AQTR 2004 (THETA 14)
2004 IEEE-TTTC - International Conference on Automation, Quality and Testing, Robotics
May 13 – 15, 2004, Cluj-Napoca, Romania

Implementing an Adaptive Noise Cancelling System Using TMS320C31 DSP

Nicoleta Stroia*, Daniel Moga*, Petru Dobra*,
Radu A. Munteanu*, Mihai R. Dumitrean*

Technical University of Cluj Napoca, Department of Automation,
Baritiu 26, Cluj Napoca, Romania
n_stroia@yahoo.com , daniel.moga@aut.utcluj.ro , petru.dobra@aut.utcluj.ro ,
radu.a.munteanu@mas.utcluj.ro , mihai.dumitrean@tedelco.ro

Abstract: In signal processing and control applications where the signals and transfer functions involved are time invariant and known at design time, designing fixed filters to achieve the desired goal is sufficient. In many applications, however, signals, transfer functions and the environment in which the system operates are time varying. In such situations, an adaptive filter could be used to achieve the desired system function. This paper presents a detailed discussion on the implementation of adaptive filters, along with the results obtained by implementing an adaptive noise cancelling system based on LMS and RLS algorithms on a PC-based TI TMS320C31 floating-point digital signal processor (DSP) system.

Keywords: adaptive filter, DSP, LMS, RLS, adaptive noise cancelling

1. INTRODUCTION

Adaptive filtering is needed to deal with uncertain changes in dynamical systems and became an important tool in the area of digital signal processing and adaptive control in applications such as: system identification, echo cancellation, channel equalization, adaptive linear prediction, adaptive interference cancelling.

An adaptive filter is a filter containing coefficients that are updated by an adaptive algorithm in order to optimize the filter response to a desired performance criterion. Adaptive filters can be used in various applications with different input and output configurations. In many applications, requiring real-time operations, an adaptive filter implementation based on a programmable DSP has many advantages. Not only that power, space, and manufacturing requirements are greatly reduced, but also programmability provides flexibility for easy system upgrade and software improvement [6].

Adaptive filtering and digital signal processing problem has been intens studied (see [1], [2], [3], [4]). The problem of adaptive noise cancelling is one of the simplest and efficient configuration of adaptive filters, and has many applications in practice (see [1], [6], [7], [8]).

In this paper, two of the most commonly used adaptive algorithms were considered: stochastic gradient algorithm LMS (Least Mean Square), and recursive algorithm RLS (Recursive Least Squares). Only finite impulse response transversal filters were considered. FIR filters present the advantage of simplicity and yield to satisfactory performance in many applications, reason for which they have been intense studied and used in practice. In order to experiment adaptive filters efficiency, an adaptive noise cancelling system was implemented in Matlab (Simulink), and then the generated code (obtained by using the Real-Time Workshop), was loaded on the TI TMS320C31 floating-point DSP.

2. DESIGN OF ADAPTIVE FILTERS

A general block diagram of an adaptive filter is shown in figure 1. The adaptive filter consists of two distinct parts: the filter part and the update part. The function of the filter part is to calculate the convolution of the input signal $x(n)$ and the filter coefficients, resulting in the filter output, $y(n)$. The set of filter coefficients are continuously adjusted by the update part, so that the output becomes as close as possible to a reference signal, $r(n)$. The adaptive algorithm changes the filter coefficients in order to minimize a certain function of the error signal, $e(n)$, defined as the difference between the reference signal, $r(n)$, and the output signal, $y(n)$.

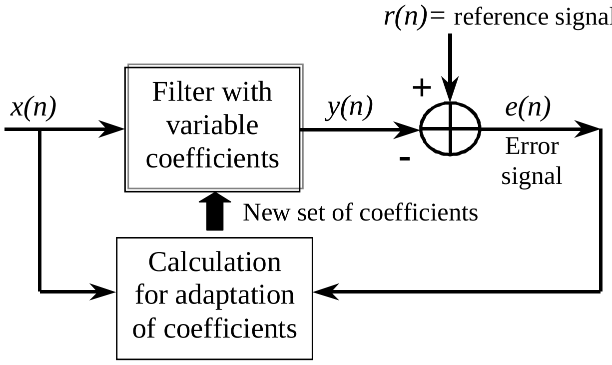

Figure 1: General block diagram of an adaptive filter

2.1. LMS Adaptive Algorithm

The Least Mean Square (LMS) algorithm tries to minimize the mean square value of the error signal:

$$J = E\{e^2(n)\} = E\{(r(n) - \sum_{k=0}^{M-1} w_k(n) \cdot x(n-k))^2\} \quad (1)$$

It is a member of stochastic gradient algorithms class. The estimation of the statistical mean operator is done by considering its current value: $E\{f(k)\} = f(k)$. It is based on the steepest descendent (SD) method (*iterative* method) for finding the optimal value. The adaptive filter coefficients are updated according to the following equation:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \cdot \mathbf{x}(k) \cdot e(k) \quad (2)$$

where μ is the step size which is important for the convergence speed of the algorithm. It has been shown that w_i will converge if the adaptation step, μ, satisfies the condition:

$$0 < \mu < \frac{2}{\lambda_{\max}} \quad (3)$$

where λ_{max} is the maximum eigenvalue of $\mathbf{R}_x$ (the autocorrelation matrix of $\mathbf{x}$).

For each sample, the LMS algorithm, performs the following operations:

- filters the input signal: $y(k) = \mathbf{w}^T(k) \cdot \mathbf{x}(k)$ (4)
- computes the current sample of the error signal: $e(k) = r(k) - y(k)$ (5)
- updates weight vector: $\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \cdot \mathbf{x}(k) \cdot e(k)$ (6)

The initialization is done by choosing an initial value for the weight vector (usually $\mathbf{w}(0) = \mathbf{0}$) and for the adaptation step, μ.

2.2. RLS Adaptive Algorithm

The cost function for RLS:

$$J(n) = \sum_{i=1}^{n} \lambda^{n-i} \, |e(i)|^2 \tag{7}$$

where λ is the forgetting factor $0<\lambda<1$.

At each sample, RLS adaptive algorithm, performs the following operations:

- computes the error signal: $\alpha(n) = d(n) - \mathbf{w}^T(n-1)\mathbf{x}(n)$, (8)
- computes Kalman gain: $\mathbf{k}(n) = \frac{1}{\lambda} \frac{\mathbf{P}(n-1)\mathbf{x}(n)}{1 + \frac{1}{\lambda}\mathbf{x}^T(n)\mathbf{P}(n-1)\mathbf{x}(n)}$, (9)
- updates weight vector: $\mathbf{w}(n) = \mathbf{w}(n-1) + \mathbf{k}(n) \cdot \alpha(n)$, (10)
- updates the covariance matrix, **P**:

$$\mathbf{P}(n) = \frac{1}{\lambda}\mathbf{P}(n-1) - \frac{1}{\lambda}\mathbf{k}(n)\mathbf{x}^T(n)\mathbf{P}(n-1) \tag{11}$$

Due to the matrix operations in this algorithm, the computational complexity is much higher than that of the LMS algorithm. For a N^{th} order filter, there are O(N) operations per iteration in LMS, while there are $O(N^2)$ operations per iteration in RLS.

3. IMPLEMENTATION OF AN ADAPTIVE NOISE CANCELLING SYSTEM

In interference cancelling theory, a usefull signal is corrupted with uncorrelated interference and it is desired to recover the signal from the observed corrupted signal.

A frequently mentioned application of adaptive noise cancelling is cleaning power-line interference from weak sensor signals. This is essential in applications such as recording electrocardiograms (ECGs), weak vibration measurements, audio frequency measurements using microphones, and many other applications that employ sensors to collect input data. The same technique has also been used in many other applications such as cancelling noise in speech signals, and cancelling antenna sidelobe interference [6].

In this work, in order to experiment the capabilities of an adaptive noise cancelling system, the problem of recovering a sinewave from a noisy environment was selected. In order to cancel the noise, an adaptive filter based on LMS, respective RLS algorithms was used. The system configuration is shown in figure 2.

Adaptive noise cancellers rely on using two separate sensors. The first sensor, the *primary input*, receives a combination of signal and noise ($r(n) = s(n) + x_1(n)$). The second sensor, the *reference input*, receives a noise, $x(n)$, which is uncorrelated with $s(n)$, but correlated in some sense with $x_1(n)$. The reference interference, $x(n)$, is filtered by an adaptive filter in order to produce the filter output, $y(n)$. The adaptive filter coefficients are adjusted so that $y(n)$ becomes as close as possible to the interference at the primary sensor, $x_1(n)$. The recovered signal is then the adaptive filter error: $e(n) = r(n) - y(n)$.

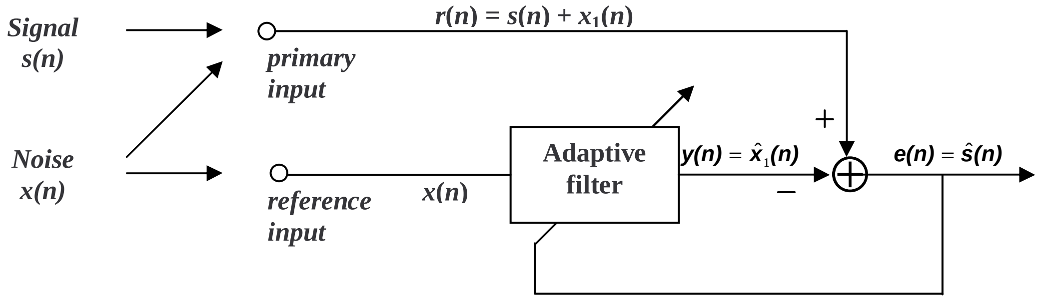

Figure 2: System configuration

The estimation criterion is the minimization of the system output power $E\{|e(k)|^2\}$ (in an adaptive noise cancelling system, the system output serves as the error signal for the adaptive process). By minimizing the system output power, the Mean Square Error (MSE) is minimized: $E\{|e(k) - s(k)|^2\}$.

4. RESULTS ANALYSIS

In the first stage, some Matlab simulations were done. The adaptive filter was implemented using both LMS (initial value for the filter coefficients: $\mathbf{w}(0) = \mathbf{0}$, filter order: $M = 4$, adaptation step: $\mu = 0.01$), and RLS ($\mathbf{P}(0) = 0$; $\lambda = 1$, $M = 4$) algorithms. The sampling frequency used: $f_e = 1000$ Hz. The usefull signal considered is a sinewave generated at a frequency of 50 Hz. The results are presented in figures 4 ÷ 7.

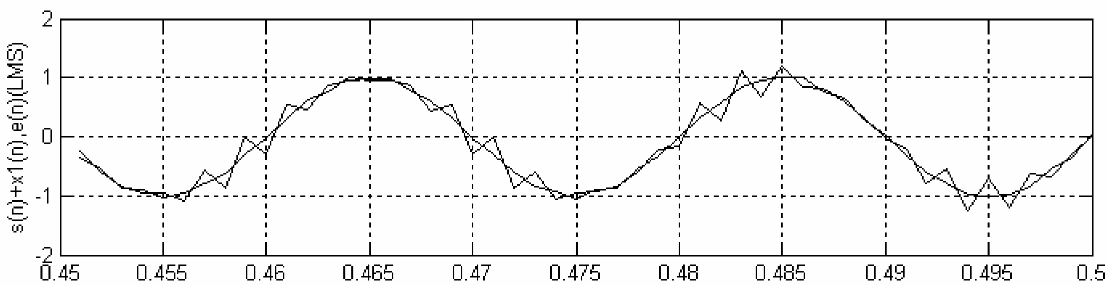

Figure 4: Primary input signal (s(n)+x1(n)); system output (LMS);

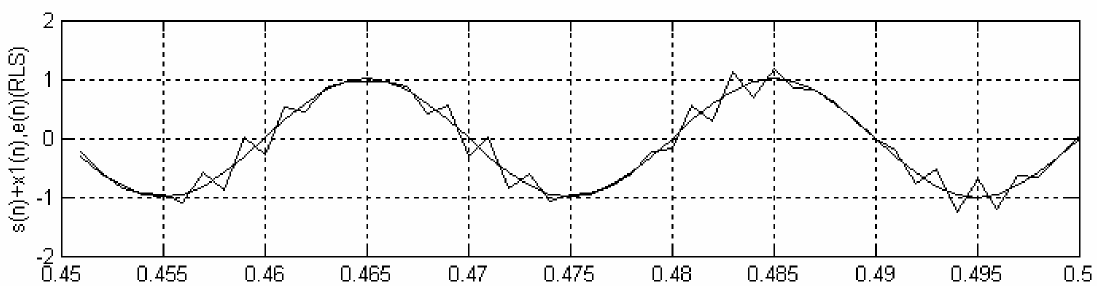

Figure 5: Primary input signal (s(n)+x1(n)); system output (RLS);

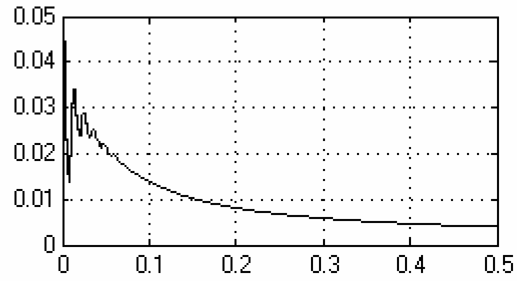

Figure 6. Mean Square Error (LMS)

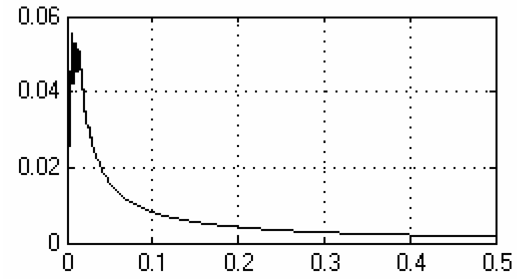

Figure 7. Mean Square Error (RLS)

From the observed results it can be seen that the implemented adaptive filters present good performance in cancelling the noise which affects the usefull signal considered.

In order to compare the convergence speed of the two algorithms, the mean square error, is presented (figure 6, 7). Analyzing MSE, we can say that RLS algorithm presents a better convergence speed for the case considered.

The convergence speed is mainly determined by the adaptation step. Using a larger step size yields to an increase in the convergence speed. Note that if the step-size becomes too large, the filter might become unstable.

In the second stage, the implemented adaptive system based on LMS, respective RLS, was simulated on a PC based TMS320C31 floating-point DSP. The results obtained (figure 8) show that the adaptive system implemented provided good results, and the usefull signal was recovered.

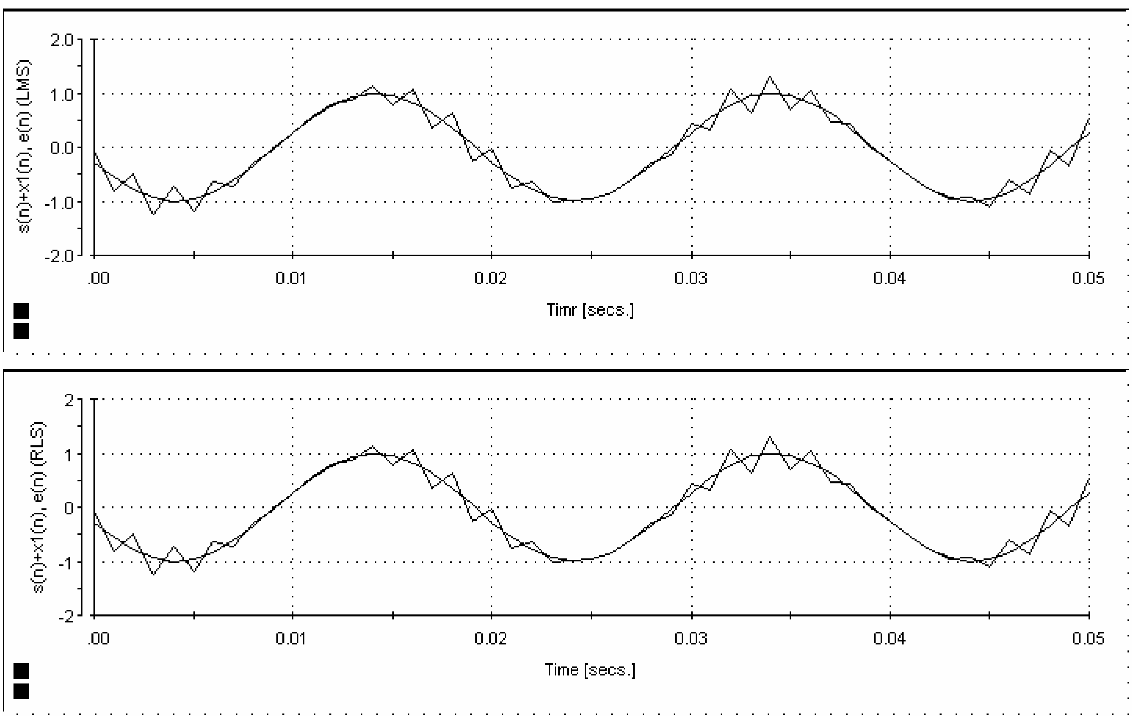

Figure 8. Usefull signal and system output (LMS) ; usefull signal and system output (RLS)

5. CONCLUSIONS

An adaptive noise cancelling system was presented along with the results analysis. From the results analysis it can be seen that the implemented adaptive noise canceller succeeded in estimating the usefull signal from the corrupted signal considered.

The adaptive algorithms used were LMS and RLS. The LSM algorithm is commonly used because of its numerical stability and simplicity. The main problem with the LMS algorithm is that it takes a long time for the filter coefficients to converge because they are adjusted at an identical rate. The RLS adaptive algorithm improves this greatly at the expense of larger computational complexity.

Although the TMS320C31 does not provide any specific instruction for adaptive filter coefficients update, good results can be achieved, because of its powerful architecture. Theoretically, it can implement a very high order of adaptive filter. However, for the most efficient implementation, the limitation of filter order is 2000.

6. REFERENCES

[1] S. Ciochina, C. Negrescu (1999). *Sisteme adaptive*, Editura Tehnica, Bucuresti.

[2] J. Proakis, D.G. Manolakis (1996). *Digital Signal Processing*, Prentice-Hall, New Jersey.

[3] J. Proakis, Charles M. Rader, Fuyun Ling, Christopher L. Nikias (1992). *Advanced Digital Signal Processing*, MacMillan Coll Div.

[4] J. Proakis, V. K. Ingle (2003). *A Self-Study Guide for Digial Signal Processing*, Prentice Hall; Study Guide edition

[5] V. K. Ingle, J. G. Proakis (1999). *Digital Signal Processing Using MATLAB*, Brooks Cole.

[6] ***Texas Instruments (1997). An implementation of adaptive filters with the TMS320C25 or the TMS320C30, *Application Report*: SPRA116

[7] ***Texas Instruments (1997). Implementing a Single Channel Active Adaptive Noise Canceller with the TMS320C50 DSP Starter Kit, *Application Report*: SPRA285

[8] ***Texas Instruments (1996). Design of Active Noise Control Systems with the TMS320 Family, *Application Report*: SPRA042