The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM)
ISSN: 2602-3199

The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM), 2023

Volume 26, Pages 254-261

IConTES 2023: International Conference on Technology, Engineering and Science

dsPIC Implementation and Performances Evaluation of Adaptive Filters for System Identification

Benziane Mourad
University of Bouira

Arabi Abderrazak
University of Setif

Medjedoub Smail
University of Bouira

Ayad Mouloud
University of Setif

Saoudi Kamel
University of Bouira

Abstract: This paper aims to implement and evaluate performances of adaptive filters applied to linear systems identification on a dsPIC platform. To do so, we considered two different applications: the identification of DC motor transfer function and the acoustic echo cancellation. For the DC motor case, real-time identification is not necessarily required. However, for the acoustic echo cancellation case, real-time identification is crucial, making execution time a key parameter in this scenario. We have implemented three adaptive algorithms: LMS, NLMS, and RLS. The first two are known for their computational simplicity, while RLS is recognized for its convergence speed. Performances evaluation is primarily based on accuracy and computation speed. A comparison is carried out in both cases to determine which algorithm is more suitable for each application. The results obtained showed that RLS is better suited for the DC motor case, while NLMS is more suitable for acoustic echo cancellation, particularly when the impulse response of the echo path is long.

Keywords: Systems identification, dsPIC implementation, Adaptive algorithms, LMS, NLMS, RLS, Real-time.

Introduction

Adaptive methods in signal processing aim to automatically adapting processing operators to the statistical properties of signals and systems, as well as their temporal variations. This processing aims to eliminate, or at least reduce, the effect of temporal variations in the properties of signals and systems. Without these variations, there would be no need for adaptivity; one could simply calculate the "optimal filter" once and deploy it (Benesty & Huang, 2000). These methods have known significant growth since the 1960s, motivated by the development of digital processing and the continuous increase in the processing power of Digital Signal Processors (DSPs), allowing the real-time implementation of increasingly sophisticated algorithms at higher speeds processing (Paulo & Diniz, 2013). They have reached a certain level of maturity both in terms of algorithm development and implementation, as well as from the perspective of theoretical tools for performance analysis (Diniz, 2020).

The applications of this approach to processing using adaptive filters are numerous, including system identification, inverse modeling, interference elimination, and predictive filtering (Ghauri & Sohail, 2013). Adaptive filtering algorithms can be categorized into two main groups: LMS, known for their simplicity but limited performance, and RLS, offering significantly better performance at the cost of higher computational complexity (Haykin, 2014). Implementing such algorithms in specific processors, such as dsPIC, ensures their proper operation with improved performance (Colak et al., 2015; Siddiqui et al., 2015; Uriz et al., 2012).

The objective of this work is to implement three basic adaptive algorithms: the LMS, its normalized version NLMS, and the RLS, on a dsPIC microcontroller in the context of linear system identification. We have chosen two different applications: the DC motor, which is a simple linear system, and its identification does not require real-time processing, and echo cancellation, where real-time processing is essential. Evaluating the performance of these algorithms in terms of executing time and accuracy allows us to study their adaptation for the two selected examples.

This paper is organized as follows: Section 2 outlines the methodology used for system identification for the two selected systems. Section 3 presents the mathematical formulation of the three conventional adaptive algorithms. Finally, results are discussed in Section 4.

System Identification Using Adaptive Filters

Presentation of Systems

In this work we have chosen two different systems:
The first one is the DC motor, it has a mathematical model represented by the transfer function given by:

$$G(s) = \frac{K}{s(Ts+1)} \tag{1}$$

The discrete time model is given by

$$d(n+1) = -ad(n) - bd(n-1) + cx(n) \tag{2}$$

$$d(n+1) = \mathbf{h}^T \left[d(n) \quad d(n-1) \quad x(n) \right] \tag{3}$$

Where:

$\mathbf{h}^T = \left[a \quad b \quad c \right]$: impulse response to be identified.

The figure 1 shows the block diagram of DC motor identification.

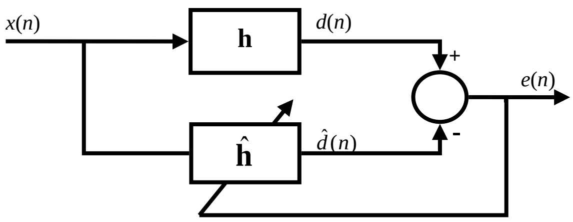

Figure 1. Block diagram of DC motor identification

For this case, the adaptive algorithm tries to find the vector **h**=[*a* *b* *c*] minimizing the error signal between desired signal $d(n)$ and estimated signal $\hat{d}(n)$ (Ogata, 2010). The second system is the echo cancellation. The principle of echo cancellation based on adaptive filtering is shown in figure 2. The general concept is to estimate the echo signal $\hat{d}(n)$ from the far-end signal $x(n)$ using an adaptive filter. This is a typical example of system identification where the system to be identified is the impulse response of the echo path (Vaseghi, 2000).

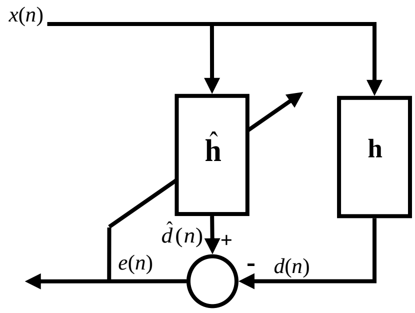

Figure 2. Block diagram of echo canceller

The microphone signal $d(n)$ is obtained as follow:

$$d(n) = \mathbf{h}^T \mathbf{x}_L(n) \tag{4}$$

Where:

$\mathbf{h}^T = \left[h_0 \; h_1 \; \dots \; h_{L-1} \right]$: impulse response of the echo-path with a tap-length L.

$\mathbf{x}_L^T(n) = \left[x(n) \;\; x(n-1) \;\; \dots \;\; x(n-L+1) \right]$: vector of L past samples of the far-end signal.

The echo cancellation is obtained by subtracting the estimate of echo signal $\hat{d}(n)$ from the microphone signal $d(n)$, the estimated echo signal is a combination of fad-end signal $x(n)$ and coefficients of adaptive filter $\hat{\mathbf{h}}$ as following:

$$\hat{d}(n) = \mathbf{h}^T \mathbf{x}_L(n) \tag{5}$$

Where:

$\mathbf{h}^T(n) = \left[h_0(n) \; h_0(n) \; \dots \; h_{L-1}(n) \right]$

We get the error signal:

$$e(n) = d(n) - \hat{d}(n) \tag{6}$$

$$e(n) = d(n) - \mathbf{h}^T \mathbf{x}_L(n) \tag{7}$$

Ideally, this error is equal to zero.

Algorithms

In 1959, engineers Bernard Widrow and Marcian Hoff developed an adaptive filtering algorithm called the Least Mean Square (LMS) algorithm. It has become one of the most widely used algorithms in various applications such as channel equalization, echo cancellation, and noise reduction. The principle of LMS is to adjust the coefficients of a filter to minimize the mean squared error between the desired signal and the output signal (estimated signal). One of the advantages of the LMS algorithm is its computational simplicity, which makes it an attractive solution for many users. The following equations describe the LMS algorithm for updating the adaptive filter weights at each iteration (Gay et al., 2000).

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{x}(n) e(n) \tag{8}$$

$x\,(n)$: vector of input signal with the same length of adaptive filter;
$\mathbf{w}\,(n)$: Vector filter coefficients at time n ;

μ : step size ;

The Normalized LMS (NLMS) is a variant of the LMS algorithm used in signal processing to estimate adaptive filters. Like the LMS algorithm, NLMS iteratively adjusts the filter coefficients based on the prediction error between the actual and estimated filter outputs. The main difference between LMS and NLMS is the normalization of the coefficients updating. In fact, NLMS uses normalization based on the input power of the filter, which improves the algorithm's convergence especially in noisy environments. By the use of the normalized step-size parameter μ in LMS, we obtain another algorithm called Normalized LMS (NLMS). The formula used to calculate the weight update vector is as follows:

$$\mu(n) = \frac{\beta}{c + \|x(n)\|^2} \tag{9}$$

With:

$\mu(n)$: step size at time n.
β : normalized step size ($0 < \beta < 2$)
c : a small constant to avoid division by zero when x(n) is equal to zero.

The Recursive Least Squares (RLS) algorithm implements an exact least squares solution recursively. The Wiener solution for a finite-length adaptive filter is given by:

$$w_{opt} = R^{-1}P \tag{10}$$

Where R is the autocorrelation matrix of the inputs, and P is the cross-correlation between the inputs and the reference signal. At each time interval, RLS recursively estimates R^{-1} and P based on all previous data and calculates the weight vector as follows:

$$w_n \underset{=\!=}{\underline{\mathrm{def}}} R^{-1}P_n \tag{11}$$

The best current approximation of the Wiener solution can be expressed as follows in the coefficients updating formula:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu_n R_{n-1}^{-1} a_n \mathbf{x}_n \tag{12}$$

Where:

$$a_n = d(n) - \mathbf{w}_{n-1}^T \mathbf{x}_n$$

$$\mu_n = \frac{\lambda^{-1}}{1 + \lambda^{-1}\mathbf{x}_n \mathbf{x}_n^T R_{n-1}^{-1}}$$

λ : is a constant slightly less than 1, called forgetting factor.

Results and Discussion

Matlab Simulation

First, a simulation is performed in Matlab to select the parameter values for each algorithm before the implementation in the dsPIC. We will implement the two system identification examples using adaptive filters based on the basic LMS, NLMS, and RLS algorithms.

DC motor Identification

The first system is the DC motor, which has a second-order transfer function, meaning we have three coefficients to identify. To achieve this, we will use the signal x(n) as the excitation signal, which is given by:

$$x(n) = \sin(0.1n) + \sin(0.2*n) + \sin(0.3n) + 3\sin(n) \quad (13)$$

In this case, the length of the adaptive filter is 3. The number of samples in the signal $x(n)$ is 16000. After several simulations, we have chosen the following parameters:

- LMS (μ=0.05)
- NLMS (β=0.5, C=0.00001)
- RLS (λ=0.95, δ=0.005).

The comparison criterion in this case is the square error between the system output and the output of the adaptive filter, given by:

$$SE(n) = \left[d(n) - \hat{d}(n)\right]^2 \quad (14)$$

By setting the target for this error to 0.0001, we obtained the following results:

Table 1. SE obtained with Matlab simulation for DC motor model

Algorithm	Number of necessary iterations	SE after 200 iterations	SE after 1000 iterations
LMS	1127	$2.04*10^{-4}$	$0.69*10^{-4}$
NLMS	385	$7.76*10^{-4}$	$7.12*10^{-9}$
RLS	133	$0.01*10^{-4}$	$2.15*10^{-23}$

It's evident that the RLS algorithm demonstrates better convergence compared to LMS and NLMS, meaning it reaches the desired SE error after a lower number of iterations.

Echo Canceller

For the echo canceller, the goal in this case is to identify the impulse response of the echo path to be able to estimate the echo signal and subtract it from the transmit signal. We have used an impulse response composed of 128 samples (Figure 3). For the excitation signal (far-end signal), we have chosen an audio signal sampled at 8 KHz and consisting of 40000 samples (Figure 4).

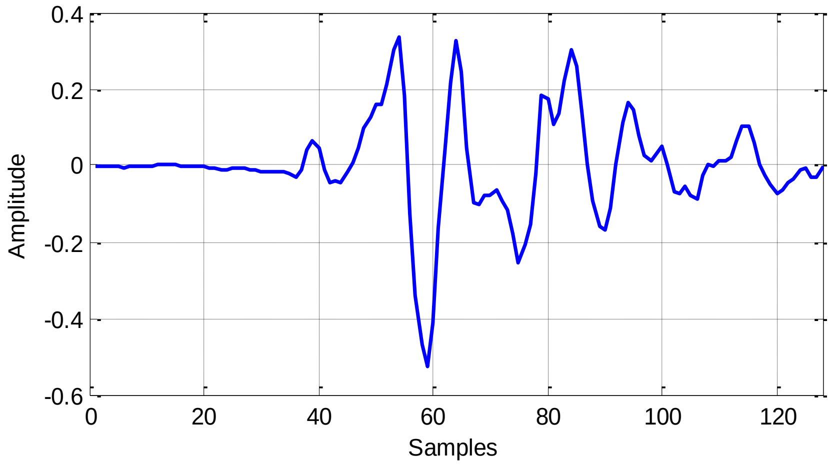

Figure 3. Impulse response of echo path

We proceeded in the same manner to set the parameters for each algorithm to ensure good convergence, this time based on misalignment as the evaluation criterion, with its expression given by:

$$Mis(dB) = 10\log_{10}\left[\frac{\left\|\hat{\mathbf{h}}(n) - \mathbf{h}\right\|^2}{\left\|\mathbf{h}\right\|^2}\right] \quad (15)$$

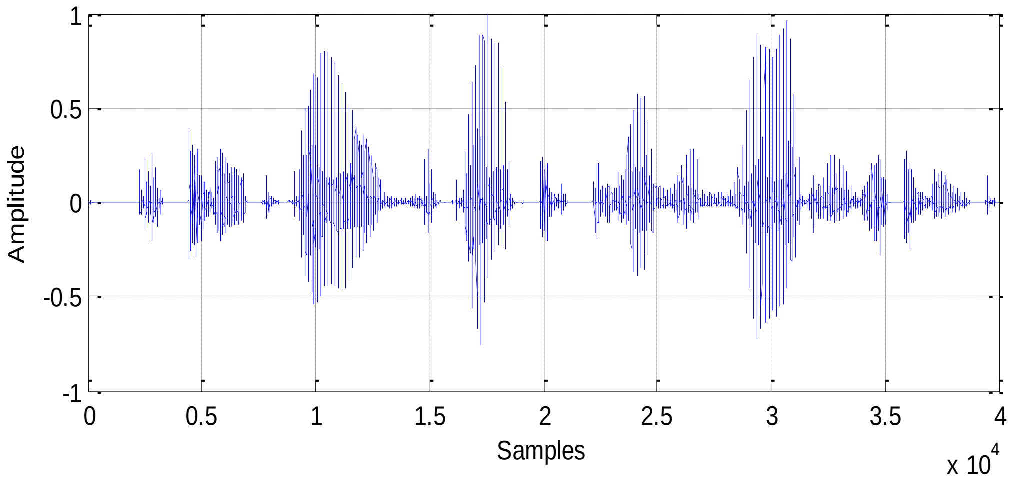

Figure 4. Far-end signal

In this case, the length of the adaptive filter is 128, and the number of samples in the signal x(n) is 40000. After several simulations, we chose the following parameters:

- LMS (μ=0.05)
- NLMS (β=0.6, C=0.09)
- RLS (λ=0.95, δ=0.005)

By setting the misalignment objective to -15dB, we obtain the results presented in the table 2:

Table 2. Misalignements obtained with Matlab simulation

Algorithm	Number of necessary iterations	Misalignement after 10000 iterations	Misalignement after 20000 iterations
LMS	17480	-5.024 dB	-17.03
NLMS	5500	-25.71 dB	-36.36
RLS	2375	-127.1 dB	-212.5

We observe that the RLS algorithm consistently shows good convergence, even with a longer impulse response than in the first case.

Implementation and Performances Evaluation

In this step, we will implement the three algorithms for the two selected systems on a dsPIC. The dsPIC is a fast 16-bit microcontroller designed for signal processing applications. This component is used in various embedded applications and is suitable for real-time signal processing tasks, particularly for linear system identification that requires real-time adaptation of adaptive filter coefficients. Among the microcontrollers in this family, we have chosen the dsPIC33FJ64MC804, which integrates more than 35 programmable digital I/Os and over 6 analog inputs [14]. The programming is done using an advanced language, specifically C++, and a compiler dedicated to dsPIC. The objective of this implementation is to evaluate the performance in terms of execution speed and accuracy.

Accuracy Analysis

The calculation accuracy is evaluated based on the results obtained in Matlab for defining the parameters of each algorithm. The tables 3 and 4 summarize the results obtained for the DC motor and the echo canceller, respectively.

The difference between the results obtained with Matlab and on the dsPIC is primarily due to that Matlab uses 64-bit floating-point calculations, while the dsPIC performs calculations using 16 bits. In comparison to the results obtained, the calculation precision of the dsPIC is more than sufficient for the two implemented systems.

Table 3. SE obtained for DC motor model on dsPIC

Algorithm	SE after 200 iterations	SE after 1000 iterations	SE after 1000 iterations
LMS	0.1032	0.0002	$<10^{-6}$
NLMS	0.0033	$<10^{-6}$	$<10^{-6}$
RLS	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$

Table 4. SE obtained for echo canceller on dsPIC

Algorithm	SE after 10000 iterations	SE after 20000 iterations	SE after 30000 iterations
LMS	0.0303	$0.96*10^{-4}$	$0.11*10^{-4}$
NLMS	$0.03*10^{-4}$	$0.02*10^{-4}$	$<10^{-6}$
RLS	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$

Processing Speed

The tables 5 and 6 represent the results obtained for the processing speed in the cases of the DC motor and the echo canceller, respectively.

Table 5. Execution time of DC motor identification

Algorithm	Execution time of one iteration (µs)	Number of necessary iterations	Necessary time for identification (ms)
LMS	7.25	17480	126
NLMS	8.5	5500	46.7
RLS	17	2375	40.3

Table 6. Execution time of echo canceller system

Algorithm	Number of necessary iterations	Exeution time of 1 iteration (µs)
LMS	17480	19.5
NLMS	5500	20.75
RLS	2375	45

For the DC motor identification, we have calculated the execution time for one iteration and the total time required for identification based on the error margin set during the Matlab simulation. It's clear that RLS has the shortest identification time among the three algorithms (40.3 ms), despite its longer iteration execution time (17 µs), thanks to its high convergence speed.

In the case of the echo canceller, execution time plays a crucial role since it's a real-time application where execution must occur within the sampling period of the processed signal. For telephony applications, the sampling frequency is typically 8 KHz, which corresponds to a sampling period of 125 µs. All three algorithms meet this constraint and have execution times below 125 µs. It's worth noting that execution time would increase if a longer adaptive filter were used to improve precision.

Conclusion

In this study, the implementation of three adaptive filtering algorithms was carried out on a dsPIC for the identification of linear systems. Two different systems were considered: the DC motor, requiring the identification of three coefficients, and an echo canceller, where identification was performed on the 128 samples of the echo path's impulse response. The objective of this study is to evaluate performance in terms of execution time and accuracy.

The results showed that LMS has a slow convergence rate and low execution time due to its simplicity. Therefore, it is not recommended for real-time applications requiring fast adaptive filter convergence. These algorithms are better suited for cases where identification can be done offline, as in the DC motor case where the time required for identification is not critically important. The performance of LMS was improved with its normalized version, NLMS, which has faster adaptive filter convergence. Considering its performance compared to its simplicity, NLMS is widely used in many applications. RLS exhibited the fastest convergence speed. Its major drawback is its computational complexity, especially for adaptive filters with long impulse responses. However, with current advancements in specialized processors for complex calculations, RLS is becoming increasingly viable for various applications.

Scientific Ethics Declaration

The authors declare that the scientific ethical and legal responsibility of this article published in EPSTEM journal belongs to the authors.

Acknowledgements or Notes

This article was presented as an oral presentation at the International Conference on Technology, Engineering and Science (www.icontes.net) held in Antalya/Turkey on November 16-19, 2023.

References

Benesty, J., & Huang, Y. (2003). *Adaptive signal processing-application to real-world problems*. Berlin : Springer.

Colak, I. S. (2015). Controller design for a limited angle torque motor and dsPIC implementation In *OPTIM, ELECTROMOTION Conference* (pp.649-654). Side, Turkey

Diniz, P. S. R. (2020). *Adaptive filtering*. Switzerland AG: Springer Nature

Gay, S. L., Steven, L., & Benesty, J. (2000). *An introduction to acoustic echo and noise control, acoustic signal processing for telecommunication* (pp.1-19). Boston, MA: Springer

Ghauri, S. A., & Sohail, M. F. (2013). System identification using LMS, NLMS and RLS. *2013 IEEE Student Conference on Research and Development (SCOReD*). Retrieved from http://www.microchip.com

Haykin, S., (2014). *Adaptive filter theory* (5 th ed.). NJ: Pearson Education.

Ogata, K., (2010). *Modern control engineering* (5th ed.). Prentice Hall.

Paulo, S., & Diniz, R. (2013). *Adaptive filtering, algorithms and practical implementation* (4th ed.). Newyork, NY: Springer

Siddiqui, R. A., Grosvenor, R. I., & Prickett, P.W. (2015). dsPIC-based advanced data acquisition system for monitoring, control and security applications. *12 th International Bhurban Conference on Applied Science and Technology*. Islamabad, Pakistan.

Uriz, A. J. (2012). An Implementation in dsPIC of a denoising algorithm based on the discrete wavelet transform. *Andean Region International Conference.*

Vaseghi, S. V. (2000) *Echo cancellation, in advanced digital signal processing and noise reduction*, (pp.396–413). Newyork, NY: John Wiley & Sons

Author Information

Benziane Mourad
University of Bouira, Algeria
Contact e-mail : *m.benziane@univ-bouira.dz*

Arabi Abderrazak
University of Sétif, Algeria

Medjedoub Smail
University of Bouira, Algeria

Ayad Mouloud
University of Sétif, Algeria

Saoudi Kamel
University of Bouira, Algeria

To cite this article:

Mourad, B., Abderrazak, A., Smail, M.., Mouloud, A., & Kamel, S. (2023). dsPIC implementation and performances evaluation of adaptive filters for system identification. *The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM), 26,* 254-261.