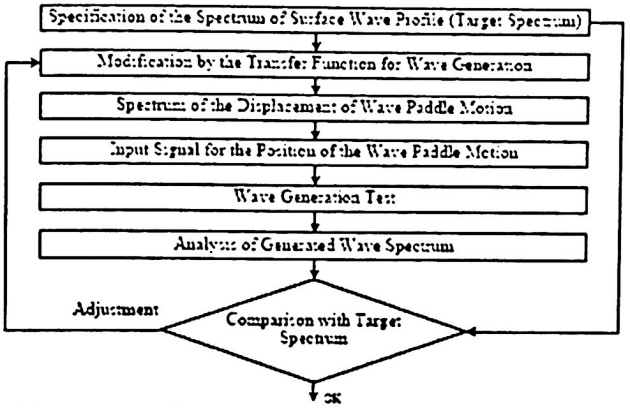


Fig. 5 Process of the preparation of input signal to an irregular wave generator

4.1 Training Patterns

The neural network topology is the following:

Input layer: 65 neurons. Output layer: 64 neurons. Hidden layer: 16 neurons.

Training Patterns:

P^{μ} : Input patterns [f(1), f(2),.....f(nf), h]

T_0^{μ} : Output patterns [f_o(1), f_o(2),.....f_o(nf)]

where f: power spectrum harmonics; h: channel level

Quality factor in the spectrum estimation:

Although the simulation of the sea disturbance is random (pseudorandom). The variability of the spectrum is given by:

$$\hat{S}(f) = S(f)\chi_2^2 \tag{7}$$

The variability of the spectrum is determined by a distribution of chi square with two degrees of freedom, that is the estimation by the periodogram method [2]. In order to reduce the variation, the temporary registry of the wave measurement is divided in a set of M windows.

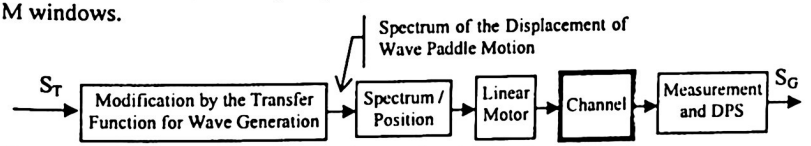


Fig. 6. Acquisition diagram for the neural network training patterns

The training patterns for neural network are obtained with the scheme of Figure 6.

4.1.1 Example of Patterns and Control Neural Performance

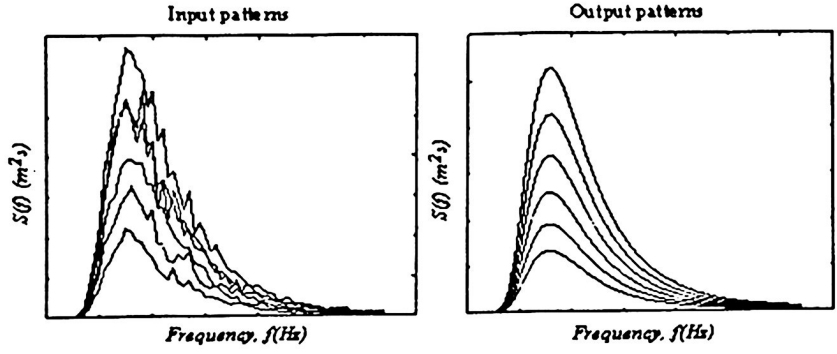


Fig. 7. Example of training patterns. Training performance is $6.97697e^{-10}$, Goal is $1e^{-10}$. Epochs: 25

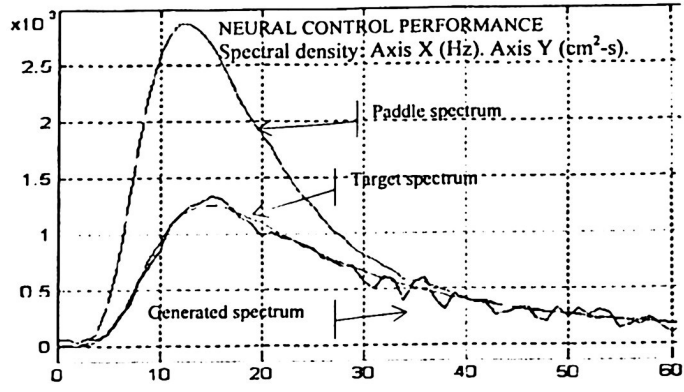


Fig. 8 Example 1 of neural control performance.

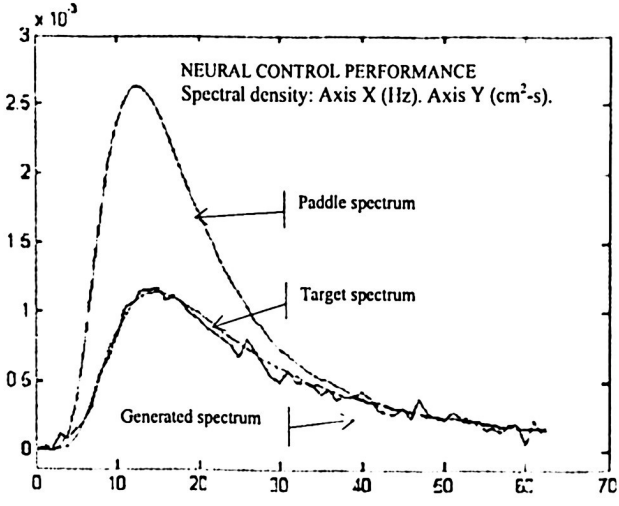


Fig. 9 Example 2 of neural control performance.

Conclusions and Future Work

The coastal and marine constructions are complex and highly expensive. The optimal design requires the physical modeling. The sea phenomena are reproduced in the research laboratories and the designs can be tested. This constructions are "pedraplenes", oil platforms, artificial beaches, protective installations of the coasts, conservation of the ecosystem, etc.

The presented work creates a novel method that uses linear motors and neural networks to generate irregular wave with high exactitude and fast calibration, obtaining itself satisfactory results. The control is made with a distributed architecture, because the linear motor has a system of independent control and it uses digital signal processing because the control variable is a power spectrum.

As future work, self-learning elements will be introduced. These elements will make possible to be creating spectral patterns during the operation of the system and to suggest a new training of the neural network, when the conditions of channel operation have great changes.

References

1. Goda, Y. *"Numerical Experiments on Wave Statistics with Spectral Simulation"* Report 9-3 Port and Harbor Research Institute, 1970. pp-197.
2. Goda, Y. *Random Seas and Design of Maritime Structures.* Scientific World, 2000.
3. Bell, Paul S. Shallow water bathymetry derived from an analysis of X-band marine radar images of wave, *Coastal Engineering. Vol. 37* No.3-4. 1999, pp 513-527.
4. Bretschneider, C.L. 1973. *Deep water wave forecasting curves. En: Shore Protection Manual. U.S. Army* Coastal Engineering Research Center. 36-37.
5. Bretschneider, C.L. 1968. *Significant waves and wave spectrum*, Ocean Industry. Feb. pp 40-46
6. Bullock, G. N. and Morton, G. J. "Performance of a Wedge-Type Absorbing Wave Maker" *Journal Waterw. Port Coastal and Ocean Eng. Div.*, ASCE, Enero, pp 1-17, 1989.
7. Carvahlo, M. M. *"Sea Wave Simulation" Recent Advances in Hidraulic Physical Modelling*, R. Martins, Ed. Kluwer Academic, Dordrecht, 1989. pp. 447-502.
8. Lizano, O., Ocampo F. J. et. al. "Evaluación de modelos numéricos de Tercera Generación para el pronóstico del oleaje en Centroamérica y MéxicoTop". *Meteor. Oceanog.*, 8(1):40-49,2001
9. Mitsuyasu H. et al. Observation of the power spectrum of Ocean Waves using a C over eaf Buoy, *JGR. J Geophysics, 1980.*
10. Hertz, J, Krogh, A, Palmer R. G. *Introduction to the Theory of Neural Computation.* Addison-Wesley Publishing Company, 1994.
11. Psaltis, D. et al. A multi layered neural network controller. *IEEE Control System Magazine, 8(3)*:17-21. Abril 1988.
12. Su C. Y. y Stepanenko, Y. Adaptive control of a class of nonlinear systems with fuzzy logic. *IEEE Trans, Fuzzy Systems Vol 2 No. 4* pp. 285-294. 1994.
13. Alter D. M. and Tsao T. C. Control of linear motors for machines tool feed: design and implementation of optimal feedback control, ASME Journal of Dynamics system, Measurement and Control, Vol. 118, 1996. pp649-656.
14. Barrientos, Antonio et. al. *Control de Sistemas Continuos.* McGraw-Hill, España. 1996.
15. Ashour, M. and Aboshosha, A. Adaptive Neural Control of Npr.Ncrrt, Aea. Atomic Energy Authority. Nasr City, Cairo, Egypt. 1996.
16. Narendrand, K. S. and Parthasarathy, K. "Identification and Control of Dynamical Systems using Neural Networks", IEEE transaction on Neural Networks. vol. 1, No. 1, March. 1990.
17. Liaw, C. and Cheng, S.. Fuzzy two-degrees-of-freedom speed controller for motor drives. *IEEE Transaction on Industrial Electronics, Vol. 42. No 2*, pp. 209-216. 1996.
18. Oppenheim, A.V., Schafer, R.W. and Buck, J.R., *Discrete-Time Signal Processing*, 2nd Edition. Prentice-Hall Int. Ed. 1999.

Using Neural Networks in the Estimation of Consonant Imprecision Ratings.

Carlos A. Ferrer, Anesto del Toro, Eduardo González & Maria E. Hernández-Díaz.

Center for Studies on Electronics and Information Technologies, Central University of Las Villas, Zip code: 54830, Santa Clara, Cuba.
{cferrer, anestodt, moreira, mariae}@uclv.edu.cu

Abstract. This paper deals with the objective measurement of the articulatory imprecision severity in patients with motor speech disorders. Acoustic recordings of repetitive sequences of plosive consonants from 58 patients are used, along with the corresponding subjective ratings of consonant imprecision made by two judges. An estimate of the subjective perception of imprecision per patient is made using energy and sonority measurements of the acoustic signal. Several Neural Networks (NN's) architectures are tested and compared to the linear regression approach to evaluate their prediction abilities. A reduction of more than 25% of the linear regressions error variance is obtained with the use of NN's without a significant increment of computational requirements.

1 Introduction

There is a general agreement that objective measures should be used in the assessment of voice disorders, as a complement to the perceptual judgments of the specialist [1][2][3][4]. Features related to voice quality, pitch perturbations and laryngeal function have been widely addressed, and acoustic correlates of them have been devised. In other symptoms, like those related to prosody, articulation, and nasality fewer results have been accomplished. This paper proposes a method to obtain an estimate of the perceived articulatory imprecision in voiceless plosive consonants. To this end, measures of energy and sonority are used, and several function approximators are tested, including multiple linear regression and feed-forward neural networks.

The structure of this paper is as follows: Section 1 introduces the antecedents of this work, regarding the methods to obtain the measurements of energy and sonority used as inputs to the approximators. In Section 2 the selection of the neural network topologies employed is discussed, together with the experiments conceived to evaluate their effectiveness. In Section 3 the results of the experiments are shown and analyzed, focusing on network performance and generalization capacity. In Section 4 the conclusions of this work are stated, along with recommendations for further research.

The following subsections introduce the reader with the motives and explanation of some approaches that might seem otherwise arbitrary.

1.1 Perceptual Analysis

The term motor speech disorders (MSD) stands for pathologies that cause a disturbance in the control of speech muscular movements as a consequence of a lesion in the central or peripheral nervous system. There are two types of MSD: dysarthria and apraxia. Due to their characteristics [5], it is in dysarthria and not in apraxia where there can be consistent acoustic correlates of the perceived characteristics. Some examples of dysarthrias are Parkinson Disease, Chorea, Amiotropic Lateral Sclerosis.

Several studies [5][6][7] carried out in the late 60's and early 70's were focused on the perceptual characteristics of dysarthric speech. The results of these and other related studies are still considered [4] the basis of clinical differential diagnosis of dysarthria. The methodology created consisted in the realization of three exercises by the patient, and the judgment of 38 perceptual characteristics by a panel of three judges. The exercises proved to convey the maximum clinical information in the minimum time possible [5], and consisted in:

- The phonation of a sustained vowel ("a"), that allows the panel to judge the quality, amplitude, duration and persistency of the fonatory control.
- The repetition of series of syllables using plosive consonants ("Pa" "Ta" and "Ka"), as fast and steady as possible, giving information of rhythm, regularity and duration of every kind of articulatory movements.
- The reading of a standard paragraph, to appraise the way the patient integrates the phonatory, resonatory and prosodic characteristics of contextual speech.

The 38 perceptual characteristics were judged in a 7 point scale, from 0 to 6, with 0 the least perceivable and 6 the most severe level of the feature.

It was shown in these studies that each dysarthria is described by a unique set of groups (clusters) of perceptual characteristics, and that differential diagnosis of dysarthria is possible on the basis of the way the speech sounds.

1.2 Objective Measurement of Consonant Imprecision

This paper is focused on the objective measurement of consonant imprecision, one of the 38 dimensions used in [6][7]. Consonant Imprecision was found significant in all the types of dysarthria reported in the mentioned studies. This fact makes a measurement of consonant imprecision non useful in differential diagnosis, but a good indicator instead of therapy adequacy and for rehabilitation documentation in most dysartrias. To the authors' knowledge, there are no previous objective indexes reported to quantify the degree of consonant imprecision, but only indexes related to the percentage of correctly perceived consonants in a predefined word set [8]. In previous studies by the authors [9][10], it was decided to work with recordings of the Pa/Ta/Ka exercise to obtain an objective measurement of consonant imprecision for several reasons:

- This exercise is part of the standard Mayo Clinic Methodology.
- It is used to appraise the articulatory functioning, where consonant imprecision is included.

- The determination of the position of the consonants can be located more easily than in the paragraph reading.
- All the consonants have similar characteristics (voiceless stops), making easier to devise a procedure for measuring deviations from normality.
- For the latter reason, there is no need to detect the particular consonant of each syllable.

The common characteristics of the consonants present in the Pa/Ta/Ka are the release of a total occlusion in a certain place of the vocal tract, and the lack of sonority previous to that release. A listening to the Pa/Ta/Ka recordings revealed that the possible distortions present were sonorization, nasalization and affrication (see Table 1). The distortions found were the presence of energy prior to the release of the occlusion (turbulent noise in fricatives and periodic sounds in nasal and voiced consonants) and the presence of periodicity before the release in nasal and voiced consonants. Two indexes of abnormality were then established, related to the energy and the level of sonority prior to the release of the constriction.

Table 1. Substitutions found in Pa/Ta/Ka recordings, grouped by the place of the constriction in the vocal tract

	Labial	Palatal	Velar
Unvoiced Stops (Original)	Pa	Ta	Ka
Voiced Stops	Ba	Da	Ga
Nasal (Voiced)	Ma	Na	-
Fricatives (Unvoiced)	Fa	Sa	Ja

The first index, denoted *CIE* (*C*onsonant *I*mprecision by *E*nergy), is a ratio of the areas of the energy envelope before (*Ab*) and after (*Aa*) the release of the constriction (see Fig. 1) as defined in equation (1).

$$CIE = \frac{A_b}{A_a} = \frac{\sum_{T_{on}-T_r}^{T_{on}-T_r} E(t)}{\sum_{T_{on}+T_r}^{T_{on}+T_r+T_a} E(t)} \qquad (1)$$

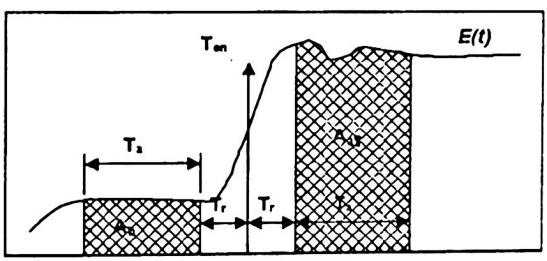


Fig. 1. Graphical representation of the variables used in the calculation of *CIE*.

The instant of release is denoted T_{on} in (1), and the values of the time intervals T_r and T_a were 10 and 20 milliseconds (*ms*), respectively. The average of the *CIE* values

for all the syllables found in the patient's recording (CIE_m) was used as the objective index of consonant imprecision regarding inadequate release of the occlusion.

The second index was calculated using the autocorrelation function of a segment of the speech signal previous to the T_{on}. The value of the maximum peak of the autocorrelation in the range of possible values of fundamental period of voice (2-20 *ms*, 50-500 Hz) is found on each syllable. The average of these maximums is denoted CIS_m (*C*onsonant *I*mprecision by *S*onority *m*ean) and is calculated for each patient's recording. The speech segment's end is set 10 *ms* before T_{on}, and its length is 40 *ms*.

For the calculation of both indexes, the determination of T_{on} is required and crucial. To this end, a syllable detector was devised [9], based on the analysis of the energy envelope maximums. The maximums that do not satisfy some heuristics that must be met to be perceived as a separate syllable are eliminated from the list of syllable candidates, and the remaining maximums are considered syllable centers. The heuristics used were:

- Amplitude greater than 20 times the minimum value of the energy envelope.
- Separation between maximums greater than 100 *ms* (when two maximums are closer than this value, the one with the lower amplitude is suppressed).
- Presence of a minimum of less than 75% of the maximum's amplitude between itself and the previous and posterior maximums.
- Separation of the mentioned minimums of more than 50 *ms*.

Once the syllables are detected, the instant of release is determined as the point in the energy envelope with the greatest positive slope in the segment between the syllable's center and the previous one.

This method of syllable position determination showed an 89% of correct detection in a set of 3750 syllables of dysarthric patients [9].

With the obtained values of CIE_m and CIS_m and the subjective evaluations of two judges, the linear regression of the averaged subjective judgments (SJ_m) was obtained for both indexes and their combination. The results of the correlations and error variances of the three regressions to the original subjective judgments are shown in Table 2. The correlation between judges was 0.75.

Table 2. Correlations and error varainces of the linear regressions of CIE_m, CIS_m and their combination $CIES_{lr}$

	CIE_m	CIS_m	$CIES_{lr}$
Correlation with SJ_m	0.5866	0.5642	0.6711
Error Variance	2.3527	2.4451	1.9712

1.3 Hypotheses and Objectives

Regarding consonant imprecision, the analysis of the location in the CIS_m/CIE_m plane of the four types of consonants involved suggests that a linear relationship can be a gross estimate (See Fig. 2). It is considered by the authors that a linear regression of the indexes devised is not a precise approximation to the way human ears perceive consonant imprecision. If the subjective judgment is zero for "Normal" and six for the other three consonants, it is evident that a nonlinear approximation should outperform the linear regression approach.

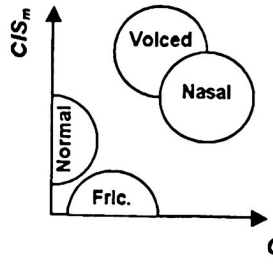


Fig. 2. Representation of the consonants in Table I on the plane CIS_m/CIE_m.

The objective of this paper is to explore the improvement in performance (i.e. reduction in error variance or increment in correlation) a nonlinear "approximator" (the term "predictor" will also be used in this paper) can achieve with no significant increment in model complexity compared to the multiple linear regression.

2 Non-Linear Models and Experiments.

Different non-linear prediction models, based on feed-forward neural networks (FNNs), were selected and used for testing purposes. The models included various topologies of multilayer perceptrons (MLP) and radial basis networks (RBN) [11]. These architectures had been widely used and their abilities as function approximators had been demonstrated in different applications [12].

2.1 Topology Selection.

The maximum number of neurons for the FNNs was set to 5. This was decided to avoid a significant increment in the predictor's model complexity compared to the linear regression approach.

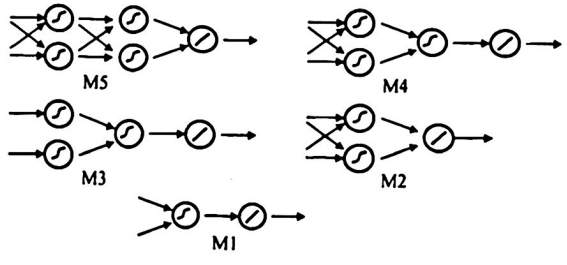


Fig. 3. MLPs' topologies tested.

For the MLP architecture the topologies used are shown in Figure 3, in descending order of complexity. From now on they will be referred as M5, M4, M3, M2 and M1, respectively. A linear positive transfer function was employed in the output layer while the hyperbolic tangent sigmoid was used for the input and hidden layer neurons [11]. The RBF networks evaluated ranged from 2 to 5 neurons, and will be referred correspondingly as R2, R3, R4 and R5.

2.2 Training Experiments

The data used for the evaluation of the predictors were the obtained CIE_m and CIS_m indexes for 58 recordings of the Pa/Ta/Ka exercise, along with the corresponding subjective ratings of consonant imprecision taken from [10]. In all cases the objective indexes were employed as inputs and the subjective ratings were used as output targets.

The generalization ability of each topology was tested (experiment "A"), to have a better appraisal of the tradeoff with the predictor's performance. To this end, the whole data set was randomly divided in two halves, a training and a control set. Each MLP was randomly initialized several (15) times to reduce the chance of convergence to local minimums, a common failure of the *backpropagation* learning function used, and the whole training set was fed to the network 300 times, when the training process was stopped. The best resulting MLP among the 15 initializations was chosen as the optimal predictor for the training set. The RBF networks were obtained for the same training set, with different spreads of the radial basis function. The best resulting RBF among the different spreads was considered the optimal approximator within each topology. This halving procedure was repeated 60 times, and the averaged results for training and control sets for both MLP and RBF networks were obtained. Each network was also trained for the whole data set (experiment "B") to obtain their global performances, using the same random initialization procedure for the MLPs.

3 Results and Discussion

The performances of the networks for the total data set and the training/control experiment are shown in Tables 3 (RBFs) and 4 (MLPs and linear regression). Table 4 includes the number of outliers obtained in experiment "A" for the 60 control sets. An error variance greater than ten times the median of the results with the control sets was considered as outlier. The mean error variance of the control sets results with the outliers removed is also shown in Table 4 (row denoted "C.S. w/o Outl."), since the one including the outliers is practically meaningless. Another extra row in Table 4 is the value of the Pearson's correlation coefficient obtained between the subjective ratings and the outputs of the MLPs predictors in experiment "B".

Table 3. Mean error variances of the 60 training/control experiments and total error variance for the RBF networks

		R2	*R3*	*R4*	*R5*
A)	Training Set	2.2354	1.7110	1.6478	1.5734
	Control Set	2.5247	2.2109	2.2855	2.7619
	Average	2.3801	1.9609	1.9667	2.1677
B)	Whole Set	2.4794	2.211	1.7531	1.7416

Table 4. Mean error variances of the 60 training/control experiments and total error variance for the MLP networks and the Linear Regression approach.

		M5	*M4*	*M3*	*M2*	*M1*	*LR*
A)	Training Set	0.9893	1.1126	1.3870	1.2427	1.7779	1.7615
	Control Set	*66.643*	*20.418*	*2.4949*	*2469.4*	*2.4995*	*2.2449*
	# of Outliers	*8*	*2*	*0*	*1*	*0*	*0*
	C.S. w/o Outl.	4.5264	3.2534	2.4949	2.8237	2.4995	2.2449
B)	Average	2.4977	2.1649	1.9404	2.0266	2.1387	2.0052
	Whole Set	1.3940	1.5136	1.6938	1.5239	1.9163	1.9712
	R	0.7819	0.7603	0.7265	0.7584	0.6824	0.6711

From these results it is apparent than RBFs of up to 5 neurons are not well suited to be good approximators for this data set. The best overall performance (obtained for R5) only surpasses the linear regression and M1 results. Besides, there is no significant increment in performance from R4 to R5, while the generalization ability is actually deteriorated. The lower the average performance of a topology in experiment "A" the higher its generalization ability.

The MLP architecture showed a better performance, although some topologies presented outliers in the results. There is an exception in the logical increment sequence of performance in experiment B) going from topology M1 to M5. It is in the case of M2 and M3, where the latter has one neuron more than the former, and in spite of this, M3 has a lower performance and a better generalization capacity. This is due to the greater number of connections used by M2, allowing the formation of more complex surfaces than M3. It is precisely this pair of topologies the ones with better results taking into account global performance (in terms of error variance), generalization capability and absence of outliers. The M2 topology has an error variance comparable to the ones of M4 and M5, together with the second best generalization ability, and only one outlier result, while M3 has no outlier, presents the best generalization ability, but shows a lower global performance. Even though, the correlation coefficient for the M3 predictor is higher than 0.707, so this model explains more than 50% of the error variance, result that is considered acceptable in the literature [13][14] when dealing with subjective judgments. The lowest error variance obtained (1.39) represents a reduction of almost 30% of the obtained with the linear regression (1.97). The values of correlation obtained (0.72~0.78) are comparable to the 0.75 between judges.

4 Conclusions.

From the two architectures of nonlinear approximators tested, the RBF networks did not show a good performance or generalization ability. The MLP predictors presented the best results, with a reduction of more than 25% of the error variance of the original linear regression approach in the 5 neurons topology. The best results considering all the factors evaluated, were obtained for the M2 and M3 variants, with 3 and 4 neurons, respectively. This represents no significant increment in the approximator model complexity compared to the linear regression. The values of correlations found are similar to the ones obtained between the subjective judgments of the specialists.

Further research is needed to obtain a third index in order to increase predictor's performance. Specifically, an index that could make a better separation of the fricatives from the normal plosive consonants (see Fig. 2.) would be desired.

References

[1] Baken, R.J. Clinical Measurement of Speech and Voice. Singular Publishing Group Inc. San Diego. (1996)

[2] Cannito, M.P., Yorkston, K.M. & Beukelman, D.R. Neuromotor speech disorders. Nature, Assessment, and Management. Paul H. Brookes Publishing Co. (1998).

[3] Kent, R.D. & Ball, M.J. Voice Quality Measurement. Singular Publishing Group, Inc. (2000).

[4] Yorkston, K.M., Beukelman, D.R. & Bell K.R. Clinical Management of Dysarthric Speakers. Austin, TX: Pro-Ed. (1988).

[5] Darley, F.L.; Aronson, A.E. & Brown, J.R. Motor Speech Disorders. Philadelphia. Saunders. (1975).

[6] Darley, F.L.; Aronson, A.E. & Brown, J.R. Clusters of deviant speech dimensions in the dysarthria. Journal of Speech & Hearing Research. 12, pp 462-496. (1969).

[7] Darley, F.L.; Aronson, A.E. & Brown, J.R. Differential diagnostic patterns of dysarthria. Journal of Speech & Hearing Research. 12, pp 246-269. (1969).

[8] House, A. S. et al, Articulation-Testing Method: Consonantal Differentiation with a Closed-Response Set J. Acoust. Soc. Am. Vol 37 (1), pp 158-166. (1965).

[9] Ferrer, C.A.; Hernández, M.E. & González, E. Isolated Syllable Position Detector in Recordings of Patients With Motor Speech Disorders Using Speech Processing Techniques. Proceedings of the TELEC'02 International Conference, Santiago de Cuba, ISBN 84-8138-506-9, July. (2002).

[10] Ferrer, C.A. & Hernández, M.E. A Measure of Articulatory Imprecision in Dysarthric Patients Recordings. Proceedings of the VIII International Congress on Social Communication. Santiago de Cuba, ISBN 84-8138-506-9, January. (2003).

[11] H. Demuh, Neural Network Design, PWS Publishing Company. (1996).

[12] B. Widrow, E. Rumelhart, A. Lehr, Neural Networks: Applications in industry, business and science, Communication of ACM, 37 (3) 93-105. (1994).

[13] Rabinov C. R., Kreiman J. Comparing Reliability of Perceptual Ratings of Roughness and acoustic Measures of Jitter. Journal of Speech & Hearing Research. 38, pp 26-32. (1995).

[14] Fukazawa, T.; El-Assuooty, A. & Honjo, I. A new index for evaluation of the turbulent noise in pathological voice. Journal of the Acoustical Society of America. Vol. 83, No 3. pp 1189-1193. March. (1988).

Author Index

Índice de Autor

Volume Reviewing Committee

Comité editorial del Volumen

FE DE ERRATAS

PORTADA DICE	DEBE DECIR
Vol. 14	Vol. 15

Advances on Digital System Design
Impreso en los Talleres Gráficos
de la Dirección de Publicaciones del
Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
Septiembre de 2005
Edición 500 ejemplares.

ISSN: 1665-9899

INSTITUTO POLITÉCNICO NACIONAL