Audio Engineering Society

Convention Paper

Presented at the 125th Convention
2008 October 2–5 San Francisco, CA, USA

Determination and correction of individual channel time offsets for signals involved in an audio mixture

Enrique Perez Gonzalez[1], and Joshua Reiss[1]

[1] Centre for Digital Music, Queen Mary, University of London, London, E1 4NS, England
enrique.perez@elec.qmul.ac.uk, josh.reiss@elec.qmul.ac.uk

ABSTRACT

A method for reducing comb-filtering effects due to delay time differences between audio signals in a sound mixer has been implemented. The method uses a multi-channel cross-adaptive effect topology to automatically determine the minimal delay and polarity contributions required to optimize the sound mixture. The system uses real time, time domain transfer function measurements to determine and correct the individual channel offset for every signal involved in the audio mixture. The method has applications in live and recorded audio mixing where recording a single sound source with more than one signal path is required, for example when recording a piano with multiple microphones. Results are reported which determine the effectiveness of the proposed method.

1. INTRODUCTION

It is common in recording or live mixing to use more than one microphone or signal path to record a source [1]. Although using multiple microphones can improve in some cases the sound characteristics of the source, it can also introduce artifacts in the form of destructive interference. For this reason it is of paramount importance to ensure all signal paths involved are synchronized while sharing compatible polarity. The reason for this is to avoid any undesired audible cancellation artifact in the audio signals. Common examples of mixing practices which can introduce audible interference due to differences in time arrival and polarity errors are:

-Using more than one microphone to record a drum set or recording a piano with more than one microphone.
-Recording an electric guitar / base using a direct box together with a microphone placed at the amplifier.
-Using a wireless signal, while simultaneously using a microphone to record the amplifier.
-Using a parallel digital sound effect or digital device next to an analogue or direct feed. This is a common practice in live sound when sending the digital effect return through a stereo channel.
-When using implementations of digital mixers or workstations that do not compensate for plug-in processing latency [2].
-Use of more than one microphone on a podium or stage.

All previous examples are common audio practice procedures which have destructive interference in the form of comb filtering. It is the aim of the authors to present a method that corrects these artifacts therefore we will begin with a review of the relevant concepts underlying comb filtering.

1.1. The Comb-Filter

It is well known from signal processing theory that the summation of two signals, which are highly correlated and have different time arrivals, when added together, results in a spectrum artifact known as comb filtering. Comb filtering is a time domain problem that affects the spectrum in a perceptible manner [3]. Figure 1 shows the comb filtering effect of the addition of white noise to audio signals with the same amplitude and a 1ms delay between each other.

The comb filter minima and maxima points are directly related to the delay between signals. Given that d is the delay time between signals, the first notch, F, is located at

$$F = 1/2d \qquad (1.)$$

and each successive minima will be located at odd multiples of F , while each successive maxima will be located at even multiples of F.

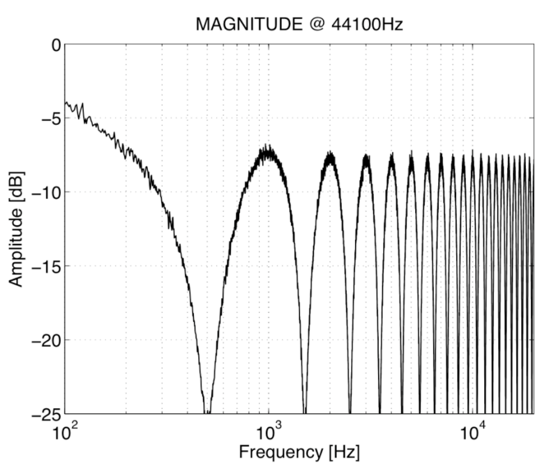

Figure 1 Comb Filtering of two white noise signals, both having the same amplitude, with a, 1ms delay between them.

The existence of comb filtering spectral artifacts in the audio signals is audible, and can make an audio engineer erroneously equalize the signal to improve its spectral texture. Unfortunately, due to its time domain nature, comb filtering is not equalizable and requires a time delay compensation to remove it. Finding the right amount of delay in a multi-channel mix that will minimize the comb filtering between tracks is not an easy task. For this reason we have devised and investigated a method that automatically detects the relationship between channels by determining the impulse response, with the aim of obtaining the minimal delay per channel required to minimize comb filtering.

1.2. Impulse Response

The impulse response of a system determines its dynamic response. Therefore it can be used to determine polarity of the system and delay times as well. As its name indicates, it is the time domain output resulting from inputing an impulse to a system.

Given a linear system such as the one represented by figure 2, where *x(t)* is the input of the system and *y(t)* is the output and *H* represents the transfer function of the system.

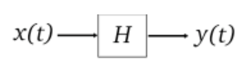

Figure 2 General diagram of a linear audio processing system.

Therefore we can approximate the transfer function in the frequency domain, *H(f)*, by using the following equation:

$$H(f) = \frac{FFT\{y(t)\}}{FFT\{x(t)\}} \qquad (2.)$$

Where the Fourier Transform (FFT) of the output is divided by the FFT of the input. Therefore by using FFT identities we can obtain the impulse response $\delta(t)$ of the system by applying an Inverse FFT (IFFT) to the transfer function of the system [4].

$$\delta(t) = FFT^{-1}\{H_a(f)\} \qquad (3.)$$

Whether $\delta(t)$ is positive or negative will determine the polarity of the system. Finally, given that the output of the system contains only pure delay with respect to the input we can determine the distance between the impulse response absolute maxima and the instant, t_0, when the impulse was input into the system by using the following equation:

$$\Delta(t) = t_0 - \delta(t) \qquad (4.)$$

Where $\Delta(t)$ is the delay time between $x(t)$ and $y(t)$.

Now given that $x(t)$ and $y(t)$ are two highly correlated signals, such as the signals of microphones used to record the same piano, but the first channel microphone is located at a different distance than the other channel microphone with respect to the sound source; then we can use the same method to determine the delay time between the two microphones. Therefore, for the purpose of this paper we will call $x(t)$ the reference channel and $y(t)$ the measured channel.

Unfortunately reverberation and noise will adversely affect the result of the calculation. Further modifications to this method to overcome these problems will be explained later in the implementation section of this paper. At this point it should be clear that for the method to work optimally the system will find the delay of one channel against a reference channel and the amount of reverberation and noise will limit the scope of the method.

If $\Delta(t)$ is positive it determines that the measurement channel is delayed by a amount $|\Delta(t)|$ with respect to the reference channel. While if $\Delta(t)$ is negative it determines that the reference channel is delayed by a amount $|\Delta(t)|$ with respect to the measurement channel.

Given that there are applications in audio where more than a pair of audio signals is used simultaneously, a mechanism to establish the relationship between the reference channel and the multiple measured channels should be established. This would mean that all delay determinations should share a common interdependent reference, therefore a relationship between all the measured channels and the reference channels must be established. In order to achieve this, a cross-adaptive architecture has been used to extend the impulse response delay determination method to work with multiple channels.

1.3. Cross-Adaptive Effects

A cross-adaptive effect is a signal-processing device, in which processing is dependent on the relationship between the existing channels in the mix and not just on the characteristics of one signal [5]. A general diagram of a cross-adaptive effect is presented in figure 3. The signal processing applied to each channel is dependent on a control vector *cv-n,* derived from the processing rules established inside the cross-adaptive feature processing devise, which is driven by the feature vectors, *fv-n*, extracted from the individual channel signals. More on this topic can be found in [6, 7].

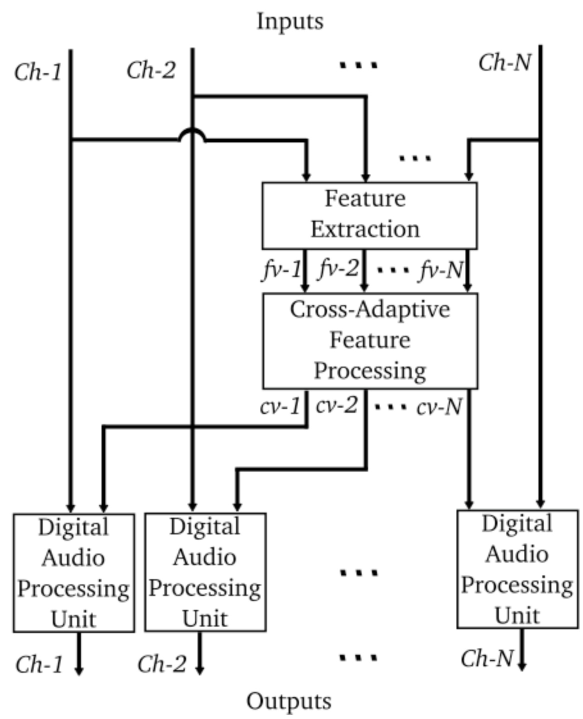

Figure 3 General cross-adaptive effect diagram. *ch-1* to *ch-N* are audio signals, *fv-1* to *fv-N* are the features extracted from the audio channels and *cv-1* to *cv-N* are the control vectors that drive the digital audio processors.

For the purpose of this paper the method makes use of a cross-adaptive processing topology in order to measures the features, delay and polarity, and established the interaction between channels with respect to a user specified reference signal. Thus the cross adaptive feature processing can establish the optimal solution to minimize the amount of delay added to synchronize all channels involved.

2. IMPLEMENTATION

The system consists of individual delay and polarity inverter units inserted on each channel. Each of these units is controlled by a control vector derived inside the cross-adaptive feature device which in this case is a delay polarity optimizer, The control vectors are derived by processing the feature vectors obtained from the interrelationship between the user selected reference channel and the other channels. Therefore the system aims to determine the optimal polarity and delay times to avoid comb filtering between channels. A depiction of the system flow diagram is presented in figure 4.

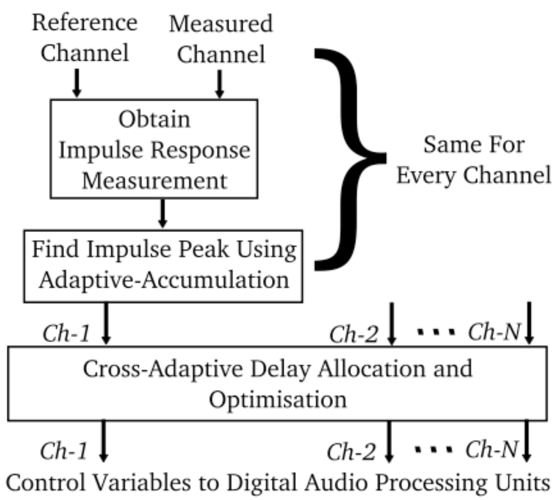

Figure 4 General algorithm flow diagram for an automatic mixture cross-adaptive time offset corrector.

2.1. Obtaining the Impulse

The researched method makes use of real time transfer function methodologies to determine the impulse response relationships between the channels involved in the cross-adaptive effect buss. A transfer function is determined for every channel involved with respect to the user selected reference channel. An inverse transform is then used to determine the individual impulse response of the sources. As mentioned obtaining a good approximation of the transfer function in a noisy or reverberant environment can be problematic. In order to obtain an unbiased estimate of the transfer function, $H_a(f)$, when the measurement channel has been contaminated with uncorrelated noise we must divide the auto-spectrum of the measured channel against the cross-spectrum of the reference channel, equation 5.

$$H_a(f) = \frac{FFT\{y(t)\}.FFT\{y(t)\}^*}{FFT\{y(t)\}.FFT\{x(t)\}^*} \qquad (5.)$$

Therefore when the measurement is contaminated by noise, the transfer function may be improved given that the noise is averaged out when performing the cross spectrum [8].

Once we obtain $H_a(f)$ we could probably proceed to apply an IFFT as shown in equation 3 in order to determine the impulse response. Unfortunately reverberation, can be treated as noise, which is correlated to some degree to the measurement channel, and can still have some undesired effects over the transfer function measurement. Therefore we borrow a technique commonly used for speech correlation that is known as the Phase Transform or PHAT. This is a weighing procedure in which equal emphasis is placed on each frequency. In other words, all frequency components are neglected and forced to have a unity value, $|H_a(f)|=1$, while taking into account only the phase information of the transfer function, $\measuredangle H_a(f)=\measuredangle H_a(f)$. This type of weighting tends to be sub-optimal under ideal conditions, but tends to be less susceptible to anomalous conditions, particularly to reverberation [9]. The resulting phasor equation for obtaining the phase dependent impulse response, $\delta_{PHAT}(t)$, is given by equation 6.

$$\delta_{PHAT}(t) = FFT^{-1}\{1\angle H_a(f)\} \quad (6.)$$

So given that we have applied a an equal weighing of frequencies together with a cross-spectrum noise averaging we can now assume that the corrected delay time, $\Delta_{PHAT}(t)$, between the reference channel and the source channel is given by the following equation.

$$\Delta_{PHAT}(t) = t_0 - \delta_{PHAT}(t) \qquad (7.)$$

The general block diagram of the procedure and the exact implementation used in this paper is presented in figure 5. The current implementation uses 1024 point FFTs with a Hann window with no overlap. All the system currently runs at a 44.1K sample rate. This means one sample is equivalent to 0.023ms.

In order to obtain meaningful results from equation 7, a mechanism for determining the signed magnitude of $\Delta_{PHAT}(t)$ needs to be performed. Subsection 2.2

describes a robust method to obtain the signed magnitude $\Delta_{PHAT}(t)$.

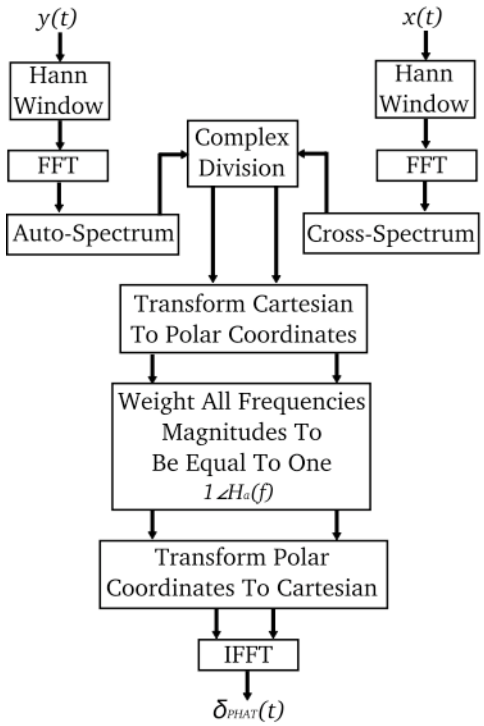

Figure 5 Robust impulse response measurement flow diagram, implemented using the PHAT method.

2.2. Adaptive Accumulation

In order to determine correctly the location of the signed magnitude of the impulse response, it is necessary to obtain its amplitude and position in time. This is done by a peak finder, which searches for the biggest absolute value inside a 1024 buffer, which is obtained from equation 7. Then the algorithm proceeds to store the corresponding signed magnitude for that value and the position where it was found, which is used to determine the delay time between the reference and the measured signal. Due to the fact that the impulse has been truncated by using finite length FFTs, the impulse obtained is a noisy signal in itself, and it is necessary to accumulate the signed amplitude using the following equation:

$$\delta_{Pa}(t_m) = \frac{\sum_{m=0}^{M} \delta_{PHAT}(t_a)}{M} \qquad (8.)$$

Where M goes from 0 at t_0 up to infinity at t_∞ and $\delta_{Pa}(t_m)$ corresponds to the accumulated signed magnitude of the impulse.

A similar accumulative approach was initially used to determine the delay position but unfortunately these proved slow and for low amplitudes it was impossible to determine the impulse position accurately. This was due to the fact that the peak finder will continuously accumulate noise peaks that were confused with the impulse, response peaks. In other words, the smaller the amplitude of the impulse the smaller the signal to noise ratio and therefore the more corrupted data gets stored into the time delay position accumulator. For this reason an adaptive-accumulative method for determining the delay position was devised.

The delay time calculation is adaptive because the amount of accumulations needed in order to output a valid number is adaptively increasing or decreasing in inverse proportion to the absolute magnitude of the impulse response. In other words if the signal to noise ratio is large, a small amount of accumulations are needed and if the signal to noise ratio is small, more accumulations are needed before a valid time delay position is output. Once valid data has been output then it can be sent into an accumulator similar to the one presented in equation 8. This adaptive accumulation is shown in equation 9.

$$\Delta_a(t) = \frac{\sum_{a=0}^{A} \Delta_{PHAT}(t_a)}{A} \qquad (9.)$$

where A is a function of the amplitude of the absolute maxima of the impulse response,

$$A = \mathrm{int}\left[\frac{K}{\left|\delta_{Pa}(t_m)\right|}\right] \qquad (10.)$$

and K has been chosen to be 2 in order to duplicate the number of minimum operations to validate the calculated delay time. An important implementation step in Equation 9 is that $\Delta_a(t)$ was chosen to be reset every time $\delta_{Pa}(t_m)$ changed in magnitude by a factor of $+/-10^{-2}$. The final equation for calculating the adaptive accumulated delay time feature, $\Delta_{am}(t)$, for every channel is presented in equation 11.

$$\Delta_{am}(t) = \frac{\sum_{M=0}^{M} \Delta_a(t_m)}{M} \qquad (11.)$$

Where $\Delta_{am}(t)$, is the result of the adaptive accumulation with respect to its amplitude.

This approach permitted the system to work at levels where the impulse response was practically buried in the noise while still being able to correctly determine its position; it also converged faster than pure accumulation. Figure 6, top, depicts the determination of the delay time with no accumulation. The lower plot in figure 6 shows the comparison between pure accumulation and adaptive accumulation. Notice that variability is reduced and the adaptive accumulation tends to escape to accumulated error in a faster manner. It also manages to converge faster than the pure accumulation approach.

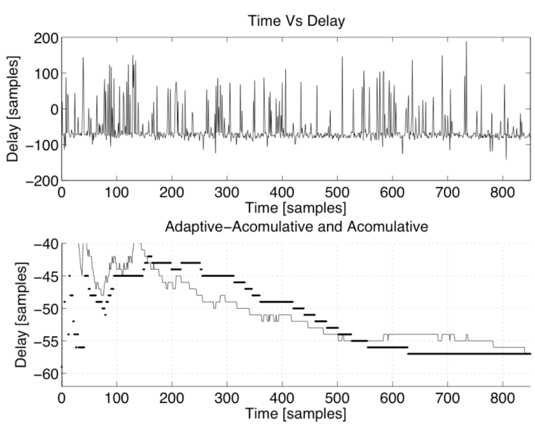

Figure 6 Non-accumulated $\Delta(t)$, (top). Comparison of accumulated $\Delta_a(t)$ in gray, vs. accumulative adaptive validation $\Delta_{am}(t)$ in black, (Bottom)

2.3. Cross-Adaptive Delay Optimization

During the cross-adaptive processing a minimization solution is obtained from the impulse response relationships of the channels involved with respect to the other channels. This gives the optimal delay time to reduce the comb filtering between the channels to be mixed. The algorithm calculates the impulse response for every channel with respect to the reference channel. The cross-adaptive algorithm scans the delay times for every channel and finds out if there are any negative delay values. If there are no negative delay time magnitudes the algorithm sends the delay compensation values to all individual channel-processing units, figure 7.

Figure 7 Individual channel processing unit user interface. The processing unit is driven by the implemented cross-adaptive feature processing.

In the case where negative delay times exist, the algorithm scans for the most negative delay value and finds the channel responsible for it. Once the channel responsible for the most negative delay has been found the algorithm sets it as the new reference channel and the whole process begins again. In this manner the algorithm is capable of offering an optimal delay solution for all interchannel delay dependencies. In the case of polarity issues, the cross-adaptive effect uses the signed magnitude of the amplitude of the reference impulse response in order to match the polarity of it to all other dependent channels. This means that if a channel has an inverted polarity with respect to the reference the algorithm will flip its polarity in order to obtain a constructive interaction between all channels.

The implementation of the cross-adaptive automatic mixture time offset corrector is presented in figure 8. The user has the ability to select the reference channel and the channels involved in the cross adaptive procedure. The user has the ability to bypass individually or overall corrections. A manual accumulator reset is also available. The top window shows the impulse response of the chosen reference channel against the chosen measured channel before correction. The lower window shows the impulse response of the chosen reference channel against the chosen measured channel after correction. The chosen measured channel field is a form of visual aid. The rest of the measurement channels involved in the process are being synchronized simultaneously.

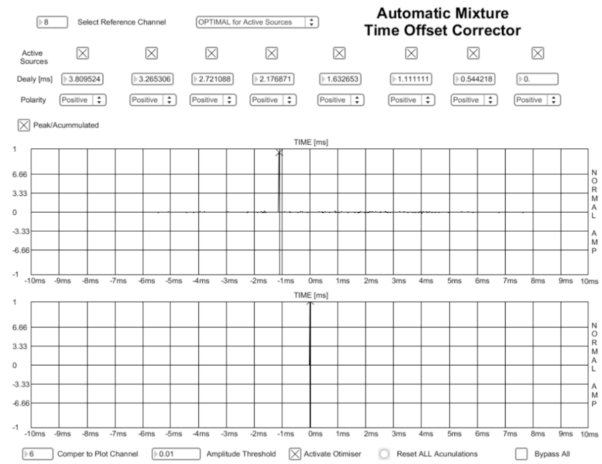

Figure 8 Master User interface of the implemented cross-adaptive time offset corrector.

3. TESTS AND RESULTS

In order to evaluate the robustness of the algorithm against noise and reverberation we proceeded to do the following experiment. Given a reference signal and a measurement signal with the same amplitude and content and synchronized at t_0, thus having an ideal impulse amplitude of one, we proceeded to add pink noise to the measurement channel, Figure 9 top. The noise was added in increments of 0.5 dB. Although the amplitude of the impulse response decreased when the noise was added, the system was able to keep track of the signal delay time at t_0 without a single sample error for pink noise up to a value smaller than 6dB. It was also found that for additive pink noise below -40 dB the effect on the measurements is negligible. Adding noise of amplitude 6dB grater than the signal proved completely impossible to track as the impulse completely disappeared in the background noise.

Next we proceeded to perform the same test but this time by adding reverberation. The reverberatior used is one of the most common implementations of the Schroeder and Moor reverberatior model called freeverb~ implemented by Olaf Matthes [10]. The settings for it were the default settings, which are:

Bypass: OFF
Room Size: 0.84
Damping: 50
With: 100
Wet level: 0dB

The only parameter varied was the dry level. In the case of freeverb~ a dry value of 0db means no reverberation has been mixed to the signal while a negative value represent a relative ratio of reverberation has been added. This means that a certain amount of relative reverberation has been mixed to the signal with respect to the relative level of the pure signal, figure 9 bottom. It was found that for added reverberation of up to -26dBs it was possible to track the impulse at t_0 without a +/-1 sample error and for added reverberation of -30 dB it was possible to track the impulse at t_0 with +/-2 sample accuracy.

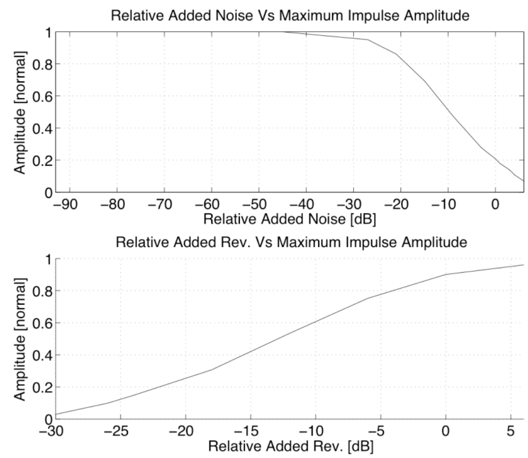

Figure 9 Impulse Response amplitude change due to the addition of noise (top). Impulse Response amplitude change due to the addition of reverberation (bottom). Measurements were performed for an impulse with no delay between reference and measured signal for a 0 sample error. The reverberation and noise were added to the measurement channel only.

It was also noticed early during the development of the algorithm, that a “windowing effect” occurred on the impulse response amplitude. This effect consisted in a reduction of the amplitude of the impulse, as the reference signal and the measured signal were pulled apart in time. Given that the two signals are exactly the same the algorithm should show a single impulse with unity amplitude at t_0, and this unit amplitude should be maintained even when the delay between the reference channel and the measured channel changes. Unfortunately this was not true and the rate of change of amplitude against the delay between the reference and the measured channels is depicted in figure 10 (top). The implication of this was that the correct calculation of the delay would be adversely affected as the delay time between the reference and the measurement channel changed. This is due to the fact that the impulse would be buried in the background noise causing the peak finder to erroneously take some noise peaks into account, figure 10 bottom. This is the main reason why the adaptive accumulative peek averaging method used for deriving the delay times performs better than standard accumulation. It was found that system was able to maintain a +/-2 sample accuracy for delay times up to 5.31ms with a +/-4 sample rate accuracy up to a delay time of 6.4ms. It is thought by the authors that time windowing $\Delta_{am}(t)$, could eliminate some of the problems associated with this windowing effect phenomena.

Once the system was characterized with the above experiments we preceded to test it with music. It was found that with pitched music where the reference signals and the measured signals are highly correlated the system tends to perform as expected within a +/-2 sample accuracy. The top plot of Figure 11 shows a piano signal that has been delayed with respect to the measurement channel by 4.76ms. Such delay displacement between the reference and the measured channel is extremely noticeable both in level and in spectral texture. Figure 11 bottom shows the impulse of the piano signals once both channels have been corrected by the cross-adaptive system. All highly correlated pitch signals such as this example performed in a similar manner. Polarity measurements were successfully corrected in all tests performed.

A second trial was performed with more difficult musical signals. An electric guitar was recorded directly with an analogue box and simultaneously recorded from the guitar amplifier, while containing a moderate amount of distortion. The performance of the system is shown in the top lot of figure 12. A 0.54ms delay error is found. But when the system added that amount of delay to correct it was unable to achieve full correction. After correction the system still showed a 6 sample error, equivalent to a 0.14ms error, as depicted in the bottom plot of figure 12. All polarity corrections were correctly identified in that test. The authors believe that if the algorithm was run recursively the system would be able to re-compensate the 0.14ms error and successively correct any further error.

The system performed poorly for non-pitched percussive sound such as drums and was unable to find a delay value. On the other hand, it managed to obtain the correct polarity for signals with inverted polarity. This was investigated mainly by using a snare, with one microphone placed on top and one on the bottom of it, thus having one microphone that would require polarity inversion.

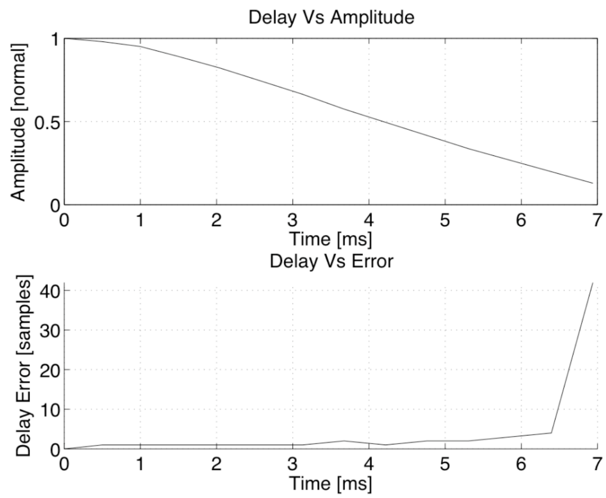

Figure 10 Impulse response amplitude windowing effect as a function of the delay offset between the reference channel and the measured channel (top). Delay calculation error as a function of the delay offset between the reference channel and the measured channel (bottom).

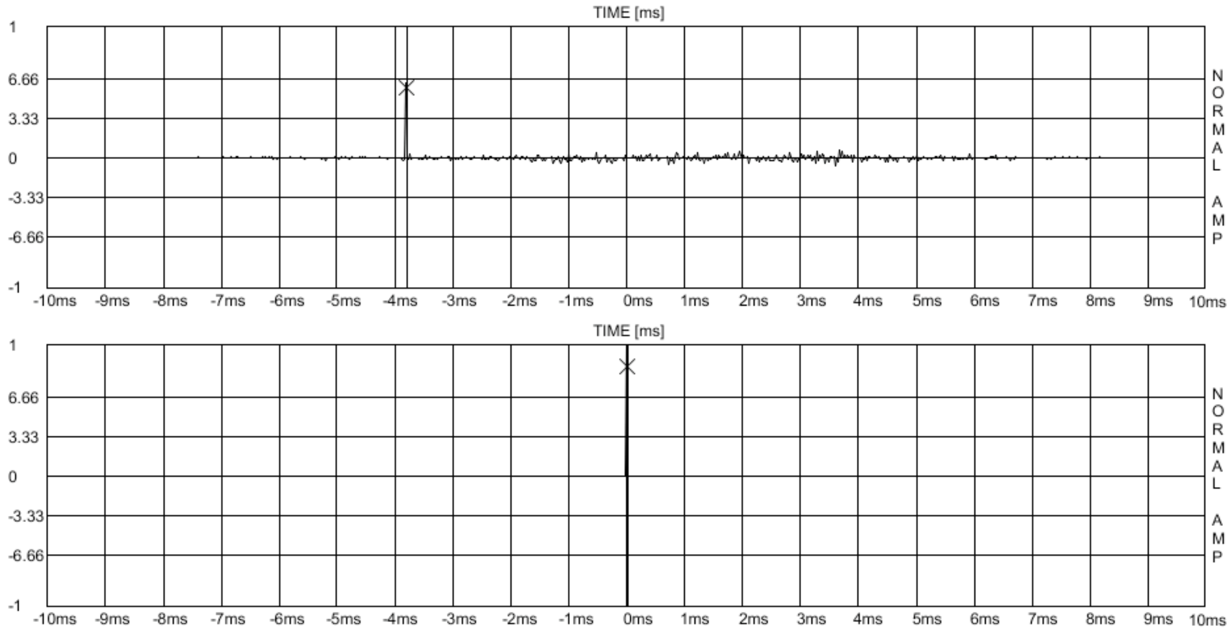

Figure 11 Measurements of impulse response of signal before correction (top) and after the correction (bottom). Measurements were made for a highly correlated signal.

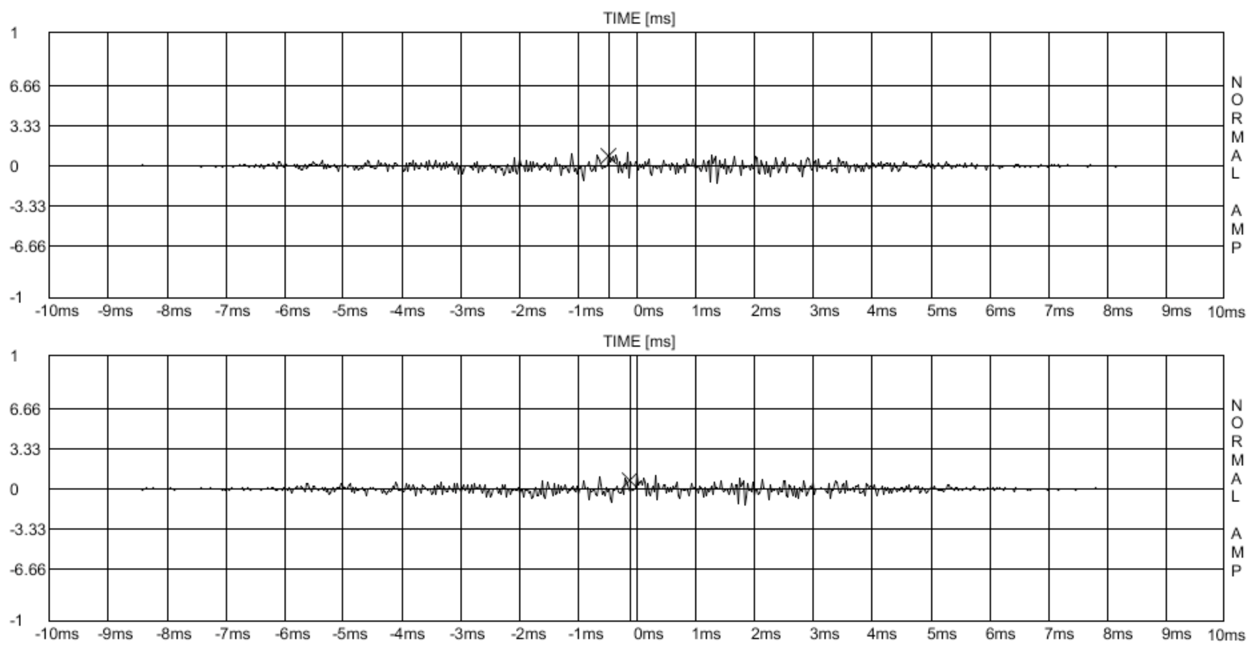

Figure 12 Measurements of impulse response of signal before correction (top) and after the correction (bottom). Measurements were made for a low correlated signal.

4. FURTHER STUDY

The study of a different method for identifying the delay between channels, such as determination of the maximum value of the cross-correlation, as often used in satellite navigation techniques, could prove beneficial to improve systems functionality. Decimation could also be used to expand the working time window of the system. The current system could be improved by the use of vector averaging techniques and coherence weighting techniques to improve the impulse response measurements. It is the current thought of the authors that the use of time windowing applied to current algorithm can correct the edge amplitude artifacts

The implementation of a delay finder algorithm, which is more robust and can track impulsive inputs, is under study. Finally it is thought that a recursive implementation of the algorithm will improve the overall performance of the algorithm for signals that are not highly correlated to each other.

5. CONCLUSIONS

A method for reducing comb-filtering effects due to delay time differences between audio signals in a sound mixer has been proposed and implemented. The results show that the algorithm is capable of correcting delay errors of +/-6.4ms with a +/-4 sample accuracy while optimizing the amount of delay to be used in the correction. The algorithm is also capable of optimizing the polarity settings for all channels involved in the cross-adaptive procedure. The system is functional within a +/-1 sample accuracy when the noise applied to one of the channels involved is less than 6dBs. The system was capable of maintaining the same accuracy for a reverberation mixture of up to -26dBs. So far the research has concluded that the algorithm is suitable for real time live multi-channel mixing and studio applications. The system performs better for highly correlated pitched signals than for impulsive percussive ones. The current system can autonomously correct common mixing problems due to polarity problems while achieving optimal time delay synchronization between channels.

6. ACKNOWLEDGEMENTS

Thanks to Andrew Robertson and Jules Espiau for recording some of the audio/music test signals used during the development of this research.

REFERENCES

[1] Shure, "Microphone Techniques, Live Sound Reinforsment," Shure AL1266H 10M 12/07 2007.

[2] SSL, "Duende Users Guide," SolidStateLogic, Ed. UK: 82S6MC060A, 2008, p. 9.

[3] M. G. Ballow, et. al., *Handbook for Sound Engineers*, Third Edition 2002 ed.: Focal Press / Elsavier, 2002.

[4] J. Meyer, "Equalization Using Voice and Music as the Sources," in *76th Audio Engineering Society Convention* New York, 1984.

[5] E. Perez_Gonzalez, and J. Reiss, " Improved control for selective minimization of masking using interchannel dependency effects," in *DAFx* Helsinki-Finland, 2008.

[6] V. Verfaille, and et al, "Adaptive Digital Audio Effects (A-DAFx): A New Class of Sound Transformations," *IEEE Transactions On Audio, Speech, and Language Processing,* vol. 1558-7916, 2006.

[7] E. Perez_Gonzalez, and J. Reiss, "Automatic mixing: live downmixing stereo panner," in *DAFx* Bordeaux-France, 2007.

[8] J. Meyer, "Precision Transfer Function Measurments Using Program Material as the Excitation Signal," in *11th Audio Engineering Society Iternational Conference*.

[9] M. S. Brandstein, and Silverman H. F., "A Robust Method for Speech Signal Time-Delay Estimation in Reverberant Rooms," *in Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing,* pp. 375-378, 1997.

[10] M. Matthes, "Freeverb~ Schroeder / Moorer reverb model," Version 1.1 ed: http://www.akustische-kunst.org/maxmsp/, 2003.