A Novel Approach to Introducing Adaptive Filters Based on the LMS Algorithm and Its Variants

Emilio Soria, Javier Calpe, *Member, IEEE*, Jonathon Chambers, *Senior Member, IEEE*, Marcelino Martínez, Gustavo Camps, *Member, IEEE*, and José David Martín Guerrero

***Abstract*—This paper presents a new approach to introducing adaptive filters based on the least-mean-square (LMS) algorithm and its variants in an undergraduate course on digital signal processing. Unlike other filters currently taught to undergraduate students, these filters are nonlinear and time variant. This proposal introduces adaptive filtering in the context of a linear time-invariant system using a real problem. In this way, introducing adaptive filters using concepts already familiar to the students motivates their interest through practical application. The key point for this simplification is that the input to the filter is constant so that the adaptive filter becomes linear. Therefore, a complete arsenal of mathematical tools, already known by the students, is available to analyze the performance of the filters and obtain the key parameters to adaptive filters, e.g., speed of convergence and stability. Several variants of the basic LMS algorithm are described the same way.**

***Index Terms*—Adaptive filters, algorithms, digital signal processing (DSP).**

I. Introduction

ADAPTIVE filters have been demonstrated to be useful since they were first introduced by Widrow and Hoff during the 1960s [1]. Thereafter, they have found countless applications [2]. As a consequence, adaptive filters have been included in the syllabus of undergraduate digital signal processing (DSP) courses. Unfortunately, these filters do not adapt well to the normal course contents for several reasons.

- *Adaptive filters are time variant*. Therefore, most of the analysis tools provided to the students (e.g., z or Fourier transforms) are not directly applicable.
- *Modification of adaptive filters coefficients is nonlinear*. This modification provokes dynamic behaviors (e.g., chaotic behavior in the output).
- *Adaptive filters rely on certain constants*. The boundaries of these constants are fixed using rather advanced mathematical tools unfamiliar to students. Therefore, students observe that the filter performs well for certain values but do not understand why. Teachers want to avoid this situation.

This communication proposes introducing the least-mean-square (LMS) algorithm and some of its variants through a real problem, such as the conditioning of a signal for estimating the weight of individual fruits traveling on a conveyor belt. The autors' aim is to provide an intuitive view of the process and show how a transfer function for an adaptive filter can be obtained in particular cases. This fact is used to explain the performance of adaptive systems as a function of the values of the constants included in their expressions.

II. Description of the Problem

Dynamic weighting is a common application in some industrial areas, for example, fruit-sorting and -grading machinery in fruit-packing houses. In this case, the authors try to estimate the weight of fruits traveling onto a conveyor belt with individual cups that contain separate fruits, using a load cell and the minimum analog conditioning and amplifying circuitry [3], [4].

In order to obtain the registers, a real commercial fruit-sorting and -grading machine provided by Maxfrut, SL, Alzira, Spain, with two sorting lines was used. The acquisition hardware is a modified board card provided by Dismuntel, SL, Algemesí, Spain, based on the LTC1100 instrumentation amplifier from Linear Technologies. Data were acquired with a DAQ-Card AT-MIO16 from National Instruments, with a low-pass filter and a cutoff frequency of 200 Hz sampled at 1 kHz with a 16-b resolution. The 10-lb load cell is steel-made by Artech Industries, Inc., with 2.096 mV/V @ 10 lbs.

The speed of the conveyor varies from 2–15 fruits/s, depending on the nature of the goods processed. Obviously, the quality of the signal strongly depends on this speed (Fig. 1). In this paper, the process by which the actual weight is estimated from the load-cell measurement is not of direct concern; rather, the focus is upon the preprocessing of the load-cell measurement over some observation window. Such a window of data is termed a data register.

At first sight, the signals seem to be heavily distorted by power-line noise. To verify this hypothesis, a basic spectral analysis of the signal was performed. An average fast Fourier transform (FFT) of a series of 10-s intervals rules out this possibility. The origin of the distortion is the dynamic response of the load cell and the machine vibration. Simple algorithms for weight estimations, such as moving average, are only acceptable at low conveyor speeds.

Then, consider the ideal waveform of a load-cell response to propose other alternatives (Fig. 2). The height of the pulse a depends on the weight and the width b on the conveyor speed.

Manuscript received March 26, 2002; revised January 30, 2003.

E. Soria, J. Calpe, M. Martínez, G. Camps, and J. D. M. Guerrero are with the Digital Signal Processing Group (GPDS), University of València, 46100 València, Spain (e-mail: emilio.soria@uv.es).

J. Chambers is with the Centre for Digital Signal Processing Research, Kings College London, London WC2R 2LS, U.K.

Digital Object Identifier 10.1109/TE.2003.822632

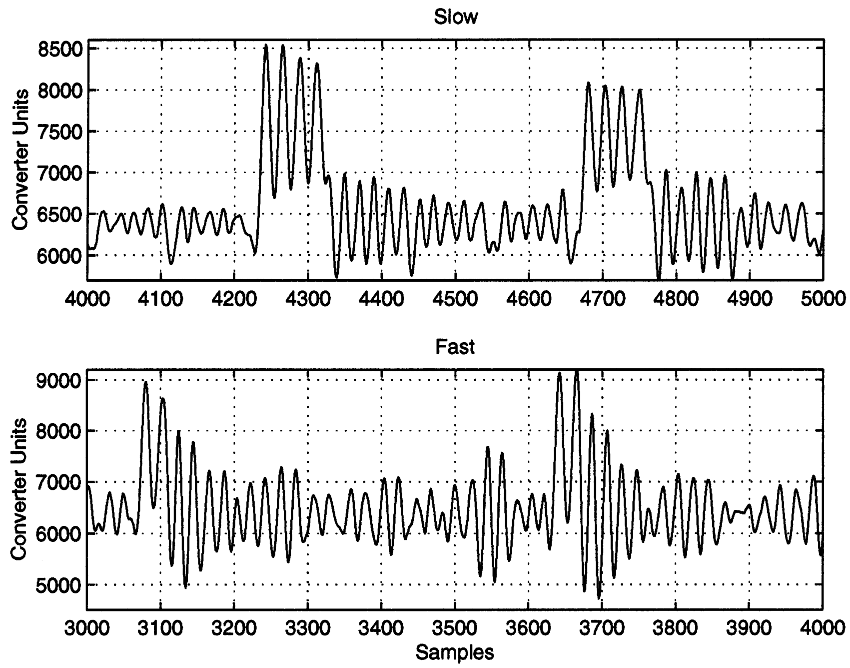

Fig. 1. Raw signals at lower (5 fruits/s) and higher (12 fruits/s) speeds of the conveyor. Vertical axis contains the values given by the analog-to-digital converter (1 gr is approximately 6 u), and the sampling rate is 600 Hz.

The algorithm must be fast enough to estimate the weight at maximum speed (minimum value of b).

Fig. 2 shows that weight estimation is basically reduced to a *local mean value estimator*, whose length depends on the conveyor speed.

Since this estimator suits adaptive filters very well, a monotonic increasing error function must be defined. The most commonly used one is the mean-square error.

Fig. 3 shows a simple adaptive structure that may solve the problem. The input to the adaptive filter is a constant value (1 for simplicity), and the length of the filter is 1. In this way, the load-cell (reference) signal and the input to the filter are uncorrelated except for the direct current (dc) component of the load-cell signal. Therefore, the minimum for the error function is achieved when the average (dc estimation) of the reference signal (weight estimation) is equal to the output of the filter.

The most exploited adaptive algorithms, the LMS and some of its variants [2], are applied to solve this problem. The next section shows the theoretical development used to obtain the performance characteristics of these algorithms in this particular problem.

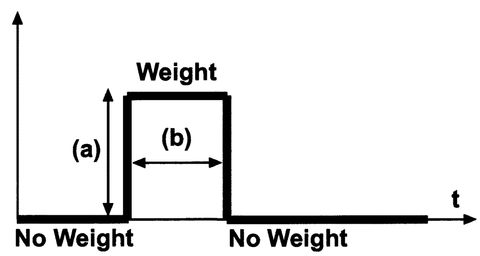

Fig. 2. Sketch of a “perfect” weighting signal.

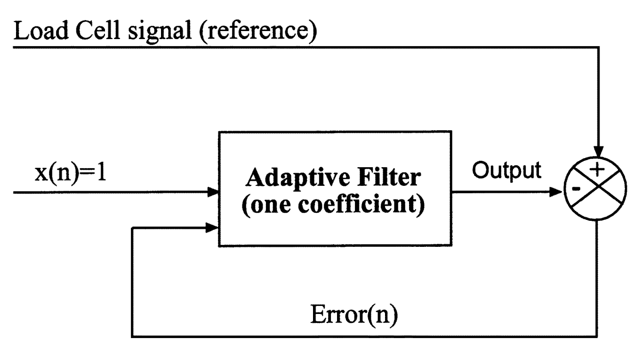

Fig. 3. Scheme of the proposed adaptive filter. Error is minimized when the dc component of the load cell matches the output of the filter.

III. Theoretical Development

A. The LMS Algorithm

A commonly exploited technique to determine the minimum of a function is the method of “steepest descent.” It is an iterative method defined by

$$w_{n+1} = w_n - \frac{1}{2} \cdot \alpha \cdot \nabla_n J \tag{1}$$

where J is the function to be minimized, w_n is a column vector that contains the parameters of the adaptive filter at instant n, and α is a parameter (adaptation constant). Equation (1) has a clear intuitive meaning. Parameters at a given instant are obtained from the current values, slightly modified according to the direction of the steepest descent of J. Vector analysis states that this direction is opposite to the gradient of the cost function [2]. The constant 1/2 is added to simplify the final expressions. Fig. 4 shows an example of the evolution of the two weights (filter coefficients) and the error (sum squared error) toward the minimum of the error function.

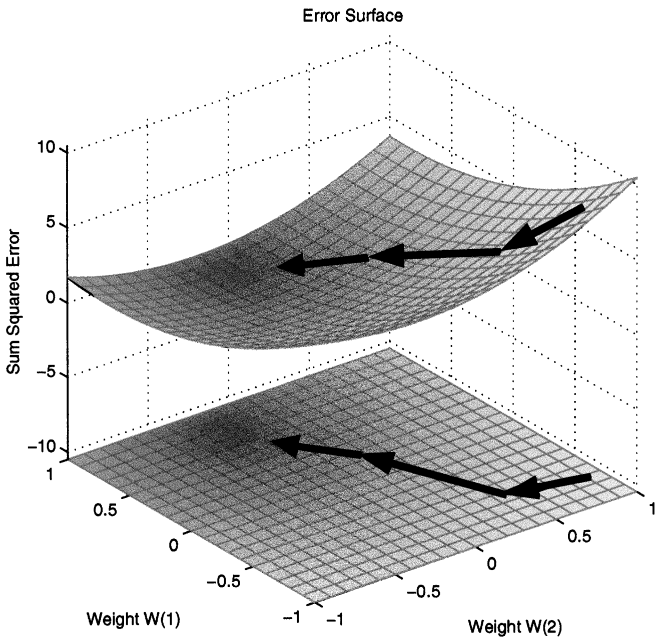

Fig. 4. Scheme of the steepest descent method.

The LMS algorithm considers J as the instantaneous squared error, defined as [5], [6]

$$J = [e(n)]^2 \tag{2}$$

where $e(n)$ is the error committed by the adaptive filter. This error is given by the difference between the desired signal $d(n)$ and the filter output $y(n)$, as

$$e(n) = d(n) - w_n^t \cdot X_n \tag{3}$$

where $X_n = [x(n), x(n-1), \ldots x(n-L+1)]^t$, $w_n = [w_n^{(1)}, w_n^{(2)}, \ldots, w_n^{(L)}]^t$ and $(.)^t$ denotes vector transposition. From (2) and (3), (1) can be written as

$$w_{n+1} = w_n + \alpha \cdot e(n) \cdot X_n \tag{4}$$

which contains the update of the filter coefficients for the basic LMS algorithm. Nonlinearity appears since there is a multiplication between the error and the input signals.

In this special case, these expressions are simplified. Since X_n is equal to one, and the length of the filter is one, the filter outcome is the filter coefficient. Therefore, (4) can be written as

$$w_{n+1} = w_n + \alpha \cdot (d(n) - w_n). \tag{5}$$

Equation (5) is a linear equation since the multiplication in (4) vanishes under these special conditions. Then, a transfer function can be obtained for the filter by applying the z transform to both sides of (5), as follows:

$$\frac{W(z)}{D(z)} = \frac{\alpha \cdot z^{-1}}{1 - (1-\alpha) \cdot z^{-1}}. \tag{6}$$

This transfer function enables the analysis of the filter performance as a function of α.

- *Stability*. There is a single pole at $1 - \alpha$; therefore, the boundaries for the adaptation constant are 0 and 2, which correspond to the values usually specified for adaptive filters [5].

$$0 < \alpha < \frac{2}{L \cdot E[x^2(n)]} \tag{7}$$

where L is the length of the filter and the $E[x^2(n)]$ is the energy of the input. In this case, both values are equal to 1.

- *Speed of convergence*. The adaptation constant controls the speed of convergence of the adaptive filter. The smaller the constant, the slower the convergence. For a small value of α, the time constant for the ith coefficient of the filter is given by [7]

$$\tau_i = \frac{1}{\lambda_i \cdot \alpha} \tag{8}$$

where λ_i is the ith eigenvalue of the autocorrelation matrix of the input signal. In this case, $\lambda_i = 1$, so $\tau_i = 1/\alpha$.

To obtain this expression for this case, from (5), the impulse response of the filter (assumed to be causal) is

$$h(n) = (1-\alpha)^n \cdot \alpha \cdot u(n-1) \tag{9}$$

where $u(n)$ is the step function.

If η is defined as the sampling instant when the signal decreases its value from the maximum $(n = 1)$ to e^{-1} of this value, one obtains

$$(1-\alpha)^\eta \cdot \alpha = \frac{\alpha \cdot (1-\alpha)}{e} \tag{10}$$

which can be written as

$$(\eta - 1) \cdot \ln(1-\alpha) = -1. \tag{11}$$

Since α is usually very small, the following approximation is proposed:

$$\ln(1-\alpha) \cong -\alpha. \tag{12}$$

Thus, (11) is reduced to

$$\eta - 1 \cong \frac{1}{\alpha} \tag{13}$$

where $\eta - 1$ is the number of samples required by the filter to decay from its maximum value to a certain percentage of that maximum $(1/e)$. This value is equivalent to the speed of convergence and matches (8).

In a similar way, different variants of the basic LMS algorithm can be analyzed for this special case, and decisions about their suitability can be derived.

Averaged LMS Variant: Any real system is contaminated by a series of random interferences, such as measurement errors, machine vibrations, and drifts in analog components. To remove this interference, the averaged variant is proposed. The averaged LMS (ALMS) algorithm updates the filter coefficients according to

$$w_{n+1} = w_n + \frac{\alpha}{N} \sum_{i=0}^{N-1} e(n-i) \cdot X_{n-i}. \tag{14}$$

The improved performance of the ALMS algorithm in the presence of noise is because of the averaging of the gradient terms, which reduces the effect of the Gaussian noise on the filter coefficients. In this case, (14) is written as

$$w_{n+1} = w_n + \frac{\alpha}{N}\sum_{i=0}^{N-1} d(n-i) - w_{n-i}. \tag{15}$$

If one applies the z transform of this expression

$$H(z) = \frac{W(z)}{D(z)} = \frac{\beta \cdot z^{-1} \cdot (1 - z^{-N})}{(1-z^{-1})^2 + \beta \cdot z^{-1} \cdot (1 - z^{-N})} \tag{16}$$

where $\beta = \alpha/N$. The dc gain is unity. In order to obtain the transient response and stability of the filter, one should obtain the poles of the transfer function. To simplify this action, (16) is written as

$$H(z) = \frac{W(z)}{D(z)} = \frac{\beta \cdot (z^N - 1)}{z^{N-1} \cdot (z^2 - 2z + 1) + \beta \cdot (z^N - 1)}. \tag{17}$$

Zeros are uniformly distributed along a circumference with radius 1. Pole locations depend on the average length and the adaptation constant α. To obtain them, we propose the root-locus method. Fig. 5 shows the results of applying this technique for $N = 2$. The unit circumference is represented to check stability. MATLAB and its control library (*rlocus* instruction) were used to generate the graphic in Fig. 5.

Using these techniques, stability can be verified, and one can ascertain the aspect of the impulse response: oscillatory if poles are complex, and exponential if real. Therefore, one has

- impulse response:

$$0 < \beta < 3 - 2\sqrt{2} \Rightarrow \text{Non-oscillating response}$$
$$3 - 2\sqrt{2} < \beta < 1 \Rightarrow \text{Oscillating response}$$

- stability:

$$0 < \beta < 1.$$

Momentum LMS Variant: As the speed of the conveyor increases, the convergence of the adaptive filter must be faster in order to avoid overlap between consecutive stimuli to the load cell. The momentum LMS (MLMS) variant [5] increases the adaptation speed by adding to the update expression of the LMS a term that depends on the gradient of the last update, as follows:

$$\Delta w_n = \alpha \cdot e(n) \cdot X_n + \mu \cdot \Delta w_{n-1} \tag{18}$$

where μ is the so-called momentum constant.

If one adapts this expression to the single filter coefficient case, one obtains

$$w_{n+1} = w_n + \alpha \cdot (d(n) - w_n) + \mu \cdot (w_n - w_{n-1}). \tag{19}$$

If one applies the z transform to (19)

$$H(z) = \frac{W(z)}{D(z)} = \frac{\alpha \cdot z}{z^2 + (\alpha - \mu - 1) \cdot z + \mu}. \tag{20}$$

Once again, a unity dc gain is observed.

Stability and transient response depend on both adaptation and momentum constants. To analyze the filter behavior, one must adjust to fix the momentum parameter and vary the adaptation constant. Therefore, if one obtains parameter α as a function of the poles and μ, one obtains

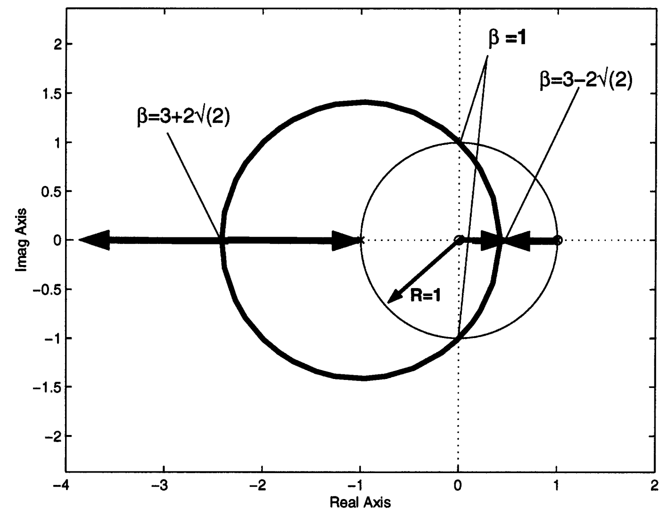

Fig. 5. $H(z)$ pole positioning (in bold face) as a function of β for $N = 2$. Stability is assured for $0 < \beta < 1$, and an oscillating response appears when $\beta > 3 - 2\sqrt{2}$.

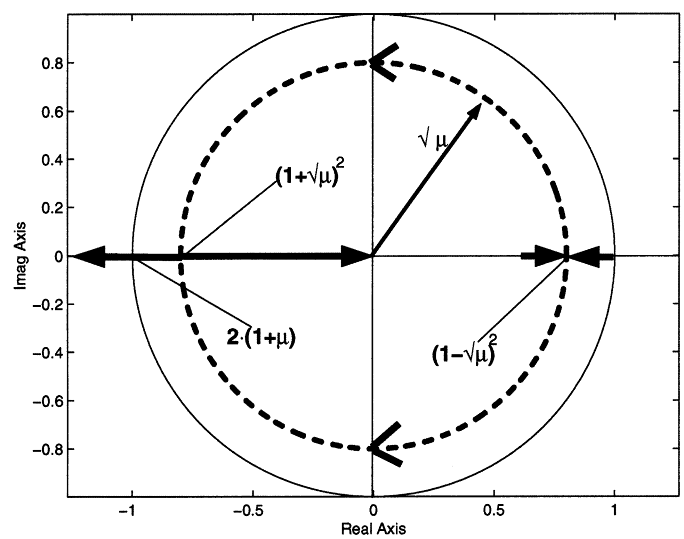

Fig. 6. Root locus obtained for α after fixing μ.

$$\alpha = -\frac{(z-1) \cdot (z-\mu)}{z}. \tag{21}$$

As $\mu < 1$ (to assure stability), the root locus for this expression is shown in Fig. 6.

Fig. 6 shows the break points and the values of α that make the filter unstable (poles out of the unit circle). This diagram makes it possible to draw conclusions about the impulse response and stability. One can ascertain

- impulse response:

$$0 < \alpha < (1 - \sqrt{\mu})^2 \Rightarrow \text{Non-Oscillating response}$$
$$(1 - \sqrt{\mu})^2 < \alpha < (1 + \sqrt{\mu})^2 \Rightarrow \text{Oscillating response}$$
$$(1 + \sqrt{\mu})^2 < \alpha < \infty \Rightarrow \text{Non-Oscillating response}$$

- stability:

$$0 < \alpha < 2 \cdot (1 + \mu).$$

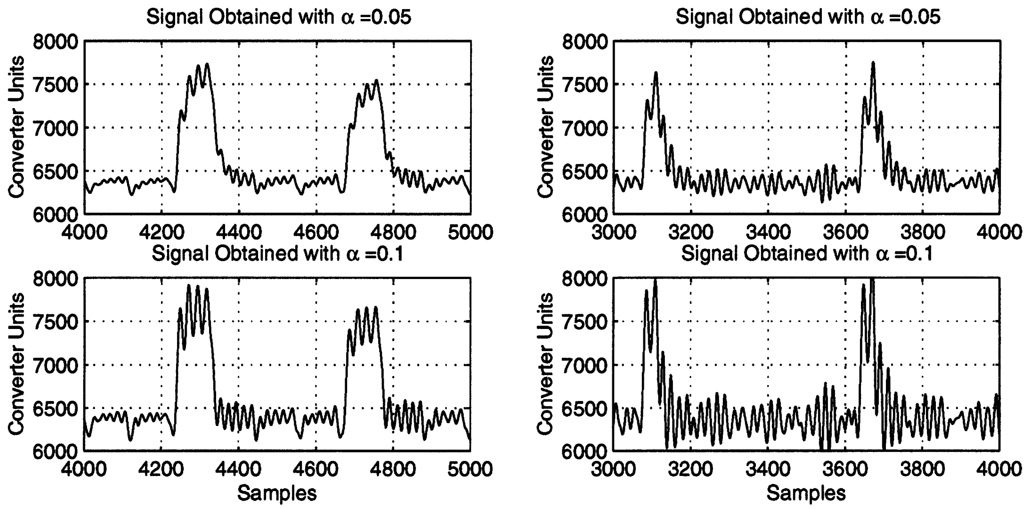

Fig. 7. Outcomes after applying the basic LMS algorithm on the low- and high-speed registers with two settings of alpha.

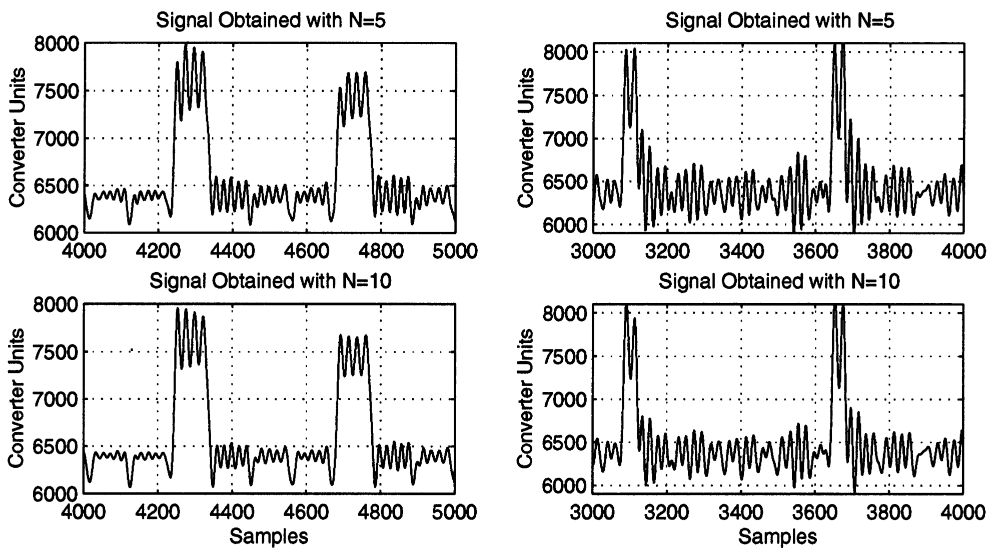

Fig. 8. Outcomes after applying the ALMS algorithm on the low- and high-speed registers with two settings of alpha.

Final Comments: The variants described here are those that yielded better results in this problem. Other variants that modify the gradient with linear operators can be analyzed in the same way [5]. The normalized LMS (NLMS) is not considered despite its generally good speed performance [8], because in this case it is equivalent to the basic LMS.

IV. Experimental Results

Students are provided with a set of real registers at several speeds, with different weights and configurations, i.e., fruits in adjacent or nonadjacent cups or not, combinations of heavy and light fruits, and dummy load cells (attached to the machine but not weighting).

They are asked to visualize certain registers both in the time and frequency domains. In this way, they rule out power-line noise as the origin of distortion. They also verify the performance of basic low-pass filtering techniques (basically, moving averages) to observe their poor performance at speeds above 4 fruits/s.

Afterwards the students are asked to program the three adaptive algorithms, and test their performance, stability, and speed of convergence depending on the constants. Fig. 1 shows the reference signals used to check the performance of the aforementioned adaptive algorithms: one for low conveyor speed (5 fruits/s), and another for higher speeds (12 fruits/s). Figs. 7–9 show the results obtained with the different adaptive algorithms and constant values. On the left, low-speed registers are shown, and on the right the high-speed ones.

Fig. 7 shows the performance of the basic LMS. The most remarkable point is that the increase of the adaptation constant speeds up the convergence but endangers stability, a situation reflected in the increase of the oscillation amplitudes.

For the ALMS variant, the adaptation value was fixed at 0.1, and different average lengths were tested. Fig. 8 shows the main characteristic of the ALMS, i.e., its ability to reduce the Gaussian noise in the signal. As the average length increases, lower amplitude oscillations are observed in the ideally flat tracts.

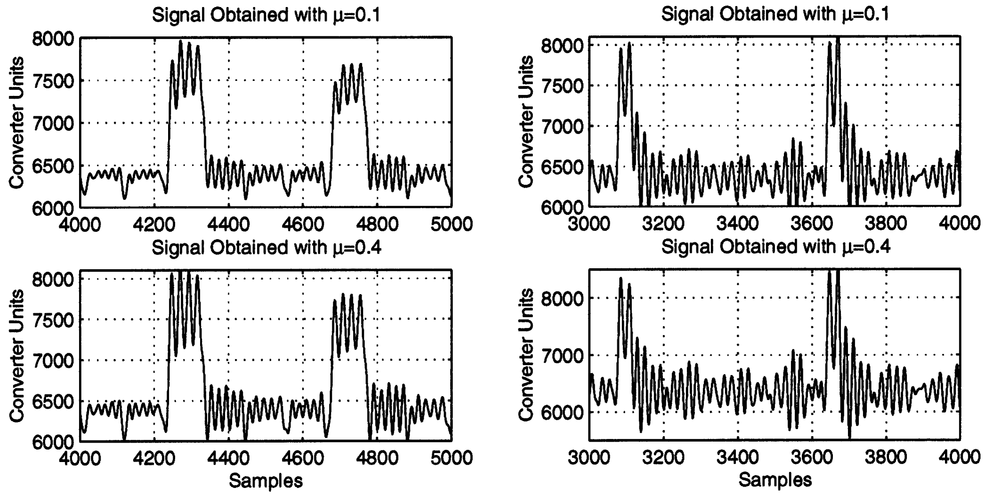

Fig. 9. Outcomes after applying the basic MLMS algorithm on the low- and high-speed registers with two settings of alpha.

TABLE I
APPROXIMATE AMPLITUDES OF THE LAST COMPLETE OSCILLATION AT LOW SPEED MEASURED IN CONVERTER UNITS (1 g ≈ 13 u)

	1st fruit oscillation	2nd fruit oscillation
Original	1500	1150
LMS (α=0.05)	250	160
LMS (α=0.1)	500	300
ALMS (N=5)	650	300
ALMS (N=10)	550	200
MLMS (μ=0.1)	550	250
MLMS (μ=0.4)	950	450

For the MLMS variant, the adaptation constant was fixed at 0.1, and different momentum constants were tested. Fig. 9 shows how, as the momentum constant increases, the convergence speed of the filter increases as well [5]. Low-momentum constants do not improve the results obtained with the basic LMS algorithm.

All these algorithms improve the quality of the original signal. This fact is easily observed by comparing Fig. 1 with Figs. 7–9 at low speeds. The oscillation amplitudes in the weighting plateaus are drastically reduced (Table I), and cup-to-cup transitions are not distinguished in the original signal but are evident in the processed ones. Table I suggests that the LMS algorithm with a small adaptation constant is the best option; nevertheless, this option slows the response of the system, and hence, it may be unsuitable for higher speeds.

Some of the real records and MATLAB routines can be freely downloaded from http://www.uv.es/~soriae/pesada.htm

V. CONCLUSION

This paper presents a strategy for introducing adaptive filters in an undergraduate DSP course based on a real application. Because adaptive filters are time variant, they require different analysis tools than the usual linear time-invariant systems. This requirement provokes a rather descriptive and unproven introduction of their characteristics. This communication solves the problem by using a real application that simplifies the problem and shows the student the usefulness of adaptive filters. The key point for this simplification is that the input to the filter is constant so that the adaptive filter becomes linear. Therefore, a complete arsenal of mathematical tools, already known by the students, is available to analyze the performance of the filters and obtain the key parameters to adaptive filters, e.g., speed of convergence and stability. Several variants of the basic LMS algorithm are described the same way. With this introduction, the student is better prepared to entirely understand the basic concepts of adaptive filters.

REFERENCES

[1] B. Widrow and M. Hoff, "Adaptive switching circuits," in *Neurocomputing*. Cambridge, MA: MIT Press, 1988, pp. 123–134.
[2] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1996.
[3] J. Calpe, J. V. Francés, M. Martínez, E. Soria, A. Rosado, J. Guerrero, J. Calleja, M. Díaz, and N. Ugolini, "DSP based weighting system for fruit sorting and grading machinery," presented at the *Int. Conf. Signal Processing Applicat. Technol.*, Dallas, TX, Oct. 16–19, 2000.
[4] J. V. Francés, J. Calpe, E. Soria, M. Martínez, A. Rosado, A. J. Serrano, J. Calleja, and M. Díaz, "Application of ARMA modeling to the improvement of weight estimations in fruit sorting and grading machinery," presented at the *25th IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Istanbul, Turkey, June 5–9, 2000.
[5] P. M. Clarkson, *Optimal and Adaptive Signal Processing*. Boca Raton, FL: CRC Press, 1993, pp. 41–42.
[6] C. F. N. Cowan and P. M. Grant, *Adaptive Filters*. Englewood Cliffs, NJ: Prentice-Hall, 1985.
[7] B. Widrow, J. R. Glover, J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, E. Dong, and R. C. Goodlin, "Adaptive noise cancelling: Principles and applications," in *Proc. IEEE*, vol. 63, Dec. 1975, pp. 1692–1716.
[8] E. Soria, J. Calpe, J. Guerrero, M. Martínez, and J. Espí, "An easy demonstration of the optimum value of the adaptation constant in the LMS algorithm," *IEEE Trans. Educ.*, vol. 41, p. 83, Feb. 1998.

Emilio Soria was born in Albacete, Spain, in 1969. He received the B.Sc. degree in physics and the Ph.D. degree in electronic engineering from the Universitat de València, València, Spain, in 1992 and 1997, respectively.

Since 1994, he has been with the Department of Ingenieria Electronica at the University of València, where he is currently an Associate Professor with the Digital Signal Processing Group (GPDS). His research activities include advanced signal processing using neural networks and fuzzy systems.

Javier Calpe (S'90–M'93) received the B.Sc. and Ph.D. degrees, both in physics, from the Universitat de València, València, Spain, in 1989 and 1993, respectively.

Since 1991, he has been with the Department of Ingenieria Electronica at the University of València, where he is an Associate Professor with the Digital Signal Processing Group (GPDS). He holds an industrial patent and has coauthored more than 40 papers in scientific magazines and 100 communications to congresses. He has worked on 16 projects with private companies and eight with public funds, including co-leadering one funded by the European Union (SmartSpectra). His research activities include digital signal processing and its industrial applications and smart sensors.

Jonathon Chambers (S'85–M'85–SM'98) was born in Peterborough, U.K., in March 1960.

He is Professor of Digital Signal Processing at King's College London, London, U.K., and the Director of the Centre for Digital Signal Processing Research. He has acted as the advisor for 18 Ph.D. graduates. He has authored and coauthored more than 150 research publications, and he is the coauthor of the research monograph *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability* (New York: Wiley, 2002). His research interests are in the area of adaptive, blind, and statistical signal processing with applications in telecommunications and intelligent sensors.

He has served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II: ANALOG AND DIGITAL SIGNAL PROCESSING. He is the co-recipent of two Institution of Electrical Engineers (IEE) premium awards. He is a Member of the European Signal Processing Society ADCOM and has served as the Chairman of the U.K. IEE Professional Group on Signal Processing and several conference committees. He is a Chartered Engineer.

Marcelino Martínez received the B.S. and Ph.D. degrees in physics from the Universitat de València, València, Spain, in 1992 and 2000, respectively.

Since 1994, he has been with the Digital Signal Processing Group at the Department of Electronics Engineering, Universitat de València, where he is currently an Associate Professor. He has worked on several industrial projects with private companies (e.g., industrial control, real-time signal processing, and digital control) and with public funds (foetal electrocardiography and ventricular fibrillation). His research interests include real-time signal processing, digital control using digital signal processing and biomedical signal processing, with special interest in developing real-time algorithms for noninvasive foetal electrocardiogram extraction.

Gustavo Camps (M'03) was born in València, Spain, in 1972. He received the B.Sc. degree in physics, the B.Sc. degree in electronics engineering, the M.Sc. degree in physics, and the Ph.D. degree in physics from the Universitat de València, València, Spain, in 1996, 1998, 2000, and 2002, respectively.

He is currently an Associate Professor in the Electronics Engineering Department, Universitat de València. His research interests are neural networks and kernel methods for time-series prediction and biomedical signal processing.

José David Martín Guerrero was born in València, Spain, in 1974. He received the B.S. degree in physics, the B.S. degree in electronics engineering, and the M.S. degree in electronics engineering from the Universitat de València, València, Spain, in 1997, 1999, and 2001, respectively. He is currently working toward the Ph.D. degree in artificial intelligence methods applied to Web mining.

Mr. Guerrero is a Member of the European Neural Network Society.