BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iaşi
Tomul LVII (LXI), Fasc. 3, 2011
Secţia
ELECTROTEHNICĂ. ENERGETICĂ. ELECTRONICĂ

# ADAPTIVE FILTERING REAL-TIME ALGORITHM USING PXI PLATFORM

BY

**IOAN TUDOSĂ*, CRISTIAN FOŞALĂU and CRISTIAN ZET**

"Gheorghe Asachi" Technical University of Iaşi,
Faculty of Electrical Engineering, Energetics
and Applied Informatics

Received, June 12, 2011
Accepted for publication: August 16, 2011

**Abstract.** The implementation and performance evaluation of two adaptive algorithms used for filtering of noisy signals are presented. The algorithms have been implemented using LabVIEW as main programming language and the Adaptive Filter module, provided by National Instruments (NI). The testing platform consists of a Real-Time PXI system and a remote PC used for design development. Least mean square (LMS) and Normalized LMS (NLMS) algorithms were developed and tested on Real-Time PXI. Comparisons with a classical finite impulse response (FIR) filter are presented by evaluating the signals-to-noises ratio (SNR) as a performance. The obtained result concludes that the adaptive filters implemented on real-time operation systems (RTOS) are more efficient in terms of filtering efficiency than the classical methods.

**Key words:** adaptive filter; signal processing; LabVIEW Real-Time; PXI platform.

## 1. Introduction

The usefulness of digital filters nowadays is very important to improve the quality of measurement signals. Implementation of digital filters on digital

* Corresponding author: *e-mail*: itudosa@ee.tuiasi.ro

signal processors (DSPs) can be performed using high level programming languages, like C/C++. By using a high level graphical programming language as LabVIEW or MATLAB, the implementation of such digital filters can become even easier (Haykin, 2000). Classical digital filters are characterized by constant coefficients (fixed frequency response) and are useful when signals or noises are known and invariant in time. Otherwise, when signal or noise are changing in time or their characteristics are unknown, classical filters cannot achieve good results in filtering (Tudosă *et al*., 2010). The adaptive filtering idea is to adapt the frequency response appropriately to the wanted characteristics. One may say that an adaptive filter is characterized by two important parameters (Haykin, 2000; Sayed, 2003; Widrow & Stearns, 1985). First parameter refers to the filter structure whilst the second one concerns the algorithm conceived for updating the coefficients.

This paper proposes the use possibility of real-time platforms in signal filtering, especially in adaptive signal processing.

In the next sections, the experimental platform structure used for signal denoising is presented, followed by a description of the LabVIEW adaptive filtering program using the LMS and NLMS algorithms. Next, the obtained results during laboratory experiments are presented and discussed. Finally, some conclusions about the efficiency of adaptive filtering operating in RTOS with respect to the classical filters are drawn.

## 2. Experimental Setup

NI company develops hardware and software for dedicated RT equipment which are used to execute applications in real-time. Our application is developed on Windows operating system and run in the LabVIEW Real-Time engine (Tudosă *et al*., 2009). After design, the LabVIEW applications are loaded on the dedicated hardware to be executed in real time. They are running independently as long as the hardware is power supplied. Thus, using this graphical programming environment, complex applications can be developed at low cost and high reliability.

The experimental platform consists of a NI-PXI platform, a network switch and a personal computer, as presented in Fig. 1. The RTOSs are used in particular applications that require deterministic control loops, as well as automation systems or systems that incorporate adaptive control structures and signal filtering. Owing to the deterministic nature of an application, it must fully execute a given operation within a time *a priori* established. The deterministic performances of the application are obtained using LabVIEW Real-Time module engine, which allows critical operations to have a high priority in execution with respect to other operations that are running at a certain time.

The hardware structures dedicated to real-time operations are running within a LabVIEW engine under LabVIEW RTOS, being integrated into a PCI Extension for Instrumentation (PXI) system, as depicted in Fig. 1. When a virtual instrument (VI) built to run on a real-time platform is executing, the deterministic behavior of the application is provided by LabVIEW Real-Time engine through a combination of methods for ordering operations like round-robin and preemptive methods.

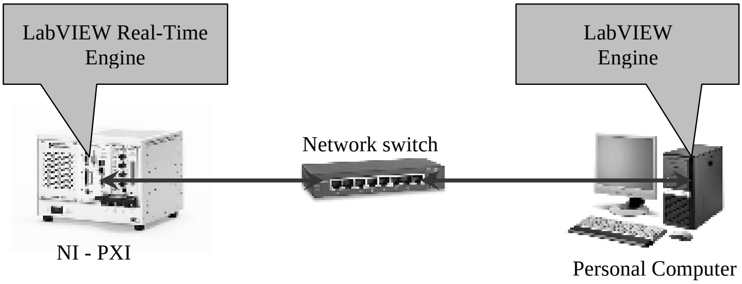



Fig. 1 – Laboratory setup for experimenting the RT adaptive filtering algorithms.

## 3. Algorithm Implementation in LabVIEW RT

The implemented algorithm in LabVIEW is presented in Fig. 2. The block diagram consists of two main deterministic while loops. In the first Timed Loop the data acquisition and filtering procedures are implemented. From Fig. 2- (1) one may observe the initialization portion of code of data acquisition and parameters setup. Into the deterministic while loop is also implemented a flat sequence structure for ordering the code execution. The data acquisitions from two analog input channels are also processed in the first part of the diagram.

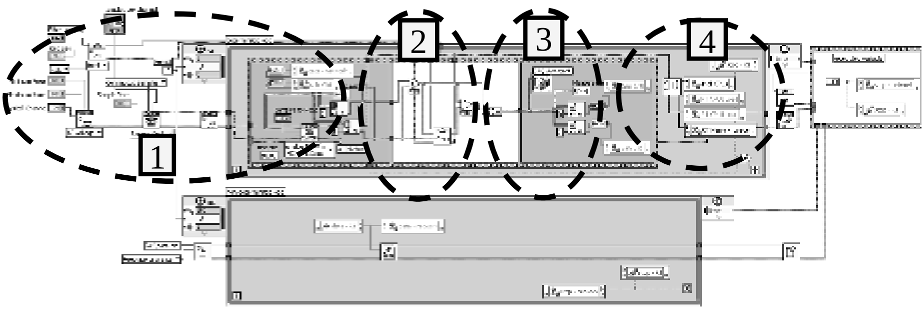



Fig. 2 – Block diagram of the implemented RT application running on the NI-PXI platform.

The first channel acquires the noisy signal and the second channel acquires the reference noise. In the second part of the flat sequence structure is implemented the filter code (Fig. 2-(2)). In the third part of the LabVIEW source code, the signals generation of filtered data is implemented. In the section 4 (Fig. 2) is implemented an algorithm that monitors the parameters from the central processor unit (CPU) followed by the transmission of processed data to the second while loop. The second Timed Loop carries out the data transfer over the network using a TCP-IP protocol and check if the remote computer requires the shutdown of PXI platform. In order to visualize the processed data inside the PXI platform, a LabVIEW VI that acquires the data transmitted over the TCP-IP protocol has been used. In Fig. 3 a screenshot of front panel of the application running on the remote PC under LabVIEW engine is shown.

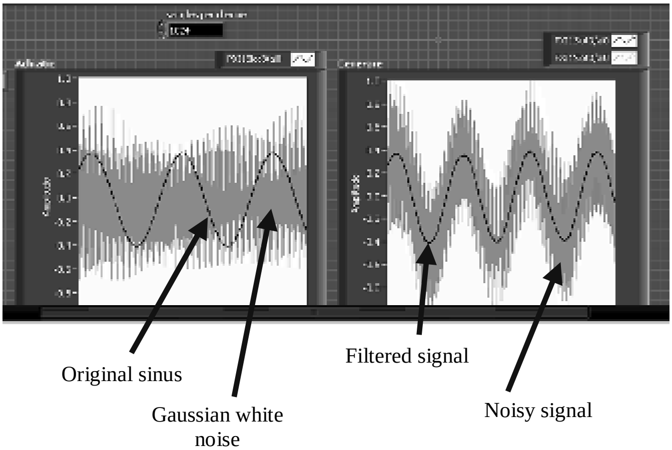



Fig. 3 – Front panel of the remote LabVIEW application.

## 4. Experimental Results

In order to test the efficiency of the adaptive filtering operating in RTOS, two adaptive algorithms have been used: LMS and NLMS (Haykin, 2000). The test signals as well as the Gaussian white noises were generated using a data acquisition board type NI-PXI-6251. Table 1 presents the SNR values calculated for the signal output delivered by a classical FIR filter as well

**Table 1**

*Experimental SNR Measured at the Filter Output*

| Filter order | SNR FIR dB | SNR NLMS dB | SNR LMS dB |
|---|---|---|---|
| 25 | 16 | 16.4 | 16.1 |
| 64 | 20 | 18.9 | 17.5 |
| 128 | 22 | 24.3 | 21.15 |
| 256 | 23 | 35.45 | 31.92 |
| 384 | 24.5 | 38 | 35 |
| 512 | 26 | 43 | 40 |

as by the two adaptive filters designed around the LMS and NLMS adaptive algorithms, all having as input a noisy sinusoid of very low SNR (SNR = 4.46

dB). The results are classified according to the filter length. The three characteristics traced in Fig. 4 present the dependence of the output signal SNR *vs*. the filter length, for the above filters.

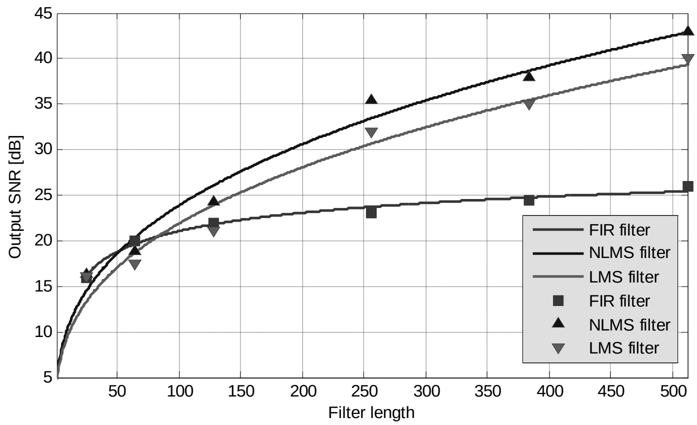



Fig. 4 – Comparisons between output SNR for three different filters: classical FIR filter and LMS and NLMS adaptive filters.

The input signal SNR measured *vs.* the noise amplitude superimposed over the useful sinusoidal signal is presented in Fig 5. It can be seen that for the high noise amplitude, the SNR is quite low (4.7 dB) whereas for low levels of overlapped noise, the SNR is much higher (22.4 dB). The execution times of the

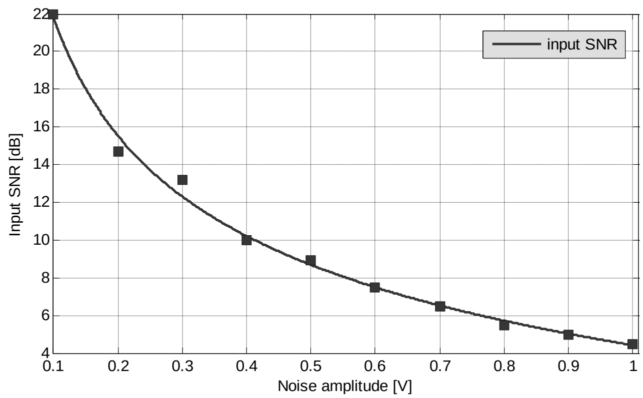



Fig. 5 – Input SNR *vs.* noise amplitude of the tested signals.

adaptive algorithms were also measured. In Table 2 is presented the measured values on RTOS for the LMS and NLMS algorithms. In Fig. 6 are presented the execution times for LMS and NLMS algorithms at high filter lengths and in Fig. 7 a comparison has been performed between execution times of LMS and NLMS algorithms at high filter lengths.

There were also measured the execution times for low filter lengths. The results are presented in Fig. 8. A comparison between these execution times of LMS and NLMS algorithms at low filter lengths are presented in Fig. 9. One may observe that the NLMS algorithm performs better filtering operation than the LMS algorithm but requires more time for execution.

**Table 2**

*Experimental Execution Time for the Adaptive Filters.*

| Experimental execution time measured for LMS and NLMS algorithms running on RTOS | | |
|---|---|---|
| Filter length | Measured time for LMS, [us] | Measured time for NLMS, [us] |
| 25 | 210.32 | 228.32 |
| 64 | 457.64 | 479.61 |
| 128 | 874.54 | 905.62 |
| 256 | 1660.52 | 1699.9 |
| 512 | 3359.21 | 3377.22 |
| 1024 | 6588.32 | 6640.13 |

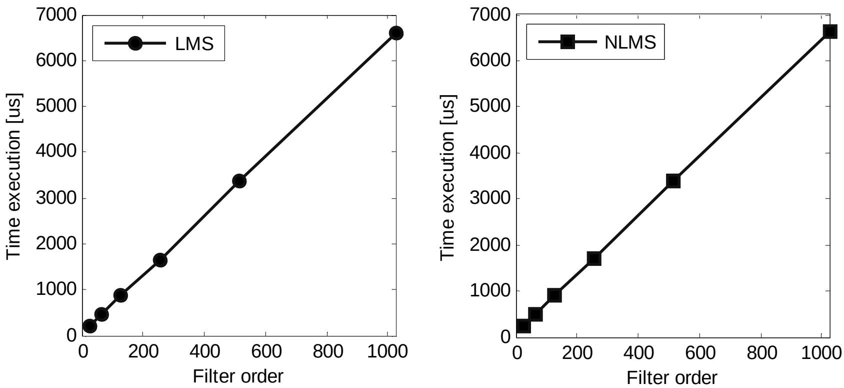



Fig. 6 – Execution time for LMS and NLMS algorithms at high filter lengths.

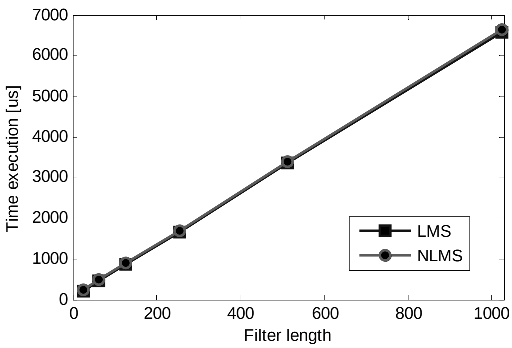
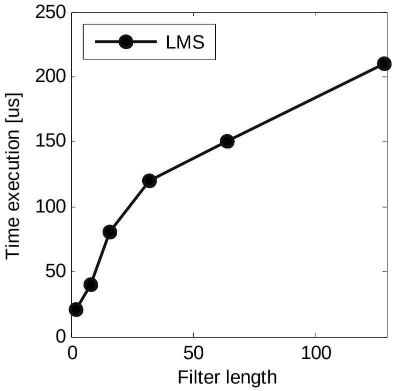



Fig. 7 – Comparisons between execution times of LMS and NLMS algorithms at high filter lengths.



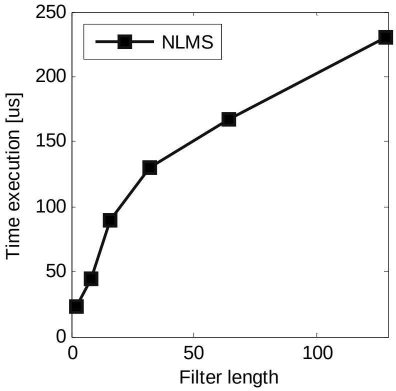




Fig. 8 –Execution time for LMS and NLMS algorithms at low filter lengths.

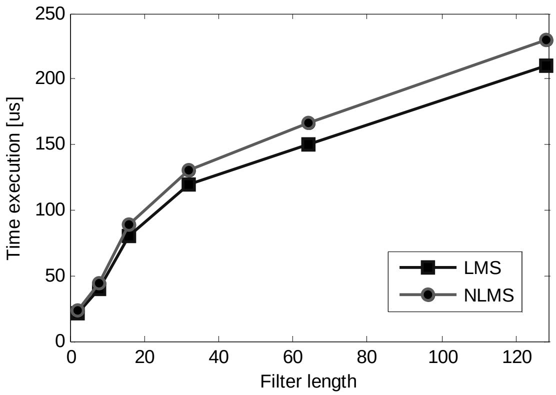



Fig. 9 – Comparison between execution times of LMS and NLMS algorithms at low filter lengths.

## 5. Conclusions

This paper presents an original point of view regarding the LMS and NLMS adaptive filters implemented on a PXI platform. The obtained results prove the efficiency of adaptive filters in comparison with the classical FIR filters.

**Acknowledgments.** This research was realized with the support of BRAIN "Doctoral Scholarships as an Investment in Intelligence" project, financed by the European Social Found and Romanian Government.

## REFERENCES

Haykin S., *Adaptive Filter Theory*. Third Edition, Prentice Hall, NY, 2000.
Sayed A.H., *Fundamentals of Adaptive Filtering*. J. Wiley, New Jersey, 2003.

Tudosă I., Foşalău C., Petrişor D., Măriuţ F., *Studies on the Effectiveness of the LMS Algorithm Used in Signal Filtering*. Proc. of the 6th Internat. Conf. on Electr. a. Power Engng., EPE2010, October 28-30, 2010, Iaşi, Romania, **II**, 357-360.

Tudosă I., Foşalău C., Zet C., Costineanu D., Petrişor D., *Experimental Test Bench for DSP Applications*. Proc. of the 5th Internat. Conf. on Metrol. & Measur. Syst., METSIM2009, November 5-6, 2009, Bucharest, Romania.

Widrow B., Stearns D.S., *Adaptive Signal Processing*. Prentice Hall, NY, 1985.

* * * http://www.ni.com/realtime/software.htm.

* * * http://www.ni.com/labview/

## PRELUCRAREA ADAPTIVĂ A SEMNALELOR DE MĂSURĂ ÎN TIMP REAL UTILIZÂND PLATFORMA PXI

(Rezumat)

Se propune implementarea şi evaluarea performanţelor a două filtre adaptive utilizate pentru filtrarea semnalelor zgomotoase. Algoritmii au fost implementaţi utilizând mediul LabVIEW şi modulul Adaptive Filter. Platforma de test constă dintr-un sistem PXI de timp real produs de către NI şi un calculator remote utilizat pentru comandă şi vizualizare. Algoritmii LMS şi NLMS au fost implementaţi pe platforma PXI. S-au realizat comparaţii cu un filtru FIR clasic şi s-au trasat caracteristici experimentale în funcţie de SNR la ieşirea filtrelor.