Extensive Evaluation Experiments for the Accumulated Cross-Power Spectrum Methods for 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 numerous real time signal processing applications, time delay estimation (TDE) is still a hot topic. A recently proposed solution, based on Generalized Cross Correlation (GCC) method, is evaluated further in this paper. Experimental results show that among traditional variants of GCC methods the accumulated $\rho$-cross power spectrum phase proposed by us is the most accurate. Another important aspect is that the computing time of this method is comparable with those of other popular GCC methods.

Keywords—Time delay Estimation; Generalized Cross Correlation; Accumulated $\rho$-Cross Power Spectrum Phase

I. INTRODUCTION

Time Delay Estimation (TDE) is an active part in various systems. It is used as a solution for echo cancelling, for acoustic, radar and sonar localization, in pattern recognition, seismic, medical and speech processing. The variety of applications with use TDE has different requirements regarding optimization and integration. Because of this it is inhibited the development of a unique solution for TDE.

Based on applications specific demanding, the various approaches to time delay estimation can be grouped into three main categories: a) generalized cross-correlation (GCC), b) least mean square (LMS) adaptive filtering and c) adaptive eigenvalue decomposition (AEVD). Each category has its own advantages, which make it optimal for particular applications. The latter of these three approaches (AEVD) was proved to be very efficient in reverberant environments [1]. The second class of TDE methods, the adaptive filtering [2-8], is a useful solution for applications with a high accuracy demanding and easy constraints on the adaptation time. But, in applications with a fast response the solution is provided by the first category based on generalized cross-correlation. It does not need any adaptation time, and offers a reasonable accuracy.

The generalized cross-correlation method was proposed initially by Knapp and Carter [9], in 1976. They also presented a particular derivation of the GCC, with weighting, CSP-m. Gradually, over the time, based on this work, multiple variations of the GCC weighting function were proposed or integrated in this technique: ROTH [10], SCOT [11], Eckart [12], Phase Transform (PHAT) or Cross-power Spectrum Phase (CSP) [9], HT (ML) [13], HB [14], Wiener [15], $\rho$-CSP [16], accumulated CSP (acc-CSP) [17], $\rho$-CSPC [18]. Over the time were presented several papers which studied these methods. In one of them [19], the methods were compared based on the mean value and root mean square deviation of the estimated delay and.

As a further development, in [20] we presented two new methods: accumulated $\rho$-cross power spectrum with coherence (acc-$\rho$CSP) and accumulated $\rho$-cross power spectrum (acc-$\rho$CSP). They benefit from the lower computational load and robustness of the acc-CSP method [17] and the higher accuracy, which characterized the $\rho$-CSP [16] and $\rho$-CSPC [18]. In [20] it was shown that acc-$\rho$CSP and acc-$\rho$CSP outperform previous methods in computation time, because of the accumulation of cross-power spectrum phase in frequency domain, introduced in [17]. This leads to only one Inverse Discrete Fourier Transform (IDFT) for any number of accumulated frames used. Also, in [20], it was shown that the first method (acc-$\rho$CSP) generally has a higher accuracy than (acc-$\rho$CSP), but, after a proper calibration, the second method achieves practically the same accuracy as the acc-$\rho$CSP at a lower computational load.

In [21], we continued to evaluate the proposed acc-$\rho$CSP method over their previous primitives’ methods. The obtained results from [21] showed that the new combination of acc-$\rho$CSP, based on previous acc-CSP and $\rho$CSP, offers higher accuracy rate and faster computational speed. Another remark was the acc-$\rho$CSP, which is performed in the frequency domain, leads to an accuracy that is more than twice than the accuracy obtained with the previous methods, which average the final results in the time domain. Also, the accuracy of the acc-$\rho$CSP increases with the number of frames, while for previous methods that use the average in time, it decreases as the number of frames grows [21].

For a complete study of acc-$\rho$CSP and acc-$\rho$CSPC methods, we continued to evaluate them, among other well known generalized cross-correlation methods. In this way, this work appends to the previous evaluations, from [20] and [21], offering detailed results across the variety of GCC functions.

The paper is organized as follows. In first part of Section II we present the time delay estimation problem and various
GCC approaches. The next part is dedicated to our recently proposed TDE methods. Experiments and results discussions stand on Section III. Conclusions and further work are presented in Section IV.

II. GENERALIZED CROSS-CORRELATION FOR TIME DELAY ESTIMATION

A. Background

In several applications, a transmitted signal \( x(t) \) could be received by two (or more) noisy and delayed signals \( y_1(t) \) and \( y_2(t) \). The problem of finding the relative delay between these signals is called time delay estimation. To solve this, over the time were proposed different methods, but a time-efficient and widely are those based on the cross-correlation. They are called generalized cross-correlation, named by the work of Knapp and Carter from [9], where it is incorporated a filtering function:

\[
R_{y_1y_2}^e(t) = \int_{-\infty}^{\infty} \Psi(f) \cdot G_{y_1y_2}(f) \cdot e^{j2\pi ft} df
\]  

(1)

where \( \Psi(f) \) is a general frequency weighting and \( G_{y_1y_2}(f) \) represents the cross-power spectrum. The insertion of the weighting function is expected to emphasize spectral information that can depend on source and noise characteristics in order to improve the delay estimation. Thus, the estimated delay is found at the argument \( t \) which maximizes the cross-correlation function.

Table I represents various weighting functions used in this work, where \( G_{y_1y_1}(f) \) and \( G_{y_2y_2}(f) \) are auto power spectrum of the noisy signals and \( \gamma_{yy}(f) \) is the signal's coherence function:

\[
\gamma_{yy}^2(f) = \frac{|G_{y_1y_1}(f)|^2}{G_{y_1y_1}(f) \cdot G_{y_2y_2}(f)}
\]

(2)

For the Cross-Correlation the weighting function \( \Psi(f) \) is 1. This is the basic and the fastest computing GCC, because practically it has not any weighting operation.

The processor proposed by Roth in 1971, has desirable effect of suppressing those frequency regions where \( G_{y_1y_1}(f) \) is large and the estimate of \( G_{y_1y_2}(f) \) is more likely to be in error [9].

The SCOT (Smoothed Coherence Transform) was introduced by Carter, Nuttall and Cable in 1973, to reduce the influence of a strong tonal. However, for smoothed signal and noise spectra, Hassab and Boucher [14],[22] have noted that the additional SCOT weighting function has weakened the performance of the basic cross correlator while other functions have improved it [23].

PHAT was developed purely as an ad-hoc technique to avoid spreading of the above two presented operators [9]. Ideally, PHAT does not suffer the spreading that other weighting functions do. Also, because it weights \( G_{y_1y_2}(f) \) as the inverse of \( G_{y_1y_1}(f) \), the errors are accentuated where signal power is smallest.

<table>
<thead>
<tr>
<th>GCC name</th>
<th>Weighting function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-Correlation</td>
<td>1</td>
</tr>
<tr>
<td>ROTH</td>
<td>( 1/G_{y_1y_1}(f) )</td>
</tr>
<tr>
<td>SCOT</td>
<td>( 1/\sqrt{G_{y_1y_1}(f) \cdot G_{y_2y_2}(f)} )</td>
</tr>
<tr>
<td>CSP (PHAT)</td>
<td>( 1/G_{y_2y_2}(f) )</td>
</tr>
<tr>
<td>CSP-m</td>
<td>( 1/\sqrt{G_{y_1y_1}(f) \cdot G_{y_2y_2}(f)} )</td>
</tr>
<tr>
<td>Eckart</td>
<td>( \frac{G_{y_1y_1}(f)}{G_{y_1y_1}(f) - G_{y_1y_2}(f) \cdot G_{y_2y_2}(f)} )</td>
</tr>
<tr>
<td>HT (ML)</td>
<td>( \frac{</td>
</tr>
<tr>
<td>HB</td>
<td>( G_{y_1y_2}(f)/G_{y_1y_1}(f) \cdot G_{y_2y_2}(f) )</td>
</tr>
<tr>
<td>Wiener</td>
<td>(</td>
</tr>
<tr>
<td>( \rho )-CSP</td>
<td>( 1/</td>
</tr>
<tr>
<td>( \rho )-CSPC</td>
<td>( \frac{1}{</td>
</tr>
</tbody>
</table>

The Eckart filter derives its name from the work done in this area in [12] published in 1951 and it aims to maximize the deflection criterion [9].

In 1971 it was proposed HT processor. It assigns greater weight in regions of frequency domain where the coherence is large. In [9], it was shown that HT processor is a maximum likelihood (ML) estimator for time delay under usual conditions. Under a los signal-to-noise ratio restriction, the HT processor is equivalent to Eckart prefiltering and cross-correlation.

Eight years later, it was presented the HB processor. For highly dynamic spectra, in attempt to reject strong tonals from the observations high and low SNR regions are suppressed from the cross-spectral estimate [14].

The Wiener processor [15] was proposed in 1985. Based on channel’s linearity, it tries to estimate the original signal as best it can 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.

In 1996 it was presented a new weighting function, the \( \rho \)-CSP. It adds to the normal CSP (PHAT) the tuning parameter \( \rho \) (with values between 0 and 1) as a whitening parameter, which discards the non-speech portion (below 200Hz) of the CSP.

Relatively recently, in addition to the above work, in 2009 was proposed \( \rho \)-CSP. The presence of the minimum of the coherence function in the weighting function helps to reduces errors for relatively small energy signals.

To compute a TDE, the GCC methods use an analysis window. Because of the way the signals are received, the analysis window is usually split in several frames. Then the cross-power spectrum is calculated for each frame. An optimized implementation involves 3 Fast Fourier Transform (FFT) computations. First two are used to calculate the spectra of the input signals, followed by the weighting operation in frequency domain. The third FFT, which is in fact an Inverse Fast Fourier Transform (IFFT), is used to return to the time domain in order to estimate the delay of the current frame. Finally, the overall estimate is obtained by averaging all frame estimates.

Due to the fact that the response time is important in almost all TDE applications, the focus should be on two factors: processing time and window length. The use of a larger frame leads to a higher accuracy rate for correct TDE, but the downside is the computing time, which depends on the frame size. In 2006, Matassoni and Svaizer proposed acc-CSP [17], as a variant to compute the CSP over multiple frames. The novelty of the method was that the time delay is estimated by averaging CSP over all frames, in the frequency domain, while for the previous methods the average is computed in the time domain. In this way the computing time decreases because the cross-power spectrum phase is accumulated over multiple frames, remaining only one IFFT to be computed after the last accumulation. In the frequency domain this is expressed as follows:

\[
G_{\text{acc-CSP}}(f) = \frac{1}{K} \sum_{k=1}^{K} \frac{G_{y,\gamma_k}(f)}{G_{y,\gamma_k}(f)^2}
\]

where \( K \) represents the number of accumulated frames.

The previous methods propose to compute the TDE as the average of all partial estimated delays of each frame from the analysis window. While in previous approaches the total FFT operations is \( 3 \times K \), for the accumulation approach from (3) it is reduced to only \( 2 \times K + 1 \). In this way the computing speed is increased and also, for fixed delay during the analysis window, the delay estimation is enhanced because of the intrinsic integration, how it is shown in [17].

B. Accumulated \( \rho \)-Cross-Power Spectrum Phase Methods

Recently, in [20], we proposed to combine acc-CSP with \( \rho \)-CSP and \( \rho \)-CSPC, resulting the new methods accumulated \( \rho \)-Cross-Power Spectrum Phase (acc-\( \rho \)CSP) and accumulated \( \rho \)-Cross Power Spectrum Phase with Coherence (acc-\( \rho \)CSPC). In frequency domain we can express these, in the same order, by (4) and (5):

\[
G_{\text{acc-}\rho \text{CSP}}(f) = \sum_{k=1}^{K} \frac{G_{y,\gamma_k}(f)}{G_{y,\gamma_k}(f)^2}
\]

\[
G_{\text{acc-}\rho \text{CSPC}}(f) = \sum_{k=1}^{K} \frac{G_{y,\gamma_k}(f)}{G_{y,\gamma_k}(f)^2} + \min \left( \gamma_{y,\gamma_k}(f) \right)
\]

where \( \gamma_{y,\gamma_k}(f) \) is the signal's coherence function for the \( k \)-th frame, computed easily with (2).

These new methods incorporate the advantages of their primitives: accuracy improvements from pcsp and pCSPC, and the accumulation scheme from acc-CSP. These were analyzed in [20] and [21], concluding in a higher accuracy rate in low SNR conditions.

The whitening parameter \( \rho \) emphasizes the speech frequencies from the spectrum, reducing the noise impact over the non-speech frequencies of the spectrum. The addition of minimum coherence term in acc-\( \rho \)CSPC prevents the errors introduced by a small denominator in the case of small energy signal parts [18].

If acc-\( \rho \)CSP and acc-\( \rho \)CSPC are adequately calibrated, as it is proposed in [20], theirs accuracy rate are practically equal, but the first method is faster because it does not need to compute the minimum coherence term.

III. EXPERIMENTAL RESULTS AND DISCUSSIONS

A. Experimental Setup

In previous papers, [20] and [21], we evaluated the new methods, comparing them with theirs primitive methods. In this work, we continued a further evaluation and discussions for these methods, but we extended the comparison over all well-known and frequently used GCC methods.

All the methods used in next experiments were presented in sections II.A and II.B. They were implemented in Matlab and C, for accuracy, and respectively for computing time evaluation. We used the international Noizeus [24] speech corpus as the primary database. It has 30 sentences, from 6 persons (3 males and 3 females), on which were added 8 different real-world noises (suburban train, babble, car, exhibition hall, restaurant, street, airport and train station noise), from the AURORA database [25] at 4 different SNRs (0, 5, 10 and 15 dB), with the sampling frequency of 8 kHz.

Summing all combinations of noise types yields a total \( C^2_8 = 28 \) possibilities. For relevant results we present the comparison for 0 and 15 dB SNR. In the next experiments we used \( \rho = 0.73 \) for the whitening parameter, which was configured as mentioned in [20]. Additionally to this configuration, no overlapping was used this time. Despite the
fact that for the CPS-\(m\) weighting function \(m\) is chosen in
generally between 1 and 2, we obtained better results with
\(m = 4\).

In this work we used more metrics: accuracy, relative
error, standard deviation of the relative error and computing
time. The accuracy rate is defined as the ratio between the
number of perfectly estimated delays and the total number of
 estimations performed. This metric was the most relevant one
because we intend to apply this TDE to multi-channel speech
enhancement applications which need a perfect delay
estimation. The relative error represents the relative difference
between the estimated and correct delay. For the processing
time we implement the methods in C language, compiled with
Gnu Compiler Collection (gcc) version 4.7.3, and run tests on
a machine with Intel Core i5 processor and Ubuntu 13.04
operating system.

B. Evaluation of the Generalized Cross-Correlation Methods

We evaluated the above methods for 10 different
artificially introduced delays (from 5 to 50 ms) on frame size
of 1024 samples, and \(k = 4\). Taking into account the
combinations described in the previous subsection, for every
different signal-to-noise level (0, 5, 10 and 15 dB) the total
number of tested pairs was 28x30x10=8400. Obtained results
for 0 and 15 dB SNR are presented in Table II and III. The
accuracy, relative error and standard deviation represent the
average on all 8400 tested pairs.

The last three rows from Table II and III contains results
for methods proposed by us which estimate time delay by
averaging CSP over all frames, in the frequency domain (acc-
CSP, acc-pCSP and acc-pCSPC). For these three methods we
obtained high accuracy rates, and for the last two 100%. The
impressive differences between these methods and the above,
which estimate delay by averaging in time domain, impose
them as a real solution, which offer high accuracy.

Better results are obtained in Table III, because the higher
signal-to-noise ratio is, more accurate time delay estimation
is achieved. It is notable that how the accuracy for CSP and HB
is practically the same at different signal-to-noise ratios, but
mean relative error and standard deviation are better for CSP.
This is explained by the similarities between the weighting
functions.

From the group of GCC methods which does not use the
accumulating scheme, proposed by Matassoni and Svaizer in
[17], \(\rho\)CSP and \(\rho\)CSPC have the highest accuracy ration,
followed by the CSP-m. This confirms again our decision to
apply the accumulation scheme to \(\rho\)CSP and \(\rho\)CSPC to obtain
improved accuracy for acc-\(\rho\)CSP and acc-\(\rho\)CSPC.

Over all, the results show that the proposed acc-\(\rho\)CSPC
outperforms the other previous methods by obtaining higher
accuracy and a lower relative error in delay estimation. In
addition, the lower values for relative error and standard
deviation of the relative error show that even if the signals are
not perfectly aligned the gap is smaller than in the other
methods. This means that acc-\(\rho\)CSP and acc-\(\rho\)CSPC are more

\[
\begin{array}{|c|c|c|c|}
\hline
\text{GCC name} & \text{Accuracy (%)} & \text{Relative Error (%)} & \text{Standard deviation of relative error} \\
\hline
\text{Cross-Correlation} & 8.8 & 34.5 & 42.6 \\
\text{ROTH} & 0.4 & 86.6 & 169.0 \\
\text{SCOT} & 0.3 & 51.6 & 118.0 \\
\text{CSP} & 5.5 & 48.2 & 148.0 \\
\text{CSP-m} & 12.5 & 32.2 & 40.7 \\
\text{Eckart} & 0.4 & 84.0 & 153.0 \\
\text{HT (ML)} & 0.1 & 74.7 & 172.0 \\
\text{HB} & 5.5 & 102.7 & 148.0 \\
\text{Wiener} & 0.9 & 48.6 & 64.4 \\
\text{pCSP} & 17.2 & 31.6 & 41.0 \\
\text{pCSPC} & 15.1 & 32.3 & 39.0 \\
\text{acc-CSP} & 88.4 & 11.0 & 60.0 \\
\text{acc-pCSP} & \sim 100 & \sim 0 & \sim 0 \\
\text{acc-pCSPC} & \sim 100 & \sim 0 & \sim 0 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|c|}
\hline
\text{GCC name} & \text{Accuracy (%)} & \text{Relative Error (%)} & \text{Standard deviation of relative error} \\
\hline
\text{Cross-Correlation} & 25.3 & 32.0 & 46.0 \\
\text{ROTH} & 17.4 & 50.7 & 119.5 \\
\text{SCOT} & 3.0 & 30.9 & 65.9 \\
\text{CSP} & 47.6 & 16.9 & 71.1 \\
\text{CSP-m} & 53.6 & 26.6 & 44.5 \\
\text{Eckart} & 5.1 & 54.5 & 115.5 \\
\text{HT (ML)} & 1.9 & 62.9 & 143.2 \\
\text{HB} & 47.6 & 34.2 & 72.5 \\
\text{Wiener} & 4.5 & 46.4 & 53.0 \\
\text{pCSP} & 63.8 & 20.4 & 43.9 \\
\text{pCSPC} & 60.9 & 19.4 & 45.8 \\
\text{acc-CSP} & 99.2 & 0.8 & 7.0 \\
\text{acc-pCSP} & \sim 100 & \sim 0 & \sim 0 \\
\text{acc-pCSPC} & \sim 100 & \sim 0 & \sim 0 \\
\hline
\end{array}
\]

Table IV and V present important aspects regarding the
accuracy and the computing time, in different implementation
variants. In some applications it is possible to use a large
analysis window, without dividing it into smaller frames. The
advantage here is it has a higher accuracy rate. On the other hand,
the processing time increases. Also, the longer the analysis
window, the more difficult to track correctly unstable delay. In
this condition, the final choice must consider all these aspects,
and we need a good knowledge of all the possible solution.
For this reason, we present in the next tables the accuracy (for
0 and 15 dB signal-to-noise ratio) and computing time for two
implementations. First divides the analysis window in 4
smaller frames, of 1024 samples, while the second performs
the time delay estimation once, on the entire analysis window
of 4096 samples.
In Table IV it is shown that the increase of the signal-to-noise ratio and analysis frame lead to a higher accuracy rate for almost all methods. For the single frame analysis, the accumulation methods reduce to their primitives, i.e., acc-CSP, acc-pCSP and acc-pCSPc are reduced to CSP, pCSP and pCSPc.

### Table IV. Accuracy Comparison

<table>
<thead>
<tr>
<th>GCC name</th>
<th>Accuracy – 0 dB (%)</th>
<th>Accuracy – 15 dB (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Short frames</td>
<td>Long frame</td>
</tr>
<tr>
<td>Cross-Correlation</td>
<td>8.8</td>
<td>100</td>
</tr>
<tr>
<td>ROTH</td>
<td>0.4</td>
<td>77.4</td>
</tr>
<tr>
<td>SCOT</td>
<td>0.3</td>
<td>48.0</td>
</tr>
<tr>
<td>CSP</td>
<td>5.5</td>
<td>99.8</td>
</tr>
<tr>
<td>CSP-m</td>
<td>12.5</td>
<td>100</td>
</tr>
<tr>
<td>Eckart</td>
<td>0.4</td>
<td>74.5</td>
</tr>
<tr>
<td>HT (ML)</td>
<td>0.1</td>
<td>55.6</td>
</tr>
<tr>
<td>HB</td>
<td>5.5</td>
<td>99.8</td>
</tr>
<tr>
<td>Wiener</td>
<td>0.9</td>
<td>78.7</td>
</tr>
<tr>
<td>pCSP</td>
<td>17.2</td>
<td>100</td>
</tr>
<tr>
<td>pCSPc</td>
<td>15.1</td>
<td>100</td>
</tr>
<tr>
<td>acc-CSP</td>
<td>88.4</td>
<td>100</td>
</tr>
<tr>
<td>acc-pCSP</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>acc-pCSPc</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

While for a short frame size of 128 ms the maximum introduced delay of 50 ms represents almost 40%, for a 4 times larger frame it represents around 10%. From Table IV we observe that the large analysis frame leads to higher accuracy rate for the almost all methods, while the smaller multi-frame approach could not offer any reliable results for any methods, with an exception for accumulated methods. Moreover, acc-ρCSP and acc-pCSPc offer highly accurate estimation even for larger delays of about 40% of the frame size length. Thus, this remark makes them very useful for real time solution of time delay estimation.

### Table V. Computing Time Evaluation

<table>
<thead>
<tr>
<th>GCC name</th>
<th>Computing time (µs)</th>
<th>Difference (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Short frames</td>
<td>Long frame</td>
</tr>
<tr>
<td>Cross-Correlation</td>
<td>184</td>
<td>224</td>
</tr>
<tr>
<td>ROTH</td>
<td>228</td>
<td>268</td>
</tr>
<tr>
<td>SCOT</td>
<td>295</td>
<td>335</td>
</tr>
<tr>
<td>CSP</td>
<td>282</td>
<td>314</td>
</tr>
<tr>
<td>CSP-m</td>
<td>593</td>
<td>688</td>
</tr>
<tr>
<td>Eckart</td>
<td>538</td>
<td>558</td>
</tr>
<tr>
<td>HT (ML)</td>
<td>528</td>
<td>566</td>
</tr>
<tr>
<td>HB</td>
<td>353</td>
<td>383</td>
</tr>
<tr>
<td>Wiener</td>
<td>422</td>
<td>432</td>
</tr>
<tr>
<td>pCSP</td>
<td>599</td>
<td>630</td>
</tr>
<tr>
<td>pCSPc</td>
<td>826</td>
<td>850</td>
</tr>
<tr>
<td>acc-CSP</td>
<td>247</td>
<td>314</td>
</tr>
<tr>
<td>acc-pCSP</td>
<td>560</td>
<td>630</td>
</tr>
<tr>
<td>acc-pCSPc</td>
<td>788</td>
<td>850</td>
</tr>
</tbody>
</table>

The computing time for all the methods is presented in Table V. We compare 2 integration schemes. For the first one we use 4 frames of 1024 samples each, and for the second we use one large analysis frame of 4096 samples. Last column from Table V is reserved for the relative computing time difference between the two integration schemes.

It is easy to see that the lowest computing time is achieved for the basic cross-correlation method. It is the fastest one because this method does not have to apply any weighting. Depending on the complexity of each weighting function, computing time increases accordingly, and the slowest method is pCSPc. It is about four times slower than the basic cross-correlation, but in the same time, it offers a higher accuracy rate.

Results from Table V confirm that the implementations which use a large analysis frame, is slower than those which divided the large analysis frame in more smaller frames. This is explained by the complexity of the FFT. For a large frame size of \( N \) samples the FFT complexity is \( O(N \cdot \log(N)) \). If this large frame of \( N \) samples is divided in \( K \) frames of \( n \) samples, then the complexity is \( O(K \cdot n \cdot \log(n)) = O\left(\frac{N}{K} \cdot \log\left(\frac{N}{K}\right)\right) \). Thus, this reduction applies for all methods. But, the computing time for the accumulating methods is reduced more, due to the benefit offers by this scheme, which reduces the FFT numbers, form \( 3 \times K \) to \( 2 \times K + 1 \), as we presented in Section II.A.

We have to point out that these values can change if different systems are used. Digital signal processor (DSP), Field-Programmable Gate Array (FPGA) or different processor family can improve processing time for the above methods, but they can also change the proportionality between the methods presented in Table V.

The acc-pCSP method has a computing time comparable with all other methods, offering also a high accuracy rate. Comparing with acc-pCSPc, it yields practically the same accuracy, but with a faster computing time. Despite the various constrains that each application imposes for processing time and accuracy, acc-pCSP performances recommend it as an interesting and efficient solution for time delay estimation.

### IV. Conclusion

In this work, we further evaluated our proposed TDE methods, accumulated-\( \rho \)-Cross Power Spectrum Phase (acc-\( \rho \)-CSP) and accumulated-\( \rho \)-Cross Power Spectrum Phase with Coherence (acc-\( \rho \)-CSPC), among others well known generalized cross-correlation methods. We used the international Notzeus speech corpus database. The results from this work show that our recently proposed combinations between previous acc-CSP, \( \rho \)-CSP and \( \rho \)-CSPC improve the accuracy rate and the processing time.

We proved that acc-\( \rho \)-CSP and acc-\( \rho \)-CSPC outperforms popular GCC methods, with a higher accuracy rate. They are also able to estimate accurately even relatively longer delays, around 40% of the frame size length. The methods are also robust and offer accurate results for a large range of signal-to-noise ratio.
When acc-pCSP is calibrated adequately, it practically reaches the same accuracy as acc-pCSPC, but with almost 30% reduction of the processing time. Thus, we propose the acc-pCSP in TDE applications where processing speed and accurate estimations are very important. In this way it provides efficient results in various digital signal applications like echo canceling, speech enhancement, realign noisy signals, pattern detection, radar and sonar localization, medical and seismic processing.

Future work will involve development of acc-pCSPC and acc-pCSP applications for the VoIP environment. Specific analysis will also involve methods characterization for different system implementations.

REFERENCES


[24] NOIZEUS: A noisy speech corpus