Applying the Accumulation of Cross-Power Spectrum Technique for Traditional Generalized Cross-Correlation Time Delay Estimation

Radu-Sebastian Marinescu[*,#], Andi Buzo[*], Horia Cucu[*], Corneliu Burileanu[*]

[#]Research & Development, Rohde & Schwarz Topex

[*]SpeeD Laboratory, University Politehnica of Bucharest

Bucharest, Romania

radu-sebastian.marinescu@rohde-schwarz.com, {andi.buzo, horia.cucu, corneliu.burileanu}@upb.ro

Abstract—**In many real time applications, time delay estimation requires a special solution. Despite the various approaches, which were proposed over the years, the topic remains hot for digital signal processing because of its large field of applications and implementation forms. Among different classes of methods for this issue, general cross-correlation method is wildly used. It offers good results and does not need an adaptation time, like those based on adaptive filtering. In this paper, we make a survey and compare the most popular generalized cross-correlation methods. We extend the analysis, by applying the accumulation of cross-power spectrum technique, for all well known generalized cross-correlation methods. The comparisons are provided by detailed numerical and simulation analysis, using several metrics. Based on the accuracy rate, error rate, standard deviation of relative error and computing time we provide new considerations for traditional generalized cross-correlation methods.**

Keywords - Time Delay Estimation, General Cross-Correlation, Accumulated Cross-Power Spectrum.

I. INTRODUCTION

In this paper, we continue to evaluate the performances of our recently proposed time delay estimation methods. This work pushes further the analysis done in [1]-[3] for time delay estimation (TDE). Despite the various techniques developed over the years, the topic continues to be interesting. As technology evolved, more and more applications demanded a real time solution for time delay estimation. For echo canceling, acoustics, radar and sonar localization, seismic and medical processing, pattern detection and speech enhancement, scientists are still looking to improve the existent solutions. However, the variety of TDE applications, implementation aspects and proper constraints, inhibit the design of a unique solution. Instead, various approaches have been developed based on application specific aspects [1].

The various approaches for TDE can be grouped into three categories: a) *generalized cross-correlation* (GCC), b) least-mean squares (LMS) adaptive filtering [4]-[10], and c) adaptive eigenvalue value decomposition (AEVD). Based on the specific aspects required by an application, an optimal solution has to be chosen. As showed in [11] by Benetsy, AEVD technique offers an efficient solution for audio applications from reverberant environment. The adaptive filtering methods have a different approach. This leads to a high accuracy results, which need an adaption time. This solution can be very effective for some applications, but for real time systems the adaptation time makes them unusable. For the last ones, an optimal solution is represented by the generalized cross-correlation methods. They provide fast results, keeping also an acceptable accuracy level.

The main contributions of this paper are multiple. We provide an in-depth analysis of the previous and proposed methods by comparing them from the accuracy and processing speed points of view. We perform a new evaluation for the most used GCC methods and extend the accumulating cross-power spectrum scheme to all well known GCC methods, for a deeper evaluation. Finally, we show that our recently proposed methods, for multiple frames TDE, outperform the previous GCC approaches, offering a lower computation time and a higher accuracy rate even at low signal-to-noise ratios.

The rest of the paper is organized as follows. In the next section, we review the related work over the years. Section II also contains the description of our recently proposed methods. Section III is reserved for experimental results and analysis discussions, grouped in different parts: A) experimental setup, B) calibration of the proposed methods, and C) extended evaluation for accuracy and processing time of all presented methods. Finally, the main conclusions and further work are addressed in Section IV.

II. RELATED WORK FOR GENERALIZED CROSS-CORRELATION TIME DELAY ESTIMATION

For two signals $y_1(t)$ and $y_2(t)$, which are two noisy and delayed versions of the same transmitted signal $x(t)$, time delay estimation aims at finding the relative delay between them. Among the various developed approaches to TDE, the most popular and time-efficient method remains the one based on the cross-correlation of the two signals. In 1976, Knapp and Carter proposed in [12] the generalized cross-correlation methods. They pointed out that a common method of determining the time delay is to compute the cross-correlation function:

$$R_{y_1y_2}^{g}(\tau)=E\left[y_1(t)\cdot y_2(t-\tau)\right] \tag{1}$$

where E denotes expectation. The argument τ that maximizes (1) provides an estimation of delay.

The cross-correlation between $y_1(t)$ and $y_2(t)$ is related to the cross-power spectral density function by the well known Fourier transform relationship:

$$R_{y_1y_2}^{g}(t)=\int_{-\infty}^{\infty}G_{y_1y_2}(f)\cdot e^{j2\pi ft}df \tag{2}$$

To improve the accuracy of delay estimation, a pre-filtering of the inputs is necessary before calculating the cross correlation. When signals $y_1(t)$ and $y_2(t)$ have been filtered with filters having transfer functions $H_1(f)$ and $H_2(f)$ the cross power spectrum between the filter outputs is given by:

$$G_{y_1y_2}^{g}(f)=H_1(f)\cdot H_2^{*}(f)\cdot G_{y_1y_2}(f). \tag{3}$$

Therefore, the generalized cross-correlation between $y_1(t)$ and $y_2(t)$ is:

$$R_{y_1y_2}^{g}(t)=\int_{-\infty}^{\infty}\Psi(f)\cdot G_{y_1y_2}(f)\cdot e^{j2\pi ft}df \tag{4}$$

where:

$$\Psi(f)=H_1(f)\cdot H_2^{*}(f) \tag{5}$$

and denotes the general frequency weighting [12].

Over the years, different weighting functions were proposed to improve the estimation process of the basic cross-correlation. In Table I, we present the various well known weighting functions, used in this work for a detailed analysis, where G_{y1y1} and G_{y2y2} are auto power spectrum of the noisy signals and $\gamma_{y_1y_2}^{2}(f)$ is the signal's coherence function.

$$G_{y_1y_1}(f)=\int_{-\infty}^{\infty}R_{y_1y_1}(t)\cdot e^{j2\pi ft}df \tag{6}$$

$$G_{y_2y_2}(f)=\int_{-\infty}^{\infty}R_{y_2y_2}(t)\cdot e^{j2\pi ft}df \tag{7}$$

$$\gamma_{y_1y_2}^{2}(f)=\frac{\left|G_{y_1y_2}(f)\right|^2}{G_{y_1y_1}(f)\cdot G_{y_2y_2}(f)} \tag{8}$$

For the normal Cross-Correlation (CC) the weighting function $\Psi(f)$ is 1. This is the basic and the fastest computing GCC, because it has no weighting operations.

The Eckart filter derives its name from work done in this area in [13], published in 1951. It maximizes the deflection criterion, i.e., the ratio of the change in mean correlator output due to signal present to the standard deviation of the correlator output due to noise alone [12].

TABLE I. GCC WEIGHTING FUNCTIONS

GCC name	Weighting function
CC	1
Eckart	$\dfrac{\left\|G_{y_1y_2}(f)\right\|}{\left[G_{y_1y_1}(f)-\left\|G_{y_1y_2}(f)\right\|\right]\cdot\left[G_{y_2y_2}(f)-\left\|G_{y_1y_2}(f)\right\|\right]}$
ROTH	$1/G_{y_1y_1}(f)$
HT (ML)	$\dfrac{\left\|\gamma_{y_1y_2}(f)\right\|^2}{\left\|G_{y_1y_2}(f)\right\|\cdot\left[1-\left\|\gamma_{y_1y_2}(f)\right\|^2\right]}$
SCOT	$1/\sqrt{G_{y_1y_1}(f)\cdot G_{y_2y_2}(f)}$
CSP (PHAT)	$1/\left\|G_{y_1y_2}(f)\right\|$
CSP-m	$1/\sqrt[m]{G_{y_1y_1}(f)\cdot G_{y_2y_2}(f)}$
HB	$\left\|G_{y_1y_2}(f)\right\|/G_{y_1y_1}(f)\cdot G_{y_2y_2}(f)$
Wiener	$\left\|\gamma_{y_1y_2}(f)\right\|^2$
ρ-CSP	$1/\left\|G_{y_1y_2}(f)\right\|^{\rho}$
ρ-CSPC	$\dfrac{1}{\left\|G_{y_1y_2}(f)\right\|^{\rho}+\min\left[\gamma_{y_1y_2}(f)^2\right]}$

Twenty years later, in 1971, Roth proposed a new processor in [14]. It has desirable effect of suppressing those frequency regions where G_{y1y1} is large and the estimate of G_{y1y2} is more likely to be in error [12].

In the same year it was proposed another weighting function, the HT processor, by Hannan and Thomson. This assigns greater weight in regions of frequency domain where the coherence is large [15]. In [12], it was shown that HT processor is a maximum likelihood (ML) estimator for time delay under usual conditions. Under a low signal-to-noise ratio restriction, the HT processor is equivalent to Eckart prefiltering and cross-correlation.

The SCOT (Smoothed Coherence Transform) was introduced by Carter, Nuttall and Cable in 1973, to reduce the influence of a strong tonal [16]. However, for smoothed signal and noise spectra, Hassab and Boucher [17][18] have noted that the additional SCOT weighting function has weakened the performance of the basic cross correlator, while other functions have improve it.

Phase Transform (PHAT) or Cross-power Spectrum Phase (CSP) was developed purely as an ad-hoc technique to avoid spreading of the above two presented operators. Ideally, PHAT does not suffer the spreading that other processors do. Also, because it weights G_{y1y2} as the inverse of $|G_{y1y2}|$, the errors are accentuated where signal power is smallest [12].

In 1979, the HB processor was presented by Hassab and Boucher. It is similar to SCOT in that, for highly dynamic spectra, in addition to suppressing the cross-spectral estimate in frequency regions of low signal-to-noise ratio, high signal-to-noise ratio regions are also suppressed in attempt to reject strong tonals in the observations [19].

The Wiener processor was proposed in 1985 by Hero and Schwartz. Based on channel's linearity it tries to estimate the original signal from the observation $y_1(t)$ and channel output signal from $y_2(t)$, by minimizing the mean-square errors. In this way, given the channel characteristics, the solution results in Wiener filters, which yield the Wiener weighting function [20].

In 1996 it was presented a new weighting function, for acoustic localization, by Rabinkin et al., the ρ-Cross-power Spectrum Phase (ρ-CSP). It adds to the normal CSP the tuning parameter ρ (with values between 0 and 1) as a whitening parameter, which discards the non-speech portion (below 200Hz) of the CSP [21].

Relatively recently, in addition to the above work, in 2009 was proposed ρ-Cross-power Spectrum Phase with Coherence (ρ-CSPC), by Shean and Liu. The presence of the minimum of the coherence function in the weighting function helps to reduce errors for relatively small energy signals [22].

For the above presented GCC methods, an implementation block diagram is presented in Fig. 1. First, the analysis frames of input signals are converted into frequency domain using the Fast Fourier Transform (FFT) block. Then, the cross-power spectrum is computed by multiplications of resulted spectra and weighting function. Going further, the generalized cross-correlation is obtained through an Inverse Fast Fourier Transform (IFFT). The final step consists in finding the argument, which maximizes GCC and estimating the delay. This is the basic way to obtain an estimation of delay.

For a large window with L samples, FFT's complexity order is $O(L\cdot logL)$, with L a power of 2. Because these consume important processing time, it is natural to search for solutions, which increase the computing speed. A way to achieve this is to divide the larger analysis window into smaller frames, as it is shown in Fig. 2. Thus, the larger analysis window of L samples is divided in K smaller frames, of n samples each. If the length of the frame l is also a power of 2, then the new complexity order is $O(K\cdot l\cdot logl) = O(L\cdot log(L/K))$, which needs a smaller processing time. For each smaller analysis frame, the partial estimated delay is obtained similarly as in Fig. 1. Then, the final estimated delay yields as the average of all partial estimated delays. In this way, it is also easier to estimate a variable delay. This approach is recommended especially when the estimated delay is expected to be considerable less than the length of the larger window.

An alternative way for the above multi-frame approach is accumulated Cross-power Spectrum Phase (acc-CSP), proposed in 2006, by Matassoni and Svaizer [23]. It accumulates the cross-power spectrum over multiple frames in frequency domain, as showed in Fig. 3. This scheme leads to a new computing time decrease, because the number of IFFT and peak detector is reduced to 1. In frequency domain it can be expressed as follows:

$$G_{acc-CSP}(f) = \sum_{k=1}^{K} \frac{G_{y_1 y_2,k}(f)}{\left|G_{y_1 y_2,k}(f)\right|}, \tag{9}$$

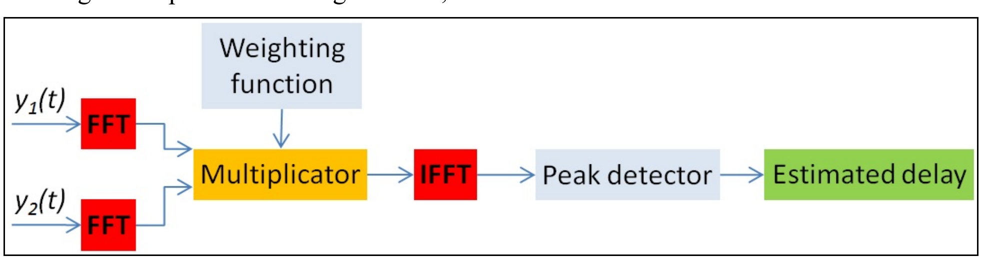

Figure 1. Block diagram for a single frame GCC implementation

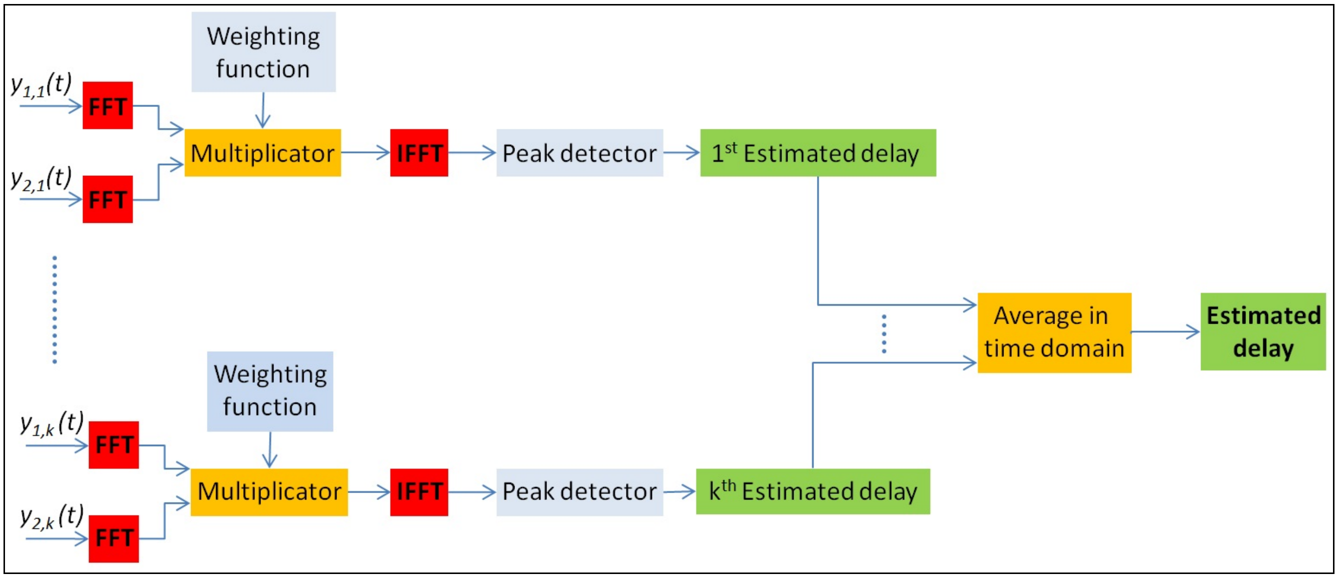

Figure 2. Block diagram for multiple frames GCC with time domain average estimation

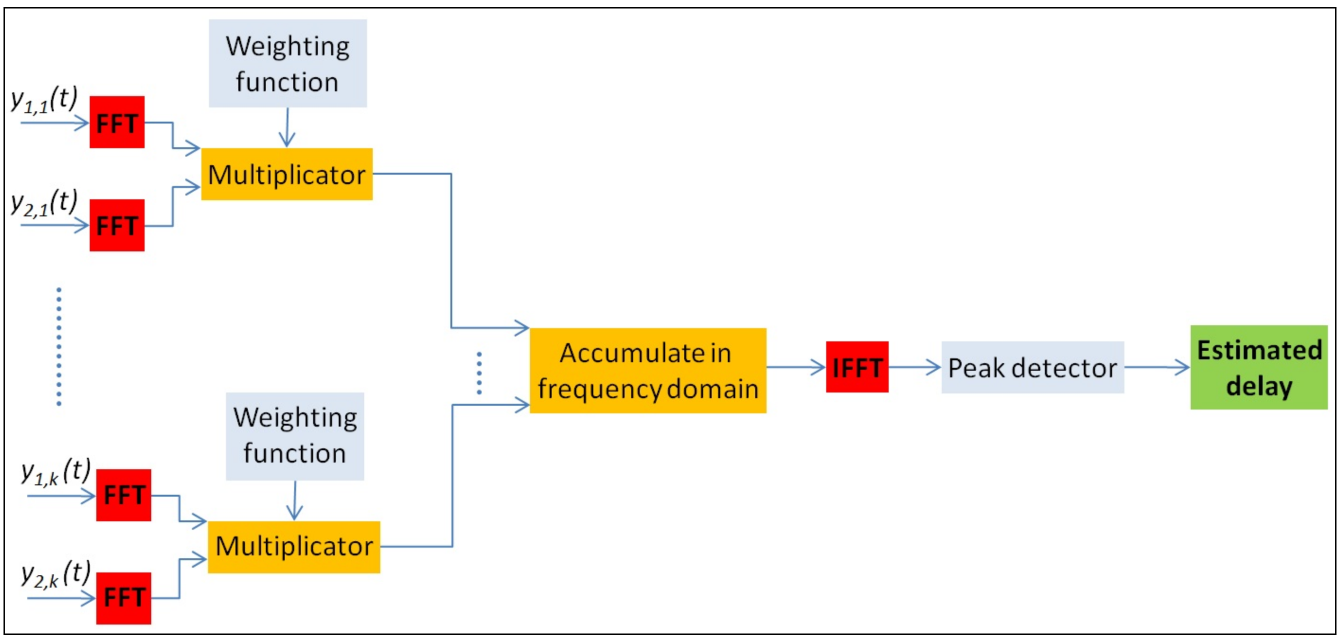

Figure 3. Block diagram for accumulating multiple frames GCC in frequency domain

where K represents the number of accumulated frames. Beside the reduced computational complexity, the *acc-CSP* method enhances the estimation by intrinsic integration for fixed delay during the analysis window [23].

The acc-CSP method proposes the accumulation scheme of cross power spectrum in frequency domain, increasing the computation speed. Methods based on the approach presented in Fig. 2 computes the TDE as the average of all partial estimated delays of each frame from the analysis window. In this way, for K frames, the number of total FFT operations is equal to $3xK$, because two FFT are used to transform the signals from time to frequency domain, and then one IFFT is used on the cross power spectrum to return in the time domain, for each frame. Instead, the accumulation scheme from Fig. 3 is faster because it does not calculate any partial TDEs. Because the cross-power spectrum averaging is computed in frequency domain, only one estimate will result, for any number of K frames. Thus, only one IFFT is needed for the final estimation and $2xK$ FFTs for time to frequency transformations. This leads to a total number of $2xK + 1$ FFT for the accumulating scheme, which is less than the $3xK$ FFT needed by previous methods [3]. Also, a small increase to the computation speed is due to the reduction to only one peak detector call.

Based on ρ-CSPC and ρ-CSP, in combination with the accumulated cross-power spectrum scheme, we recently proposed two new methods in [2] the new *accumulated ρ-Cross Power Spectrum Phase with Coherence* (*acc-ρCSPC*) and *accumulated ρ-Cross Power Spectrum Phase* (*acc-ρCSP*). In frequency domain they are expressed as

$$G_{acc-\rho CSPC}(f)=\sum_{k=1}^{K}\frac{G_{y_1y_2,k}(f)}{\left|G_{y_1y_2,k}(f)\right|^{\rho}+\min\left[\gamma^2_{y_1y_2,k}(f)\right]}. \quad (10)$$

and

$$G_{acc-\rho CSP}(f)=\sum_{k=1}^{K}\frac{G_{y_1y_2,k}(f)}{\left|G_{y_1y_2,k}(f)\right|^{\rho}} \quad (11)$$

In this way, for (10) it is possible to take advantage of both ideas (ρ-CSPC and accumulation scheme). Its effectiveness was proven by experimental results from [2], which showed a better accuracy even for low signal-to-noise ratios (SNR).

The new approach, summarized by (10), leads to faster computations compared to its previous methods, because it uses the accumulating scheme, presented in Fig. 3. It can also provide better results in unfavorable conditions for smaller frame sizes. Beside this, emphasis of speech regions from the spectrum is achieved by the whitening parameter (ρ), which reduces, at the same time, the impact of noise outside the speech region. For parts of the signal with small energy, the addition of the minimum coherence function limits the effect of a very small denominator [1][21][22].

The approach from (11) appeared as a faster variant of (10) for applications where relatively small energy signals are not encountered. In these conditions, the minimum coherence function can be omitted from (10). Thus, there is no need to compute the coherence function and to find its minimum, resulting a substantial computing time decrease.

As shown in [2], a high accuracy rate of TDE with *acc-ρCSPC* and *acc-ρCSP* is achieved if a calibration step is performed first. This procedure will be detailed and commented in the next section.

Over the years, several other studies discussed the details about time delay estimation based on generalized cross correlations, like in [24]-[31]. In this paper, we extend the analysis with the accumulating cross-power spectrum scheme, not only to our recently proposed methods, but also to the others well known GCC. We will apply the accumulation of cross-power spectrum technique (Fig. 3) to the traditional GCC functions from Table I, which are implemented as in Fig. 2 in all current applications.

III. EXPERIMENTAL RESULTS AND DISCUSSIONS

In several previous papers [1]-[3] we proposed and evaluated acc-ρCSPC and acc-ρCSP. These methods derived from their primitive forms, ρCSPC [22] and ρCSP [21], at which we applied accumulation of cross-power spectrum in frequency domain [23]. To extend our research on this topic, in this work we apply accumulating scheme to all well known GCC functions. To the best of our knowledge, this technique was not presented in any other previous study.

A. Experimental Setup

Evaluation tests were performed in Matlab and C language. The input signals were taken from Noizeus data base corpus [32]. It contains 30 sentences (produced by three male and female speakers at a sampling rate of 8 kHz) corrupted by 8 different real-world noises (suburban train, babble, car, exhibition hall, restaurant, street, airport and train station noise), from the AURORA database [33] at 4 different SNRs (0, 5, 10 and 15 dB).

We used four metrics in our experiments: accuracy and error rate, standard deviation of the relative error and computing time. We define the accuracy rate as the ratio between the number of correctly estimated delays and total number of estimations performed (we imply that a correct estimation as one where the estimated delay is equal to the reference delay in terms of samples). Complementary to this we define the error rate as the ratio between the number of incorrectly estimated delays and total number of estimation performed.

For the first three metrics we used Matlab implementations. The forth metric, is the processing time, for which we evaluated the C implementations, compiled with gcc-4.7.3, on a machine with an Intel "Core i5" processor.

B. Calibration stage

It is easy to observe that for ρCSP and ρCSPC, the whitening factor ρ is not defined yet. In [22] it is used with values between 0.78 and 0.9. Also, ρ parameter requires particular attention because it characterizes our recently proposed methods. In our approach, we need to maximize the accuracy rate for accumulated cross-power spectrum versions, so we have to find the optimum value for ρ. Thus, we divided the Noizeus database in two parts, like in [2]. 50% of the sentences were used for *development* and the other 50% were used for the *evaluation.* The signal pairs chosen for alignment cover all the combinations of noise types: $C_2^8 = 28$. By using 4 different SNR levels and 5 artificially introduced delay values (5, 10, 25, 50 and 100 ms), the total number of test pairs becomes 28x15x4x5=8400.

In order to obtain efficient results, it is also important to set adequately the methods parameters (number of frames, frame size, overlap factor and ρ), which influence the accuracy and error rate of the *acc-ρCSPC* and *acc-ρCSP*. The first three parameters have to be chosen based on the nature of the application, making a trade-off between computing time, accuracy and fast response of the system. For this operation, here we used 4 averaging frames of 1024 samples each, and an overlap factor of 25%.

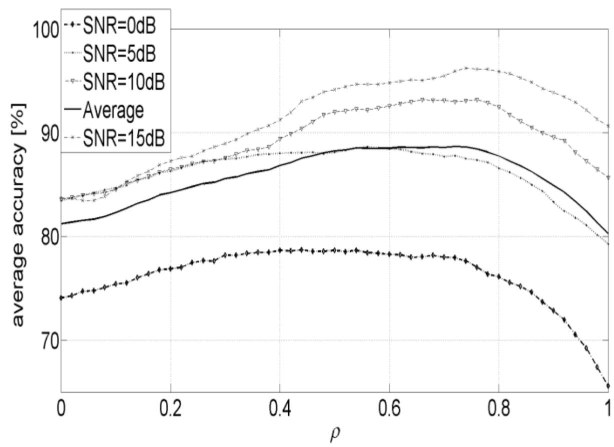

Figure 4. The influence of SNR and ρ over the *acc-ρCSPC* accuracy

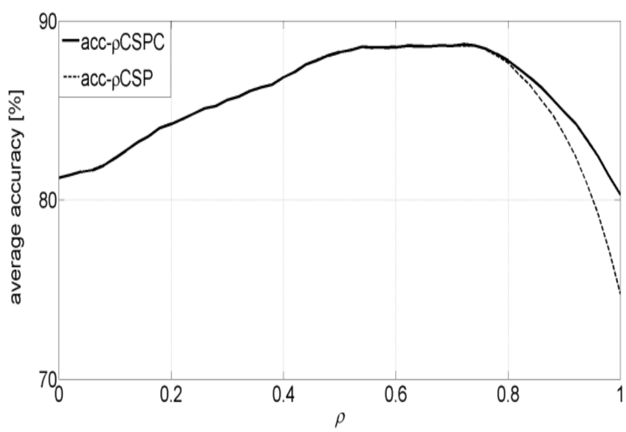

Figure 5. The dependence of *acc-ρCSP* and *acc-ρCSPC* on ρ

Fig. 4 confirms that there is an optimal value for ρ, which depends on SNR. For higher SNRs, the optimal ρ has a greater value. For the comparison with other methods we have chosen $\rho = 0.73$ as this is the value that maximizes the average accuracy in the SNR range 0-15 dB. If the development database is not available or limited, Fig. 4 can be used to choose the optimum ρ, for a general or a specific narrow SNR domain.

The accuracy effect of the omitted coherence term from formula (11) of the acc-ρCSP is visible in Fig. 5. This term makes the difference between acc-ρCSPC and acc-ρCSP. It can be observed for $\rho \in [0, 0.77]$ the accuracy of the two methods is equal. For $\rho > 0.77$ acc-ρCSPC outperforms acc-CSP. However, these are not usual values for ρ (which has the optimum value at 0.73). Hence, in order to improve the computation speed, the acc-ρCSP method can be chosen instead of acc-ρCSPC (i.e., omitting the coherence term). But, more precise computing time results will be shown in next part of this section.

C. *Extended evaluations for generalized cross-correlation time delays estimation methods*

After the calibration stage for *acc-ρCSPC* and *acc-ρCSP* we continue to evaluate the traditional generalized cross-correlation methods for time delay estimation. In this scenario we tested all GCC approaches, which were presented in Table I. We implemented all three block diagrams, shown in Fig. 1, 2 and 3. Then, we used the *evaluation* part of Noizeus database, for all four metrics. As a special notice, we used m=4 for CSP-m method.

The large analysis window was set to 2048 samples. This corresponds to a single large frame analysis for first scheme represented in Fig. 1, and K=4 smaller frames analysis (of 512 samples each) for the next two schemes represented in Fig. 2 and 3. As a naming convention, extending the accumulation cross-power spectrum technique to any method from Table I, the name of an approach is changed to *acc-* approach. In this way, our recently proposed methods were named *acc-ρCSPC* and *acc-ρCSP,* after applying the accumulation scheme to ρCSPC and ρCSP.

The length of the smaller frame size, of 512 samples, is 64ms on a sampling frequency of 8 kHz. Thus, we automatically vary the inserted delay from 5 to 50 ms, with a step of 5ms. In this way, the number of estimated perform for each delay-SNR configuration is $C_2^8 \cdot 15 = 420$ from a total of $C_2^8 \cdot 15 \cdot 4 \cdot 10 = 16800$ performed estimations. Next, the results were divided in two parts: one for the delays smaller than half of the frame size (from 0 to 30ms), and the other for delays larger than half of the frame size (from 35 to 50ms). Notice that this is meaningful only for schemes 2 and 3, because in scheme 1 we use a large frame of 2048 samples, equivalent with 256ms. In this case, all variable delays (from 5 to 50ms) are less than 128 ms, which correspond to half of the largest frame.

In the next four tables we present the error rate and standard deviation of the relative error for all 3 schemes in the above presented configuration. These metrics were computed for different SNR and delays combinations. In Table II we present results for low SNR and small delays, in Table III the results for low SNR and large delays, in Table IV the results for high SNR and small delays, and finally, in Table V the results for high SNR and large delays.

For scheme 2, in this configuration, it is important to notice that the error rate is not a relevant metric. This is due to the fact that for scheme 2 increasing the number of frames leads to a higher probability of wrong partial estimations. But, for this scheme the standard deviation of relative error is a confident metric.

Results from Table II-V confirm that if SNR increases the GCC methods perform better, yielding smaller error rate and standard deviation of relative error. Also, it is confirmed that a smaller delay has more chances to be estimated correctly than a larger one.

We notice also that not all the proposed methods perform better than the basic cross-correlation, which was already shown in [17][18][31]. This could be explained by the fact that the papers where some methods were proposed contain only mathematical presentation and no simulated results.

As expected, scheme 1 offers the smallest error rate, in all four combinations (low/high SNR and small/large delays). This is thanks to the larger analysis frame. On the other hand, scheme 1 is the slowest one. Scheme 3 is the second best scheme regarding error rate and for some weighting functions it has acceptable performances. It is faster than scheme 1 and represents a tradeoff between the computing time and the error rate. For this scheme, results confirm that *acc-ρCSPC* and *acc-ρCSP*, outperforms other methods regarding error rate results. Yet, a low error rate is observed for *acc-ρCSP*. *Acc*-CC, *acc*-CSP and *acc*-HB have reasonable error rate results. Also, the results pointed out that *acc-CPS-m* (with m=4) is an intermediate solution, between *acc-CC* and *acc-ρCSP*.

We expected that the higher the SNR, the lower the estimated delays are, better results have to be achieved. However, Tables II-V show some exceptions for standard deviation of relative error. At a first look, the results of this metric seem incomprehensible for scheme 2, because they are decreasing at larger estimated delays. To answer to this remark, we have to corroborate them with the error rate.

TABLE II. GCC METHODS EVALUATION, WITH LOW SNR (0 DB), AND SMALL DELAYS (5..30MS)

GCC name	Error rate (%)			Standard Deviation of Relative Error		
	Scheme 1	Scheme 2	Scheme 3	Scheme 1	Scheme 2	Scheme 3
CC	0.53	94.52	8.42	5.13	35.69	**13.4**
Eckart	40.46	99.72	82.71	564.68	109.59	196.95
ROTH	34.99	98.97	65.05	758.9	117.82	220.88
SCOT	51.75	99.87	64.65	201.27	78.29	78.94
CSP	4.46	95.74	19.54	228.85	99.69	94.07
CSP-m	0.48	95.29	6.93	**1.29**	33.6	17.9
HT (ML)	57.82	99.81	92.81	786.93	122.57	272.72
HB	4.46	95.74	19.54	230.8	102.75	101.65
Wiener	31.8	99.01	55.68	116.64	44.2	66.52
ρCSP	**0.44**	93.27	**4.69**	6.35	34.71	22.09
ρCSPC	0.56	93.94	5.53	3.48	**33.53**	20.37

TABLE III. GCC METHODS EVALUATION, WITH LOW SNR (0DB), AND LARGE DELAYS (35..50MS)

GCC name	Error rate (%)			Standard Deviation of Relative Error		
	Scheme 1	Scheme 2	Scheme 3	Scheme 1	Scheme 2	Scheme 3
CC	3.77	99.91	47.22	7.94	**21.54**	**26.47**
Eckart	50.6	100	97.64	223.31	40.5	79.11
ROTH	45.14	100	93.44	272.55	39.37	83.94
SCOT	55.6	100	83.16	113.09	36.73	70.90
CSP	8.07	99.96	62.42	106.65	44.31	78.89
CSP-m	1.81	99.94	31.55	**5.23**	22.34	29.99
HT (ML)	66.46	100	99.15	261.3	40.44	86.63
HB	8.07	99.96	62.42	111.73	44.84	83.10
Wiener	45.32	100	86.99	45.17	21.64	42.92
ρCSP	**1.25**	99.52	**27.52**	7.12	22.8	39.05
ρCSPC	1.35	99.57	30.87	10.31	22.91	41.78

TABLE IV. GCC METHODS EVALUATION, WITH HIGH SNR (15DB), AND SMALL DELAYS (5..30MS)

GCC name	Error rate (%)			Standard Deviation of Relative Error		
	Scheme 1	Scheme 2	Scheme 3	Scheme 1	Scheme 2	Scheme 3
CC	0	86.93	4.79	0	**27.41**	7.37
Eckart	13.63	99.69	66.95	355.88	89.89	161.42
ROTH	15.83	95.84	29.69	461.29	93.21	135.05
SCOT	49.6	99.75	56.68	0.51	55.79	39.92
CSP	0	88.82	5.86	0	61.69	38.25
CSP-m	0	84.98	1.72	0	29.89	5.76
HT (ML)	35.24	98.88	80.36	626.69	112.74	246.01
HB	0	88.82	5.86	0	67.21	45.55
Wiener	7.44	97.83	46.21	20.27	31.99	35.49
ρCSP	0	81.88	0.6	0	32.55	**3.86**
ρCSPC	0	84.13	0.6	0	35.11	4.37

TABLE V. GCC METHODS EVALUATION, WITH HIGH SNR (15DB), AND LARGE DELAYS (35..50MS)

GCC name	Error rate (%)			Standard Deviation of Relative Error		
	Scheme 1	Scheme 2	Scheme 3	Scheme 1	Scheme 2	Scheme 3
CC	0.08	100	39.62	0.51	**22.21**	17.93
Eckart	16.56	100	92.19	152.04	44.35	88.87
ROTH	34.79	100	77.77	233.78	40.71	82.75
SCOT	52.53	100	65.52	2.97	38.41	75.74
CSP	1.58	99.3	28.83	8.66	43.58	68.82
CSP-m	0	100	12.8	0	22.73	**8.33**
HT (ML)	41.83	100	96.21	213.36	42.89	86.43
HB	1.58	99.3	28.83	8.66	47.53	75.62
Wiener	28.89	100	85.85	14.04	20.58	37.10
ρCSP	0	99.4	**4.29**	0	26.99	18.18
ρCSPC	0	99.43	4.83	0	31.17	20.05

Notice that, for larger delays the error rate is 100% for almost all methods, because of incorrect estimations. In this way, it is clear that the decrease of standard deviation of relative error is due to the fact that almost all delays were much frequently estimated incorrectly, with a smaller variation.

Another important remark regards the standard deviation of relative error. In spite of a smaller error rate for GCC implemented with scheme 1, the standard deviation of the relative error is higher when comparing with implementation schemes 2 and 3. This is because scheme 1 uses a four times larger frame size, and any incorrect estimated delay varies in a larger domain. Thus, the variations of relative error are larger, leading to higher standard deviation values.

The forth evaluation metric is the processing time. Table VI provides detailed data for presented GCC methods, in all three implementation schemes. As we expected, the results confirm that the implementation scheme 1, which uses a large analysis frame, is slower than those which divided the large analysis frame in several smaller frames, like scheme 2 and 3. Moreover, the computing time for the accumulating methods is reduced even more, thanks to the benefit offered by scheme 3 (which reduces the total FFT number, form 3K to 2K+1, as we presented in Section II).

TABLE VI. COMPUTING TIME EVALUATION

GCC name	Computing time (μs)		
	Scheme 1	Scheme 2	Scheme 3
CC	92	78	61
Eckart	227	237	220
ROTH	114	100	84
SCOT	147	134	117
CSP	139	127	111
CSP-m	323	307	288
HT (ML)	268	226	209
HB	175	163	146
Wiener	199	188	171
ρCSP	296	285	268
ρCSPC	406	399	382

In all schemes, the normal cross-correlation has the fastest processing time. This is because this is the basic GCC form, which does not compute any weighting. On the opposite side, we find ρCSPC as the slowest method. Besides the ordinary FFTs, it has to spend expensive time in computing its weighting function.

Based on the presented results we conclude that, after the calibration step, ρCSP and ρCSPC provide the lowest error rate. Considering the demand of a real time application, between them, ρCSP is an obvious solution because of its lower computing time. For this kind of applications we have to take into account CC also. In any implementation scheme, CC is much faster than ρCSP, yielding acceptable error rate.

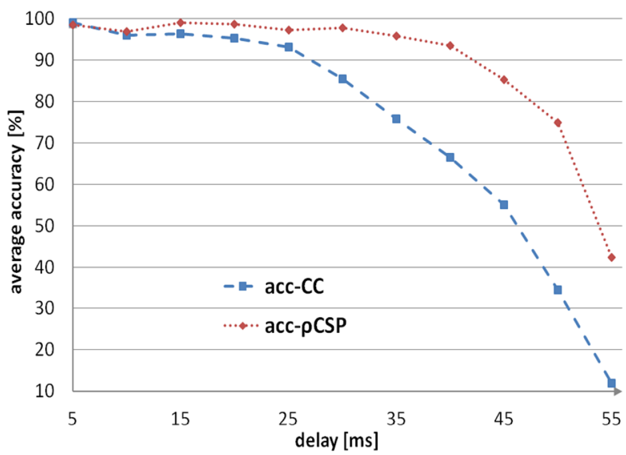

Figure 6. Comparison between acc-CC and acc-ρCSP

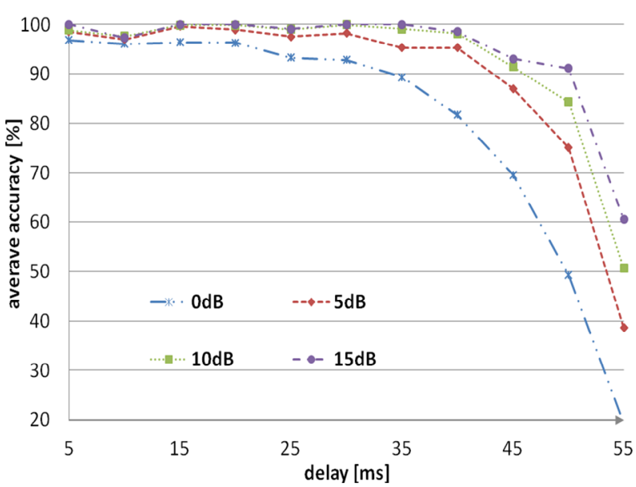

Figure 7. The influence of SNR and delay over the *acc-ρCSP* accuracy

For these reasons, we investigated the dependence of accuracy rate on the estimated delay, for *acc-ρCSP* and *acc*-CC. In Fig. 6 we used $K = 4$ frames of 512 samples each, computing the average accuracy from 0, 5, 10 and 15 dB SNR. For 50 ms delays, which represent 78% of the frame size length (64 ms), the accuracy rate for *acc-ρCSP* is around 75%, while for *acc*-CC is less than 35%. This confirms that while most of GCC methods are able to estimate accurate delays smaller than half of the frame size, *acc-ρCSP* outperform them and continues to provide reasonable accuracy for larger delays.

In Fig. 7, for *acc-ρCSP,* we present the variation of the accuracy rate by the SNR and delay. It is shown that the higher the SNR, the higher the accuracy is. For delays up to 50% of the frame size, the difference between accuracies on various levels of SNR remains almost the same. Once the delay increases over 50% of the frame size, the accuracy decreases much faster for lower SNR.

The influence of the number of frames over the GCC error rate, for scheme 3, is highlighted in Table VII. The GCC accuracies were computed for 1, 4 and 8 frames, with frame sizes of 512 samples, at 15 dB SNR. It is shown that the error rate decreases when it is used a higher number of frames. This is due to the fact that the accumulated cross power spectrum domain keeps the spectral information over multiple frames. In this way, the correlation between the frames is maintained.

TABLE VII. ERROR RATE DEPENDENCE ON THE NUMBER OF FRAMES

GCC names	**Number of frames** 1	4	8
CC	37.17	4.79	2.21
Eckart	63.67	66.95	59.69
ROTH	53.8	29.69	21.22
SCOT	64.92	56.68	45.96
CSP	26.35	5.86	1.66
CSP-m	24.94	1.72	0.05
HT (ML)	76.97	80.36	80.03
HB	26.35	5.86	1.66
Wiener	68.42	46.22	26.56
ρCSP	18.2	0.6	0.56
ρCSPC	19.3	0.6	0.56

The numerical and simulated results confirm that our recently proposed methods *acc-ρCSP* and *acc-ρCSPC* in [2] achieve the highest accuracy rate for multi frame processing.

IV. CONCLUSION AND FUTURE WORK

In this paper, we evaluated traditional generalized cross-correlation time delay estimation methods, applying them the accumulated cross-power spectrum technique. The experiments were performed using the standard Noizeus database. The obtained results showed that, a single large frame yields the smallest error rate when comparing with different multi frame implementation. On the other hand, the *accumulation scheme* over smaller multiple frames is faster than the above approach, providing acceptable error rates for a part of GCC methods. For this scheme also, the increasing number of frames leads to a smaller error rate.

Our analysis showed that ρCSP and ρCSPC provide the lowest error rate, with a small benefit for the first one. CPS-m, CC, CSP and HB have reasonable error rate results. In the same conditions, the others methods like Eckart, ROTH, SCOT, HT (ML) and Wiener do not offer acceptable error rate results.

Regarding the processing time, the normal cross correlation is the faster method because it does not compute any weighting. On the opposite side, ρCSPC has to perform many time consuming operations to calculate weighting function, so it is the slowest one. Between the three presented schemes, the first, which analyzes the signals using one large frame is the slowest, but offers the highest accuracy. The second scheme, which works on multiple frames of fewer samples averaging the final estimate in time domain, is a little faster, but does not provide any usable accuracy results. The third scheme works on smaller frames, like scheme 2, but it accumulates the cross-power spectrum in frequency domain. This leads to a good accuracy and is also the fastest scheme.

The results from this work could be used for a better decision regarding the implementation of GCC methods, based on applications' demands. For expected delays that are comparable with the available analysis window, it is recommended to use a single large frame implementation. But, if the expected delays are much smaller than the available analysis window a faster solution is represented by the accumulation scheme of the cross-power spectrum in frequency domain. Each of these schemes can be efficiently implemented to provide solution for realigning noisy signals in applications such as speech enhancement, echo canceling, seismic and medical processing, radar and sonar localization, and pattern detection.

Future work will involve analysis *for acc-CSP-m,* from which we expect better accuracy results after a proper calibration for *m*. We will continue to focus on these methods and their applications in the VoIP environment and multi-channel speech enhancement.

REFERENCES

[1] R.S. Marinescu, A. Buzo, H. Cucu, and C. Burileanu, "New Considerations for Accumulated ρ-Cross Power Spectrum Phase with Coherence Time Delay Estimation", In Procedings of ICDT 2013, The Eight International Conference of Digital Telecommunications, Apr 2013, pp. 55-59.

[2] R.S. Marinescu, A. Buzo, H. Cucu, and C. Burileanu, "Fast Accurate Time Delay Estimation Based on Enhanced Accumulated Cross-Power Spectrum Phase", European Signal Processing Conference (EUSIPCO 2013), Morocco, 2013, "in press"

[3] R.S. Marinescu, A. Buzo, H. Cucu, and C. Burileanu, "Extensive Evaluation Experiments for the Recently Proposed Accumulated Cross-Power Spectrum Methods for Time Delay Estimation", SpeD 2013, "in press"

[4] B. Widrow and S.D. Strearns, "Adaptive signal processing", Penitence-Hall, ISBN 0130040290, USA, 1985.

[5] S.N. Lin and S.J. Chern, "A new adaptive constrained LMS time delay estimation algorithm", Signal Processing, Volume 71, Issue 1, Nov. 1998, pp. 29-44.

[6] A.W.H. Khong and P.A. Naylor, "Efficient Use Of Sparse Adaptive Filters", In Proceedings of ACSSC '06, Fortieth Asilomar Conference on Signals, Systems and Computers, article ID 10.1109/ACSSC.2006.354982, Nov. 2006, pp. 1375-1379.

[7] R.A. Dyba, "Parallel Structures for Fast Estimation of Echo Path Pure Delay and Their Applications to Sparse Echo Cancellers", In Proceedings of CISS 2008, 42nd Annual Conference on Information Sciences and Systems, article ID 10.1109/CISS.2008.4558529, Mar. 2008, pp. 241-245.

[8] D. Hongyang and R.A. Dyba, "Efficient Partial Update Algorithm Based on Coefficient Block for Sparse Impulse Response Identification", In Proceedings of CISS 2008, 42nd Annual Conference on Information Sciences and Systems, article ID 10.1109/CISS.2008.4558527, Mar. 2008, pp. 233-236.

[9] D. Hongyang and R.A. Dyba, "Partial Update PNLMS Algorithm for Network Echo Cancellation", In Proceedings of ICASSP 2009, IEEE International Conference on Acoustics, Speech and Signal Processing, article ID 10.1109/ICASSP.2009.4959837, Apr. 2009, pp. 1329-1332.

[10] K. Sakhnov, E. Verteletskaya, and B. Simak, "Partial Update Algorithms and Echo Delay Estimation," Communications – Scientific Journal of the University of Zilina, Zilina – Slovakia, vol. 13, no. 2, Apr. 2011, pp. 14-19.

[11] J. Benesty, "Adaptive eigenvalue decomposition algorithm for passive acoustic source localization", J. Acoust. Soc. Am. Volume 107, Issue 1, Jan. 2000, pp. 384-391.

[12] C. Knapp and G.C. Carter, "The Generalized Correlation Method for Estimation of Time Delay", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 24, issue 4, Aug. 1976, pp. 320-327.

[13] C. Eckart, "Optimal Rectifier System for Detection of Steady Signals", Univ. California, Scripps Inst. Oceanography, Marine Physical Lab. Rep SIO 12692, SIO Ref 52-11, 1952.

[14] P.R. Roth, "Effective measurements using digital signal analysis", IEEE Spectrum, vol 8, Apr 1971, pp. 62-70.

[15] E.J. Hannan and P.J. Thomson, "The Estimation of Coherence and Group Delay", Biometrika, vol. 58, Dec 1971, pp. 469-481.

[16] G.C. Carter, A.H. Nutall, and P.G. Cable, "The Smoothed Coherence Transform", Proc. IEEE (Lett.), vol. 61, Oct 1973, pp 1497-1498.

[17] J.C. Hassab, R.E. Boucher, "A Cuantitative Study of Optimum and Suboptimum filters in Generalized Correlator", IEEE 1979 Int. Conf. Acoust., Speech, Signal Processing Conf. Rec. 79CH 1379-7 ASSP, 1980, pp. 124-127.

[18] J.C. Hassab, R.E. Boucher, "Performance of the Generalized Cross Correlator in the Presence of a Strong Spectral Peak in the Signal", IEEE Transaction on Acoustics, Speech, and Signal Processing, vol. assp-29, no. 3, Jun. 1981, pp. 549-555.

[19] J.C. Hassab, R.E. Boucher, "Optimum Estimation of Time Delay by Generalized Correlator", IEEE Transaction on Acoustics, Speech, and Signal Processing, vol. assp-27, no. 3, Aug. 1979, pp. 373-380.

[20] A.O. Hero and S.C. Schwartz, "A New Generalized Cross Correlator", IEEE Transactions, Acoustics, Speech and Signal Processing, vol. 33, Feb 1985, pp. 38-45.

[21] D.V. Rabinkin, R.J. Renomeron, A. Dahl, J.C. French, J.L. Flanagan, and M.H. Bianchi, "A DSP Implementation of Source Location Using Microphone Arrays", The Journal of the Acoustical Society of America, Volume 99, Issue 4, Apr 1996, pp. 2510-2527.

[22] M. Shean and H. Liu, “A Modified Cross Power-Spectrum Phase Method Based on Microphone Array for Acoustic Source Localization,” IEEE International Conference on System, Man and Cybernetics, San Antonio, TX, USA, Oct. 2009, pp. 1286 – 1291.

[23] M. Matassoni and P. Svaizer, “Efficient Time Delay Estimation Based on Cross-Power Spectrum Phase”, European Signal Processing Conference (EUSIPCO), Florence - Italy, Sep 2006.

[24] D.H. Youn, N. Ahmed, and G.C. Carter, “On the Roth and SCOTH Algorithms: Time-Domain Implementations”, In Proceedings of the IEEE, vol. 71, issue 4, 1983, pp. 536-538.

[25] Q. Tianshuang and W. Hongyu, “An Eckart-weighted adaptive time delay estimation method”, IEEE Transactions on Signal Processing, vol. 44, issue 9, Sep. 1996, pp. 2332-2335.

[26] M. Omologo and P. Svaizer, “Acoustic event localization using a crosspower-spectrum phase based technique”, Proceedings of ICASSP, Australia, Apr. 1994, pp. 273-276.

[27] M. Omologo and P.Svaizer, “Use of the crosspower-spectrum phase in acoustic event location”, IEEE Transactions on Speech Audio Process, May 1997, pp. 288-292.

[28] V. Zetterberg, M.I. Pettersson, and I. Claesson, “Comparison Between Whitened Generalized Crosscorrelation and Adaptive Filter for Time Delay Estimation”, In Proceedings of TS/IEEE, OCEANS, vol. 3, article ID 10.1109/OCEANS.2005.1640117, Sept. 2005, pp. 2356 – 2361.

[29] K.W. Wilson and T. Darrell, “Learning a Precedence Effect-Like Weighting Function for the Generalized Cross-Correlation Framework”, IEEE Transactions on Audio, Speech, and Language Processing, vol. 14, issue 6, Nov. 2006, pp. 2156-2164.

[30] Y. Sun and T. Qiu, “The SCOT Weighted Adaptive Time Delay Estimation Algorithm Based on Minimum Dispersion Criterion”, In Proceedings of the ICICIP Conference on Intelligent Control and Information Processing, Aug. 2010, pp. 35-38.

[31] K. Sakhnov, E. Verteletskaya, and B. Simak, “Echo Delay Estimation Using Algorithms Based on Cross-correlation,” Journal of Convergence Information Technology, Volume 6, Number 4, Apr. 2011, pp. 1 – 11.

[32] NOIZEUS: A noisy speech corpus http://www.utdallas.edu/~loizou/speech/noizeus/ [retrieved: Aug, 2013]

[33] AURORA database, http://www.elda.org/article52.html [retrieved: Aug, 2013]