The modified window correlation technique for estimation of arbitrary time delays in multichannel, multievent systems

Ben-Chung Jan, Juan A. Henriquez, Terry E. Riemer, and Russell E. Trahan

Citation: The Journal of the Acoustical Society of America **93**, 1466 (1993); doi: 10.1121/1.406804
View online: https://doi.org/10.1121/1.406804
View Table of Contents: https://asa.scitation.org/toc/jas/93/3
Published by the Acoustical Society of America

The modified window correlation technique for estimation of arbitrary time delays in multichannel, multievent systems

Ben-Chung Jan,[a] Juan A. Henriquez, Terry E. Riemer, and Russell E. Trahan, Jr.
Department of Electrical Engineering, University of New Orleans, New Orleans, Louisiana 70148

(Received 25 May 1990; revised 20 October 1992; accepted 14 November 1992)

In a previous paper, the so-called window-correlation technique [Callison *et al.*, J. Acoust. Soc. Am. **81**, 1000–1006 (1987)], a method for estimating arbitrary time delays in multichannel, multievent systems, was presented. This method windows a portion of the data record in both channels of a two-channel system, and computes the cross correlation of the data segments. However, the CPU time consumed by this technique becomes prohibitively long for most real time applications. This paper introduces a modification to the above method called the modified window-correlation technique (MWCT) that reduces the CPU time considerably by windowing only one channel and cross correlating the data record in the frequency domain. The simulation results for a two-channel, two-event system show that the MWCT can estimate event delays very accurately down to a signal-to-noise ratio (SNR = mean-square value of signal/mean-square value of noise) of 0 dB.

PACS numbers: 43.60.Gk

INTRODUCTION

The estimation of nonuniform time delays present in multichannel, multievent systems has been proposed by Callison *et al.*[1] as the window-correlation technique (WCT). This method uses the window concept together with the measurement of cross correlation in order to estimate arbitrary event delays very accurately down to a signal-to-noise ratio (SNR) of 0 dB. Moreover, when event delays cannot be identified, the simulation results still produce a good *percentage improvement in alignment* (see Sec. III) when compared to ordinary cross-correlation techniques. Unfortunately, the WCT is computationally expensive.

In this paper, the modified window-correlation technique (MWCT) is introduced. This new method windows out one single channel (instead of both), thus resulting in a considerable reduction in CPU time. However, the price paid by the reduction in computation results in a decay of alignment especially when the SNR drops below 0 dB. This suggests that this modification can only be applied to systems containing a small number of events.

We begin the presentation of the MWCT by first defining a simple model for a two-channel-two event system as follows:

$$x(k) = h_1(k) + h_2(k) + n_1(k),$$
$$y(k) = \alpha h_1(k-\tau_1) + \beta h_2(k-\tau_2) + n_2(k), \quad 0 \leqslant k \leqslant N-1, \tag{1}$$

where x is the sampled data measurement from channel 1 (reference channel), y is the sampled data measurement from channel 2 (processing channel), h_1, h_2 are different nonoverlapping signal events in both channels, k is the sample time, τ_1, τ_2 are arbitrary time delays, α, β are attenuation factors, n_1, n_2 are additive Gaussian white noise (AGWN), and N is the data record length. Note that τ_1 and τ_2 are assumed to be arbitrary; however, we do assume that the delayed versions of h_1 and h_2 do not overlap in channel 2.

I. REVIEW OF THE WINDOW-CORRELATION TECHNIQUE

The problem considered here is to estimate τ_1 and τ_2; i.e., the time delays of the signal events originating in channel 1 and also appearing as delayed versions in channel 2.

The following description is based on the WCT of Ref. 1 with some notation changed. To successfully estimate an event delay, the first step in the WCT consists in isolating a data segment from both channels by using an appropriate window. Define the windowed channels as:

$$x_w(k) = x(k)w(k), \quad y_w(k) = y(k)w(k), \tag{2}$$

where

$$w(k) = 1, \quad 0 \leqslant k \leqslant M-1$$
$$= 0, \quad M \leqslant k \leqslant N-1, \tag{3}$$

and M is the window length.

The autocorrelation of $x_w(k)$ for a time shift λ is denoted by $R_{xxw}(\lambda)$, and can be calculated by

$$R_{xxw}(\lambda) = \frac{1}{N} \sum_{k=0}^{N-|\lambda|-1} x_w(k) x_w(k+\lambda). \tag{4}$$

Similarly, the cross correlation of $x_w(k)$ and $y_w(k)$ is denoted by $R_{xyw}(\lambda)$, and can be calculated by

$$R_{xyw}(\lambda) = \frac{1}{N} \sum_{k=0}^{N-|\lambda|-1} x_w(k) y_w(k+\lambda). \tag{5}$$

Note that both $R_{xxw}(\lambda)$ and $R_{xyw}(\lambda)$ can be estimated by a fast Fourier transform (FFT) algorithm in order to decrease computation time.

[a] Deceased.

 0001-4966/93/031466-05$06.00

To measure how close $R_{xxw}(\lambda)$ remains to $R_{xyw}(\lambda)$, the squared error criterion is used as follows:

$$\epsilon^2 = \sum_{\lambda=0}^{N-1} [R_{xxw}(\lambda) - R_{xyw}(\lambda)]^2 . \qquad (6)$$

The WCT proceeds by using Eq. (6) to obtain the squared error value for the first window position, we denote the error by ϵ_1^2. The next step is to slide the window of channel 2 by one sample and compute Eqs. (5) and (6) again to find a new squared error ϵ_2^2. The procedure is repeated moving the window through the entire channel 2 in order to obtain a sequence of squared errors,

$$z = (\epsilon_1^2, \epsilon_2^2, ..., \epsilon_N^2) . \qquad (7)$$

The minimum value in Eq. (7) implies that the corresponding windowed segments have the most correlated relation. Therefore, the delay in channel 2 of the particular windowed segment of channel 1 can be estimated by searching out this minimum value and computing the distance, d_1, from the position of this minimum value to the position where the windowed segment of channel 1 starts.

Next, the WCT slides the window of channel 1 by one sample and repeats the above procedure (obtaining a new sequence z) until the window has moved through the entire channel 1. The result is a sequence of numbers $(d_1, d_2, ...)$ which represents the time delays of the corresponding windowed segments. These segment delays are then used to find the event delays by using another algorithm, details of which are given in Ref. 1.

II. THE MODIFIED WINDOW-CORRELATION TECHNIQUE

From the above description, the WCT slides the window in both channels separately and performs several calculations at every new position of each window. Assuming that the window slides N times in each channel, the method then computes N^2 cross correlations as well as N autocorrelations, thus consuming a considerable amount of CPU time. In order to significantly reduce this time, we introduce the modified window-correlation technique (MWCT) as described in this section.

The idea for the modification of the WCT comes from the fact that cross-correlation methods can be used to detect uniform delays very accurately under very noisy conditions. If we only window out a data segment of channel 1 and calculate the cross correlation between this windowed segment and the entire data record from channel 2, the calculation can be considered as the delay estimation of this windowed segment surrounded by heavy noise. Therefore, in the case in which an event is inside a windowed segment, a successful estimation can be obtained.

The MWCT begins by windowing channel 1 in order to obtain the function $x_w(k)$ as in Eq. (2). An FFT algorithm is then used to compute its spectrum as

$$X_w(n) = \sum_{k=0}^{N-1} x_w(k) e^{-j2\pi nk/N}, \quad n = 0,1,...,N-1 . \qquad (8)$$

An FFT algorithm is also used to compute the spectrum of the entire channel 2 as

$$Y(n) = \sum_{k=0}^{N-1} y(k) e^{-j2\pi nk/N}, \quad n = 0,1,...,N-1 . \qquad (9)$$

The cross correlation of $x_w(k)$ with $y(k)$ can then be calculated by first computing the product of the spectrum of each signal as

$$S_{x_w y}(n) = \overline{X}_w(n) Y(n), \quad n = 0,1,...,N-1 , \qquad (10)$$

where $\overline{X_w}$ is the complex conjugate of X_w, and then taking the inverse discrete Fourier transform (DFT) by

$$R_{x_w y}(k) = \frac{1}{N} \sum_{n=0}^{N-1} S_{x_w y}(n) e^{j2\pi nk/N}, \quad k = 0,1,...,N-1 . \qquad (11)$$

From Eq. (11), the position with the maximum cross-correlated value is considered the estimated delay d_1 of this windowed segment. Figure 1 shows an example of the cross-correlated values when the fixed window of channel 1 starts at position 1. As seen from this figure, the maximum cross-correlated value is at position 31; thus the estimated delay of this specific windowed segment is 30. The window is then moved by one sample and the procedure is repeated in order to obtain another estimated delay d_2. After moving the window through the entire channel 1, a sequence of segment delays $(d_1, d_2, ...)$ is found. These numbers are then used to estimate the event-delays by implementing Algorithm 3, *Identification of Events*, as discussed in Ref. 1.

Based on the above description, we present the following algorithm to estimate the time delay of every windowed segment.

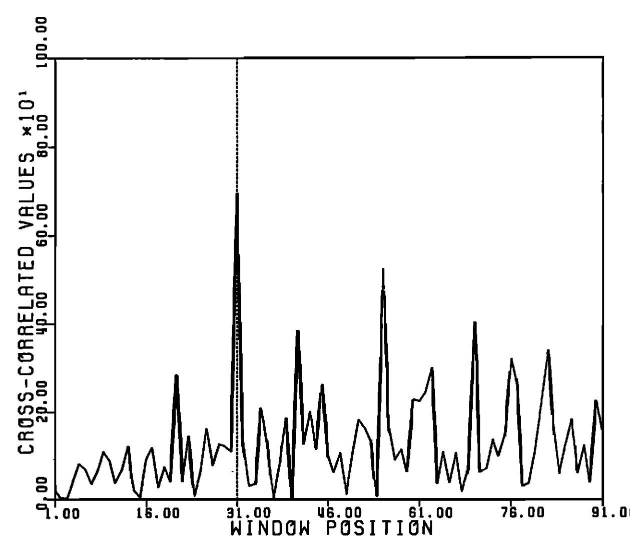

FIG. 1. Example showing the maximum cross-correlated value obtained by implementing Eq. (10). This maximum value occurs at position 31 when the window starts at position 1; therefore, the estimated window segment delay is 30. The SNR is 40 dB.

Algorithm: Estimate Segment Delays

Input:

$x \equiv$ signal in the reference channel

$y \equiv$ signal in the processing channel

$N \equiv$ number of data points (record length)

$M \equiv$ window length

Output:

Delay $(i) \equiv$ set of delays

Steps:

Step 0: Set $i = 0$.

Step 1: Compute the spectrum of the entire channel 2, $Y(n)$, using Eq. (9).

Step 2: Window channel 1 starting at i to obtain:

$$x_w(m) = x(m), \quad i \leqslant m \leqslant W + i - 1$$
$$= 0, \quad \text{otherwise}.$$

Step 3: Compute the spectrum of the windowed segment $x_w(m)$, $X_w(n)$, using Eq. (8).

Step 4: Compute $S_{x_w y}(m) = \overline{X_w}(m) Y(m)$.

Step 5: Compute the inverse DFT of $S_{x_w y}(m)$ to obtain $R_{x_w y}$ using Eq. (11).

Step 6: Search for the value Imax where the maximum cross correlation occurs:

compute delay $(i) = \text{Imax} - i$.

Step 7: Set $i = i + 1$:

If $i \leqslant N - M + 1$, go to step 2;

else, go to step 8.

Step 8: Output delay(i), $i = 0, 1, ..., N - M + 1$, into a file.

Step 9: Stop.

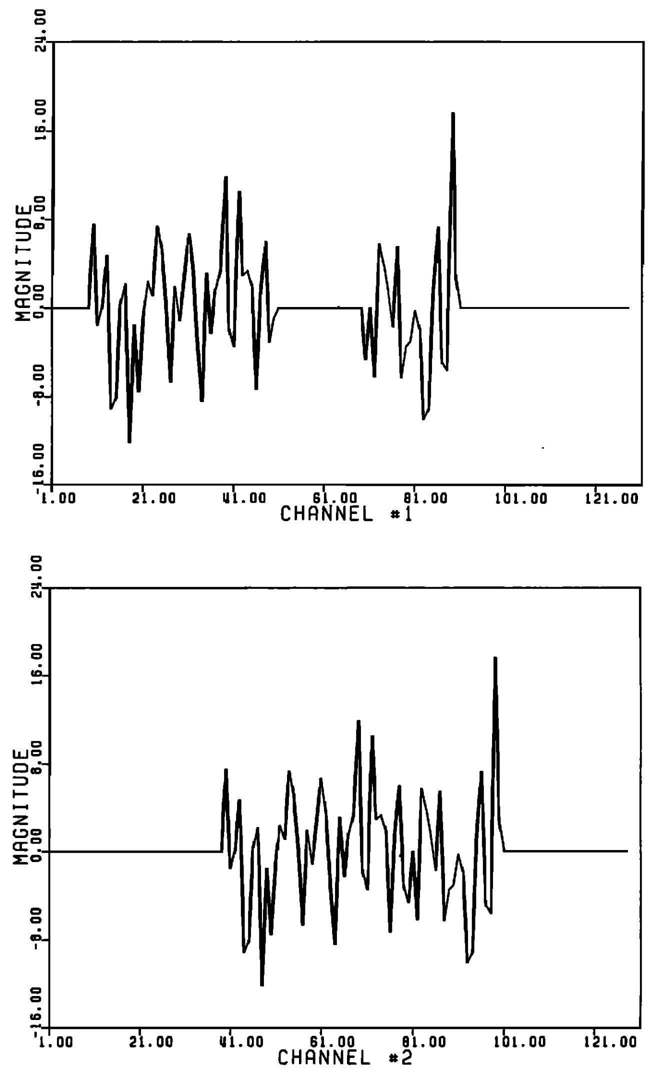

FIG. 2. (a) Synthetic data record of channel 1. This record consists of 128 samples containing two events, the first event occurs from samples 10 to 50 while the second one occurs from samples 71 to 90. (b) Synthetic data record of channel 2. This record consists of 128 samples containing two events, the first event occurs from samples 40 to 80 while the second one occurs from samples 81 to 100.

III. SIMULATION RESULTS

The algorithm described in the previous section, has been successfully implemented on a VAX-8600 cluster computer system. Our primary intention for testing the capability of the MWCT is not only to see whether it can accurately estimate the event-delays, but also to see whether this method results in a reduced CPU time when compared to the WCT. Thus the same synthetic data implemented in Ref. 1 are used as the test signals for the MWCT. These signals consist of two uncorrelated Gaussian-white zero-mean events as shown in Fig. 2. Note that there are 128 sample points in each channel. In channel 1(2) the first event occurs from sample 10(40) to sample 50(80) and the second event occurs from sample 71(81) to sample 90(100). Delays are chosen in channel 2 to be 30 sample points for the first event and 10 sample points for the second event with respect to channel 1. Moreover, there are no overlaps in time between these events.

The channel signals are corrupted by different AGWNs (for testing at several values of SNR). The system SNR is defined as:

$$\text{SNR}_{\text{dB}} = 10 \log(\overline{S^2} / \overline{N^2}), \tag{12}$$

where $\overline{S^2}$ is mean-square value of the signal and $\overline{N^2}$ is mean-square value of the noise. Note that these values can be easily calculated from the autocorrelation function.[2]

After running the *Estimate Segment-Delays* algorithm of the MWCT described above and the WCT separately, the output files (i.e., Delay(i)) are used as the input to the *Identify Event-Delays* algorithm. The estimated event-delays for several values of SNR are shown in Table I. As can be observed from this table, both techniques can estimate the two nonuniform event delays very accurately down to a SNR of 0 dB.

The advantage of using the MWCT over the WCT is

TABLE I. Identification of event delays for several values of SNR (actual delays: $\tau_1 = 30$, $\tau_2 = 10$ sample points). The asterisk indicates that the program cannot identify the delay and $\hat{\tau}$ indicates estimate of τ.

SNR (dB)	MWCT $\hat{\tau}_1$	MWCT $\hat{\tau}_2$	WCT $\hat{\tau}_1$	WCT $\hat{\tau}_2$
100	30.000	9.972	30.000	10.000
800	30.000	9.972	30.000	10.000
60	30.000	9.972	10.000	
40	30.000	9.972	30.000	10.000
20	30.000	10.000	29.943	10.000
10	30.000	10.000	29.846	10.000
5	30.000	10.000	29.761	10.000
0	30.000	10.000	30.000	10.476
− 5	*	*	*	*
− 10	*	*	*	*

shown by comparing the average CPU time needed to run the *Estimate Segment-Delays* algorithm. While the WCT's CPU time is 7 min and 54.56 s, Ref. 1, that of the MWCT is 3.43 s (over 130 times faster). Note that the CPU time needed to run the *Identify Event-Delays* algorithm is not included since we wish to compare the similar components of each technique. Moreover, the *Identify Event-Delays* algorithm in the WCT used an average CPU time of less than 0.2 s which is considered negligible.

In Ref. 1, a parameter called the *percentage improvement in alignment* is used in order to test the accuracy of the technique. This parameter is obtained by computing the squared error between the original two channel signals at every position (SE1), using the estimated delays to align the two channels by shifting the signal of channel 2 to the calculated positions, and then computing the squared error between them (SE2) again. These squared error values, SE1 and SE2, are used to find the *percentage improvement in alignment* as

$$\% \text{ Improvement in alignment} = [(\text{SE1} - \text{SE2})/\text{SE1}] \times 100 . \quad (13)$$

As can be expected, the alignment and the accuracy of the estimated delay will increase as the SNR increases. If the estimated delay in every position is correct, and there is complete alignment, SE2 is only the squared error of the noise in both channels and the percentage improvement should be close to 100%. However, when the SNR decreases, the amount of noise increases and perfect alignment is not achieved.

Table II shows the results of this test for both techniques for several values of SNR. As we can see, there is a high percentage improvement in alignment until the SNR reaches 10 dB. When the SNR falls from 10 to 0 dB, the WCT provides better results than the MWCT, but this difference is subtle. After the SNR falls below 0 dB (where both techniques cannot successfully estimate the event delays), the WCT provides the highest percentage improvement in alignment.

An important situation occurs when the SNR is below 0 dB and the alignment of the MWCT is so low that it cannot product any improvement. In fact, this result should be predictable since the MWCT takes the cross correlation between one windowed segment of channel 1 and the entire data record from channel 2. The method can then be considered as the estimation of the windowed segment delay in

TABLE II. Percentage improvement in alignment for several values of SNR.

SNR(dB)	MWCT(%)	WCT(%)
100	97.033	98.458
80	97.033	98.458
60	97.033	98.457
40	97.033	98.397
20	96.340	98.207
10	91.747	94.011
5	83.197	88.958
0	67.666	79.868
− 5	14.427	72.908
− 10	8.856	70.630

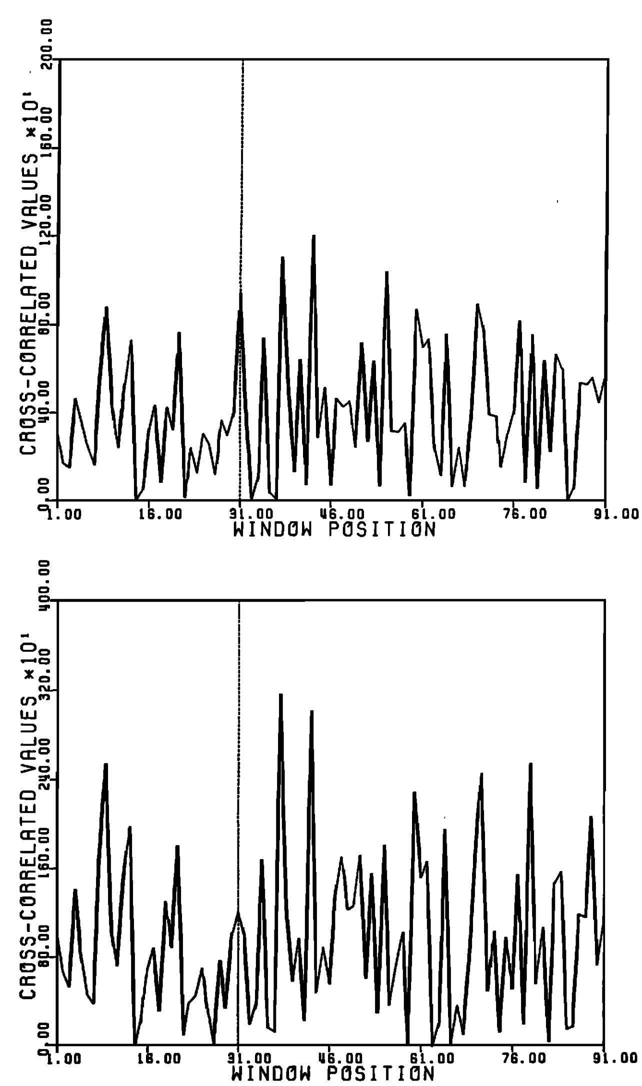

FIG. 3. (a) Plot of cross-correlated values when SNR = − 5 dB. The figure shows the local maxima surrounding the estimated segment delay at position 31. (b) Plot of cross-correlated values when SNR = − 10 dB. The figure shows the increase in local maxima surrounding the estimated segment delay at position 31.

channel 1 surrounded not only by the noise in the entire data record from channel 2, but also by the heavily corrupted data outside the window segment in channel 1. The WCT uses a window in both channels and estimates the delay between the windowed segments. Thus, in estimating the same system under the same conditions, the MWCT will "degrade" more quickly than the WCT. However, the results presented in Table I do not show this point, mainly because the synthetic data only contain two separate events. Consider the case when both channels contain several events; for example, assume each channel has ten events. To succeed in the estimation, the fixed window length should not be chosen to cover two events at one time. Assuming the chosen window is of appropriate length, we can immediately see that the MWCT will not do very well under this condition because the data points outside the windowed segment are increased considerably; however, the WCT will not be affected by increasing the number of events because it only considers the data segment inside the window. Thus, the MWCT should be used in systems containing a small number of events.

Finally, we have shown that both techniques cannot estimate the event delays whenever the SNR drops below 0 dB. Nonetheless, even below this level of SNR some information is still available. Note that the segment delay is estimated by finding the global maximum (or minimum) correlated measurement; however, when the SNR drops below 0 dB, there is a local maximum (or minimum) at the delay position. Figure 3 depicts this situation. The plots in this figure consist of the estimated cross-correlated values obtained by using the MWCT when the window of channel 1 is at position 1, and the SNR is -5 dB [Fig. 3(a)] and -10 dB [Fig. 3(b)]. As can be observed from this figure, the value at position 31 (where the delay of this windowed segment occurred) is a local maximum. Thus the MWCT is still capable of estimating this cross-correlated measurement even when there are several local maxima located elsewhere that are greater than this value. Also note that lowering the SNR [Fig. 3(b)] increases the local maxima surrounding this value.

IV. CONCLUSIONS

A modification of the window-correlation technique (WCT) is proposed for the estimation of nonuniform time delays in multichannel, multievent systems. The results presented here indicate that the modified window-correlation technique (MWCT) can estimate the event delays very accurately in an environment up to 0 dB while reducing the CPU time by a factor of over 130 over the WCT. This reduction in CPU time is the result of using a window in one channel and computing the FFT between this windowed segment and the entire data record of the other channel in order to obtain their cross correlation. It is also important to note that, for best results, the MWCT is recommended for systems that contain a small number of events.

[1] J. D. Callison, T. E. Riemer, and R. E. Trahan, Jr., "Estimation of Arbitrary Time Delays of Multi-Channel Synthetic Data," J. Acoust. Soc. Am. **81**, 1000–1006 (1987).

[2] G. Cooper and C. McGillen, *Probabilistic Methods of Signal and System Analysis* (CBS College, New York, 1986), 2nd ed., p. 245.