

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Perceptual Echo Control and Delay Estimation

Kirill Sakhnov, Ekaterina Verteletskaya and Boris Simak
*Czech Technical University in Prague
Czech Republic*

1. Introduction

Echo phenomenon has been always existed in telecommunications networks. Generally it has been noticed on long international telephone calls. As technology advances and the data transmission methods tend more to packet-switching concepts, the traditional echo problem remained up-to-date. An important issue in echo analysis is a round-trip delay of the network. This is a time interval required for a signal from speaker's mouth, across the communication network through the transmit path to the potential source of the echo, and then back across the network again on the receive path to the speaker's ear. The main problem associated with IP-based networks is that the round-trip delay can be never reduced below its fundamental limit. There is always a delay of at least two to three packet sizes (50 to 80 ms) (Choi et al., 2004) that can make the existing network echo more audible (Gordy & Goubran, 2006). Therefore, all Voice over IP (VoIP) network terminals should employ echo cancellers to reduce the amplitude of returning echoes. A main parameter of each echo canceller is a length of its coverage. The coverage means the length of time that the echo canceller stores its approximation in memory. The adaptive filter should be long enough to model an unknown system properly, especially in case of VoIP applications (Nisar et al., 2009; Youhong et al., 2005). On the other hand, it is known that an active part of the network echo path is usually much smaller compared to the whole echo path that has to be covered by the adaptive filtering algorithm. That is why the knowledge of the echo delay is important for using echo cancellers in packet-switching networks. Today, there is a wide family of adaptive filtering algorithms that can exploit sparseness of the echo path to reduce high computational complexity associated with long echo paths (Dyba, 2008; Hongyang & Dyba, 2008; Khong & Naylor, 2006; Hongyang & Dyba, 2009). In this chapter, we discuss numerous methods used for estimation of echo delay. Algorithms based on cross-correlation function and adaptive filters are used in the art. We will consider both types of them, discuss their advantages and drawbacks. Afterwards, we will pay our attention to the adaptive filtering techniques. We provide a study on different partial, proportionate, sparseness-controlled time- and frequency-domain adaptive filters. The readers will get closer to an issue of echo cancellation, which is relevant in nowadays telecommunications networks. Ones will able to recognize important features and particular areas of implementation of various adaptive algorithms. Further, we are giving a short introduction to the issue of echo control for telecommunications networks. This description emphasises on two most important aspects of perceptual echo control, which are echo loudness and echo delay.

1.1 Echo control issue

In the very beginning of the telephone age, all calls were made through an analog pair of copper wires. The technology has progressively moved to digital circuit switched networks over the past several decades. Today most of the phone traffic is handled by the Public Switched Telephone Network (PSTN), which provides end-to-end dedicated circuits. During the last years a move to packet-switched networks has been initiated to support voice traffic over Internet Protocol (IP). The main reason for the move from circuit-switched voice networks to packet-switched networks is to enable convergence between data services and voice services. It is of economical interest to be able to use the same equipment for voice and data traffic. Reduced cost of placing a phone call is another reason, since the voice-packet is treated and routed much in the same way as any other data packet (note that Quality of Service plays a vital role in this process). Thus, conventional long distance tariffs have a tendency to be completely eliminated in Voice over IP (VoIP) networks as well.

Echo issue has long been recognized as a problem on telecommunications networks, though generally it has been noticed mostly on international telephone calls or when using speaker phones. As technology advances and the information transmission methods tend more to packet-switching concepts, the traditional echo problem should be reviewed and updated. Previously unconsidered factors now play an important part in the echo characteristics. This section describes the echo delay problem, which is often encountered in packet-switched networks. This problem is highlighted in relation to VoIP networks. More specific details on the process of locating and eliminating echoes are included in conclusion to the chapter.

Consider a simple voice telephone call, where an echo occurs when you hear your own voice repeated. An echo is the audible leak-through of your own voice into your own receive path. Every voice conversation has always at least two participants. From the perspective of each participant, there are two voice paths in every call:

- Transmit path - The transmit path is usually depicted as Tx path. In a conversation, the transmit path is created when any person begins speaking. The sound is transmitted from the mouth of the speaker to the ear of the listener.
- Receive path - The receive path is also called the return and depicted as Rx path. In a conversation, the receive path is created when a person hears the conversation coming from the mouth of another speaker.

Fig. 1 illustrates a simple diagram of a voice call between two persons A (Kirill) and B (Kate). From the user A's perspective, the Tx path carries his voice to the user B's ear, and the Rx path carries the user B's voice to the user A's ear.

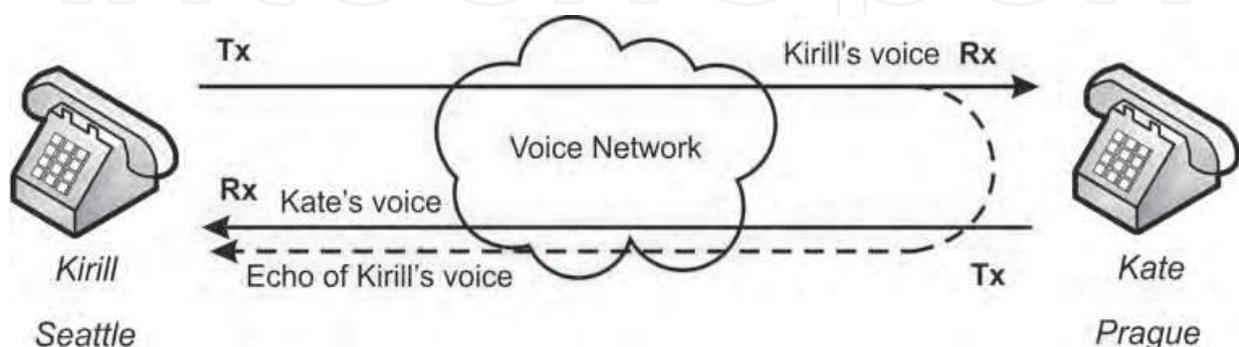


Fig. 1. A simple telephone call scenario

There is one significant factor in the echo analysis, and especially for the packet-switching networks. It is a round-trip delay of the voice network. The round-trip delay is the length of time required for an utterance from the user A's mouth, across the network on the Tx path to the source of the leak, and then back across the network again on the Rx path to the user A's ear. Let's define two important statements about echo nature, which are the following:

- The louder the echo (echo amplitude), the more annoying it is,
- The longer the round-trip delay (the "later" the echo), the more annoying it is.

Table 1 shows how time delay can affect the quality of a voice conversation.

One-Way Delay Range (ms)	Effect on Voice Quality
0-25	This is the expected range for national calls. There are no difficulties during conversation.
25-150	This is the expected range for international calls using a terrestrial transport link and IP telephony, which includes only one instance of IP voice. This range is acceptable for most users, assuming the use of echo control devices.
150-400	This is the expected range for a satellite link. Delays in this range can interrupt the normal flow of a conversation. A high-performance echo canceller must be used and careful network planning is necessary.
Greater than 400	This is excessive delay and must be avoided by network planning.

Table 1. Effect of Delay on Voice Quality

Fig. 2 shows how the echo disturbance influenced by the two parameters: delay and echo level. The metric called Talker Echo Loudness Rating (TELR) denotes the level difference between the voice and echo signals. The "acceptable" curve represents the limit for acceptable talker echo performance for all digital networks. The fact that the speaker A, in Fig. 1, hears an echo illustrates one of the basic characteristics of echo: perceived echo most likely indicates a problem at the other end of the call. The problem that is producing the echo that A hears, the leakage source, is somewhere on B's side of the network. If the person B was experiencing echo, the problem would be on the user A's side.

The perceived echo usually originates in the terminating side of the network for the following two reasons:

- Leakage happens only in analog circuits. Voice traffic in the digital portions of the network does not pass from one path to another.
- Echo arriving after very short time, about 25 milliseconds, is generally imperceptible, because it is masked by the physical and electrical side-tone signal.

A hybrid transformer is often main source of the electrical signal leakage. The typical analog telephone terminal is 2-wire device: a single pair of conductors is used to carry both the Tx and Rx signals. For analog trunk connections, known as 4-wire transmission, two pairs of conductors carry separate Tx and Rx signals. Digital trunks (T1/E1) can be virtual 4-wire links because they also carry separate Tx and Rx signals. A hybrid is a transformer that is used to interface 4-wire links to 2-wire links. Fig. 4 shows a hybrid transformer in an analog tail circuit. Because a hybrid transformer is a non-ideal physical device, a certain fraction of the 4-wire incoming (Rx) signal will be reflected into 4-wire outgoing (Tx) signal. A typical

fraction for a properly terminated hybrid in a PBX is about -25 decibels (dB), meaning that the reflected signal (the echo) will be a version of the Rx signal attenuated by about 25 dB. For a PSTN POTS (Plain Old Telephone Service) termination, the expected value is between 12 and 15 dB. Echo strength is expressed in dB as a measurement called Echo Return Loss (ERL). Therefore, an ERL of 0 dB indicates that the echo is the same amplitude as the original source. A large ERL indicates a negligible echo. Remember that an echo must have both sufficient amplitude and sufficient delay to be perceived. For local calls with one-way delay from 0 to 25 ms, an echo of strength of -25 dB relative to the speech level of the talker is generally quiet enough to not be annoying. For a one-way delay in the range of 25 to 150 ms, the ERL should exceed 55 dB to eliminate the perception of echo from the end-user perspective, as recommended in ITU-T recommendation G.168 on echo cancellation (ITU-T G.168, 2002). In this case echo cancellation is required.

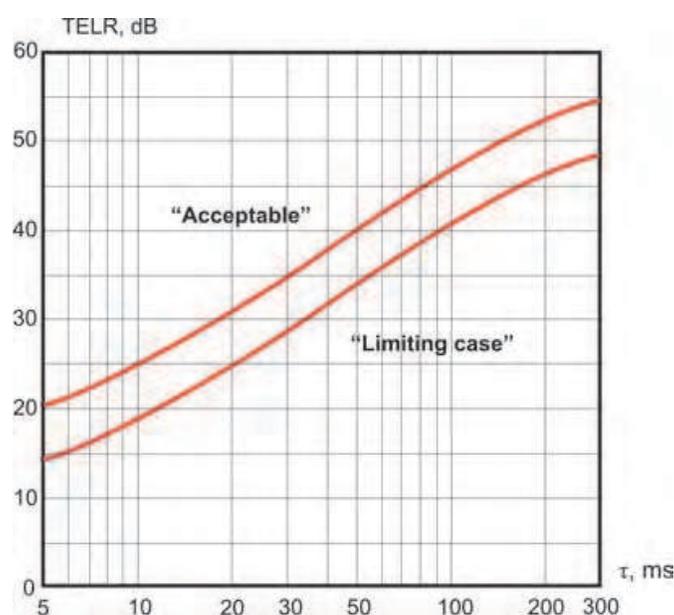


Fig. 2. Talker echo tolerance curves (ITU-T G.131, 2003)

2. Echo delay estimation using cross-correlation

The following section presents a study of cross-correlation-based Time Delay Estimation (TDE) algorithms. The main purpose is to analyze a number of methods, in order to find the most suitable one for real-time speech processing. As TDE is an important topic during transmission of voice signals over packet-switching telecommunication systems, it is vital to estimate the true time delay between Tx and Rx speech signals. We consider algorithms processing both in time- and frequency domains. An echo delay problem associated with IP-based transport networks is also included into the discussion. An experimental comparison of the performance of numerous methods based on cross-correlation, normalized cross-correlation and a generalized cross-correlation function is presented.

2.1 General scenario of delay estimation using cross-correlation functions

The known problem associated with IP-based networks is that the round-trip delay can be never reduced below its fundamental limit. There is always a delay of at least two to three packet sizes (50 to 80 ms) that can make the existing network echo more audible. Therefore, all

Voice over IP (VoIP) network terminals should employ echo cancellers to reduce the amplitude of returning echoes. A main parameter of each echo canceller is a length of coverage. Echo canceller coverage specifies the length of time that the echo canceller stores its approximation in memory. The adaptive filter should be long enough to model an unknown system properly, especially in case of VoIP applications. On the other hand, it is known that the active part of the network echo path is usually much smaller compared to the whole echo path that has to be covered by the adaptive filtering algorithm inside the echo canceller. That is why the knowledge of the echo delay is important for using echo cancellers in packet-switching networks successfully. In general, every communications system includes a communications network and communications terminals on the both sides of the network. The communications terminals could be telephones, soft phones, and wireless voice communication devices. Fig. 3 illustrates how an echo assessment device can be arranged into the defined system. The echo delay estimator has to monitor two parallel channels. An outgoing voice channel transmits an original voice waveform from the first terminal through the communications network to the second terminal. An incoming voice channel receives an echo waveform of the original signal returning from the second terminal through the communications network back. This is a delayed and attenuated version of the original voice signal.

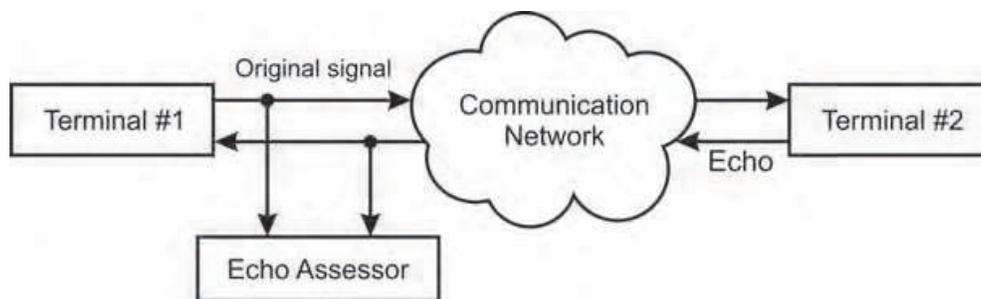


Fig. 3. Arrangement of echo assessment module in the network

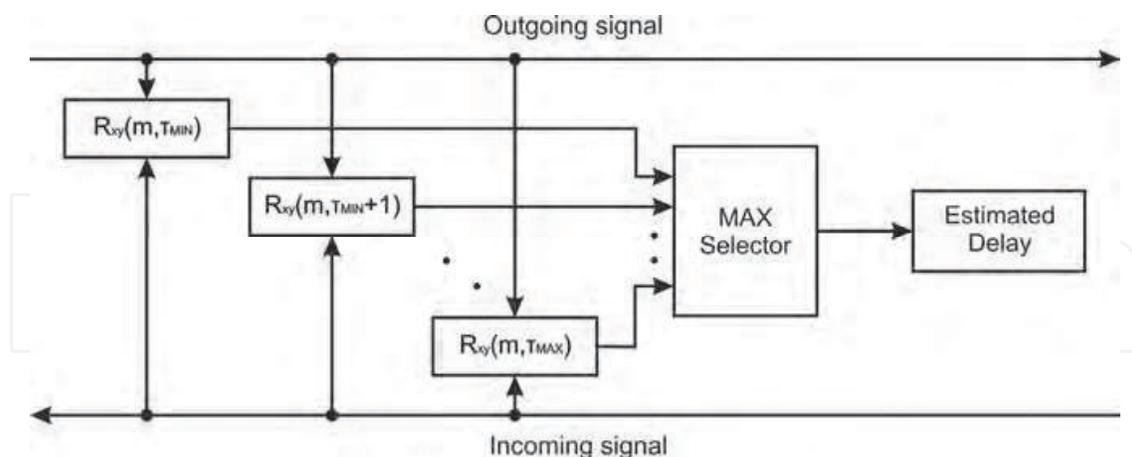


Fig. 4. General block diagram of delay estimator

Fig. 4 illustrates a general block diagram of the echo delay estimator. The echo delay estimator computes correlation between two voice channels for different set of delays in parallel manner (Carter, 1976). The delay-shift with the largest cross-correlation coefficient is selected as the delay estimate. Fig. 5 illustrates, in a flowchart form, steps performed when implementing a method of echo delay estimation utilizing cross-correlation algorithms. Once started from block 1, block 2 calculates the cross-correlation function for a buffer of

input samples of the Rx and Tx signals. Block 3 utilizes cross-correlation coefficients to compute the similarities between the transmitted signal and the received signal over a range of delays. For each particular delay, the similarity is obtained. Once the similarities have been determined for each delay within the range of delays, block 4 chooses a delay that produces the greatest similarity metric for the given input frames. Consequently, block 5 indicates that the estimation process is completed.

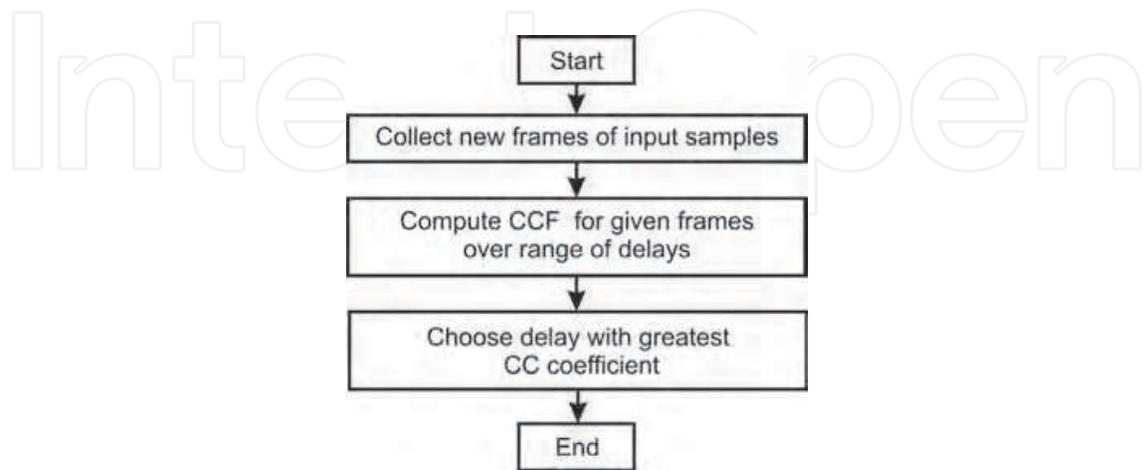


Fig. 5. Flowchart for estimating echo delay value

2.2 Algorithms proceeding in time-domain

Time domain implementation of Cross-Correlation Function (CCF) and Normalized CCF (NCCF) is presented. The cross-correlation function for a successive pair of speech frames can be estimated by (Mueller, 1975)

$$\begin{aligned}
 R_{xy\tau MIN}(m) &= \sum_{n=D}^{D+L-1} x(n) \cdot y(n-m) & n = 0, \dots, L-1, m = 0, \dots, L-1 \\
 &\dots \\
 R_{xy\tau MAX}(m) &= \sum_{n=D}^{D+L-1} x(n) \cdot y(n-m) & n = 0, \dots, L-1, m = 0, \dots, L-1
 \end{aligned} \tag{2}$$

Here, $x(n)$ simply denotes a frame of the outgoing signal, $y(n)$ is related to a frame of the incoming signal. According to Fig. 4, the estimation of the CCF is done for a supposed range of delays. The time-shift, τ , which is always in range of $[\tau_{min}; \tau_{max}]$ and causes the maximal peak value of the CCF is declared as an estimate of the true echo delay T_D . Similarly to the CCF, an estimate of the NCCF is done (Buchner et al., 2006)

$$\begin{aligned}
 R_{xy\tau MIN}^n(m) &= \frac{\sum_{n=D}^{D+L-1} x(n) \cdot y(n-m)}{\sqrt{E_x \cdot E_y}} & n = 0, \dots, L-1, m = 0, \dots, L-1 \\
 &\dots \\
 R_{xy\tau MAX}^n(m) &= \frac{\sum_{n=D}^{D+L-1} x(n) \cdot y(n-m)}{\sqrt{E_x \cdot E_y}} & n = 0, \dots, L-1, m = 0, \dots, L-1
 \end{aligned} \tag{3}$$

Here, E_x and E_y denotes a short-term energy of the outgoing and the incoming frames. These values are calculated using the following equations

$$E_x = \sum_{n=D}^{D+L-1} x^2(n) \quad (4)$$

$$E_y = \sum_{n=D}^{D+L-1} y^2(n-m) \quad (5)$$

Let us further consider generalized cross-correlation algorithms, which operate in the frequency domain (Youn et al., 1983; Zetterberg et al., 2005).

2.3 Algorithms proceeding in frequency-domain

More sophisticated way how to provide TDE is to compute the cross-correlation function in the frequency domain. This process in literature is called Generalized Cross-Correlation (GCC) (Hertz, 1986). The idea behind this method is to perform pre-filtering of the input signals before calculating CCF. It makes possible to improve the accuracy of delay estimation. Note that the filtering procedure is performed in the frequency domain. Let us describe this process in more details. It is well known, that the simple cross-correlation function, R_{xy} , between signals $x(n)$ and $y(n)$ is related to the cross-power density function (cross-power spectrum), G_{xy} , by the general inverse Fourier transform relationship, as

$$R_{xy}(m) = \int_{-\infty}^{\infty} G_{xy}(f) \cdot e^{j2\pi fm} df \quad (6)$$

When $x(n)$ and $y(n)$ have been filtered with filters having transfer functions $H_1(f)$ and $H_2(f)$, the cross-power spectrum between the filter out-puts is given by

$$G_{xy}^g(f) = H_1(f) \cdot H_2^*(f) \cdot G_{xy}(f) \quad (7)$$

Consequently, the Generalized Cross-Correlation Function (GCCF) between $x(n)$ and $y(n)$ is given by (Knapp & Carter, 1976)

$$R_{xy}^g(m) = \int_{-\infty}^{\infty} \Psi_g(f) \cdot G_{xy}(f) \cdot e^{j2\pi fm} df \quad (8)$$

$$\Psi_g(f) = H_1(f) \cdot H_2^*(f) \quad (9)$$

Here, Ψ_g , is a generalized weighting function. Table 2 represents weighting functions that were used for experiments with speech signals (Wilson & Darrell, 2006).

The parameter γ_{xy} denotes a complex coherence function. It can be calculated as (Tianshuang & Hongyu, 1996)

$$\gamma_{xy}(f) = \frac{G_{xy}(f)}{\sqrt{G_{xx}(f) \cdot G_{yy}(f)}} \quad (10)$$

$$G_{xx}(f) = \int_{-\infty}^{\infty} R_{xx}(m) \cdot e^{j2\pi fm} dm \quad (11)$$

$$G_{yy}(f) = \int_{-\infty}^{\infty} R_{yy}(m) \cdot e^{j2\pi fm} dm \quad (12)$$

Here, $G_{xx}(f)$ and $G_{yy}(f)$ are auto-power spectra of the outgoing and the incoming signal; $R_{xx}(m)$ and $R_{yy}(m)$ are auto-correlation functions of the same signals. Fig. 6 illustrates a block diagram of the implemented generalized cross-correlation algorithm, where the Fast Fourier Transform (FFT) is used for auto-spectra and cross-spectrum calculation. After the cross-power spectrum is estimated, it is multiplied by the corresponding GCC weighting function. The inverse FFT is used for obtaining the time domain generalized-cross correlation function. This operation is repeated for the specified range of possible delays. After the whole process has completed, the time shift with maximum corresponding peak value is declared as an estimation of the true delay.

Processor Name	Weighting Function
Cross-correlation	1
ROTH	$1/G_{xy}(f)$
SCOT	$1/\sqrt{G_{xx}(f) \cdot G_{yy}(f)}$
PHAT	$1/ G_{xy}(f) $
Cps-m	$1/m \sqrt{G_{xx}(f) \cdot G_{yy}(f)}$
HT (ML)	$\frac{ \gamma_{xy}(f) ^2}{ G_{xy}(f) \cdot (1 - \gamma_{xy}(f) ^2)}$
Eckart	$\frac{ G_{xy}(f) }{(G_{xx}(f) - G_{xy}(f)) \cdot (G_{yy}(f) - G_{xy}(f))}$
HB	$ G_{xy}(f) /G_{xx}(f) \cdot G_{yy}(f)$
Wiener	$ \gamma_{xy}(f) ^2$

Table 2. Various GCC weighting functions

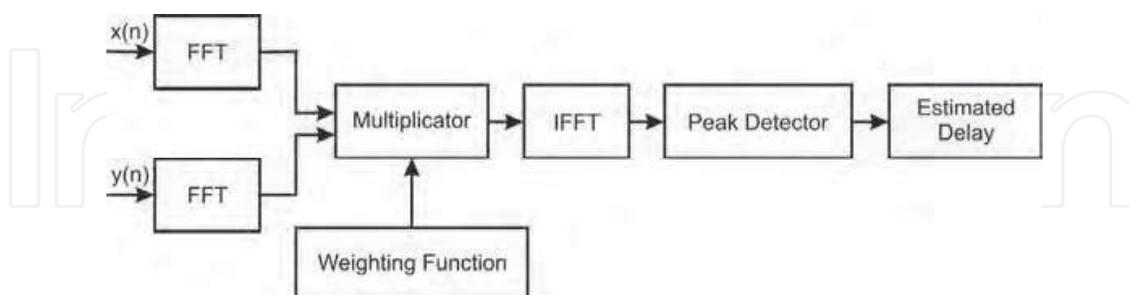
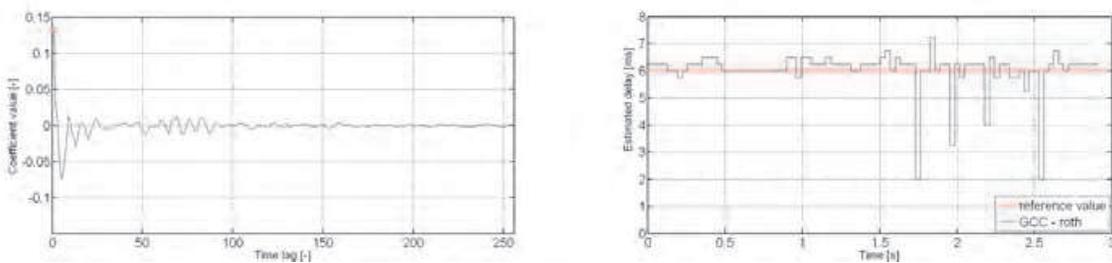


Fig. 6. Diagram of the implemented generalized cross-correlation algorithm

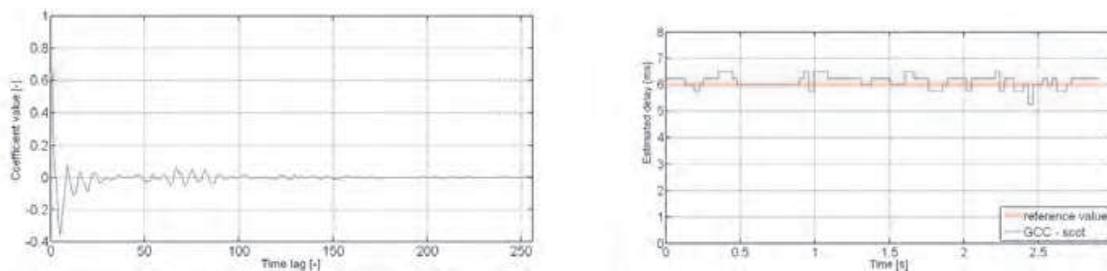
2.4 Discussion over experimental results

We used MATLAB software as a simulation environment. The time difference between time when the outgoing signal leaves the voice terminal and moment when the incoming signal containing the echo of the original signal arrives back from the network is referred to as a true echo delay. This value for the first three figures that are presented below equals 6ms (48 samples). For the purpose of TDE it is also necessary to specify time interval through which

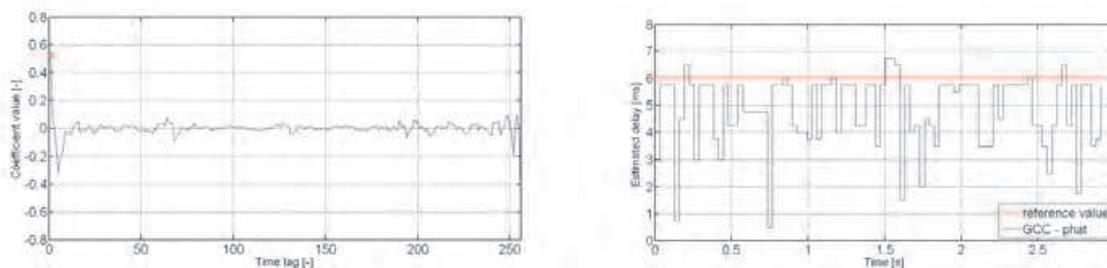
the value of the true delay is searched. To cover the 6ms delay we choose the interval between 0 and 10ms what corresponds to the maximum delay value of 60 samples. Afterwards we present the estimation results for a group of different delays. It helps to understand better performance of the algorithms. Unfortunately, because of the non-stationary nature of human speech, the CCF is not reliable for all situations. Its performance highly depends on numerous factors, i.e. signal strength, signal-to-noise ratio (SNR), etc (Chen et al., 2006). The NCCF is not so sensitive to the sudden changes in the signal's amplitude. It outperforms the CCF when we work with low level signals. The advantages of the algorithms proceeding in the frequency domain compared to the algorithms operating in the time domain are accuracy and reduced computational complexity. Fig. 7 illustrates the outputs of the GCC algorithms, which were presented in Table 2.



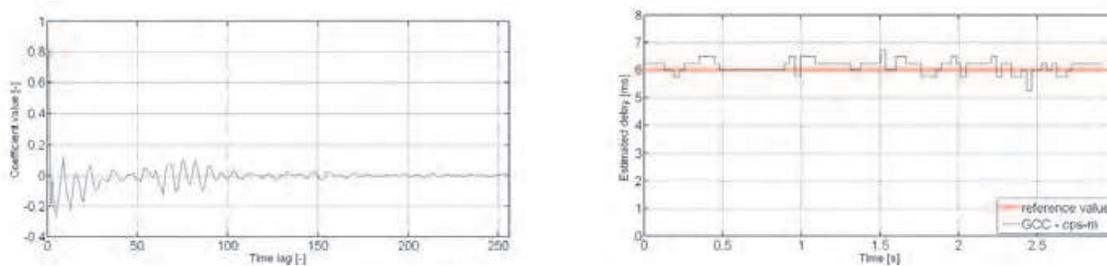
(a) ROTH - weighting function



(b) SCOT - weighting function



(c) PHAT - weighting function



(d) CPS-M (M=2) - weighting function

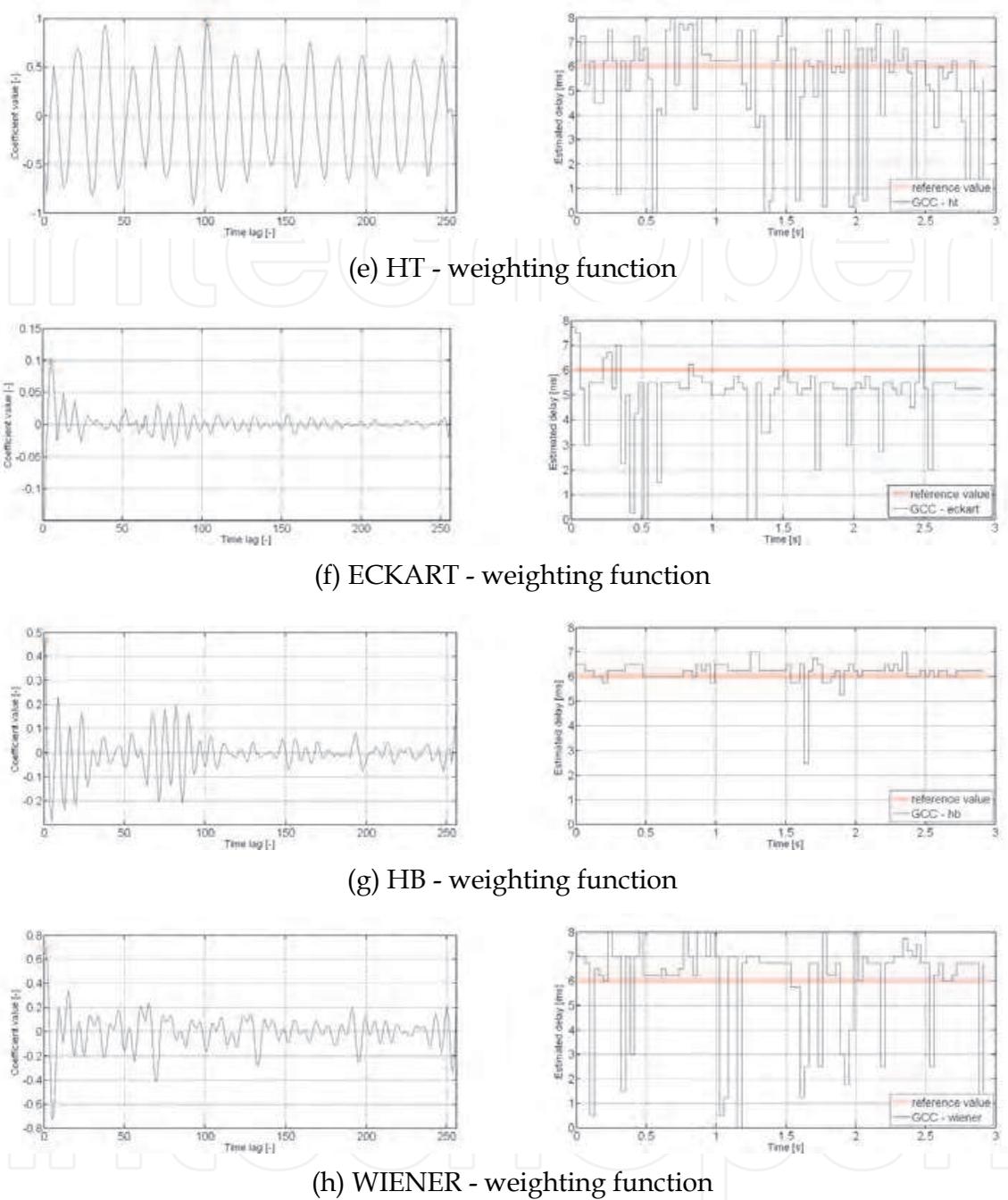


Fig. 7. Time delay estimation using GCCF

Table 3 and 4 provides us along with the following results. The joint comparison was done in terms of the estimation accuracy of the algorithms. The group of delays was chosen for this experiment. Delay values are consistent with the ones referenced in the corresponding ITU-T recommendation G.131 (ITU-T G.131, 2003). Once the respective cross-correlation function was calculated, its maximum peak value is detected using the searching procedure described in Fig. 4. SCC is related to the Standard CC function.

[ms]	SCC	ROTH	SCOT	PHAT	CPS-2	HT	ECKART	HB	WIENER
5	4,9	5,1	5,2	3,7	5,2	4,4	4,2	5,2	5,2
10	9,7	10,3	10,3	7,5	10,3	8,8	8,3	10,4	10,3
20	19,5	20,6	20,7	15,0	20,7	17,7	16,7	20,8	20,7
30	29,2	30,9	31,0	22,4	31,0	26,5	25,0	31,2	31,0
50	48,7	51,4	51,6	37,4	51,6	44,2	41,7	52,1	51,7
100	97,3	102,9	103,3	74,8	103,3	88,5	83,3	104,2	103,4
200	194,6	205,7	206,6	149,6	206,6	177,0	166,6	208,3	206,8
300	292,0	308,6	309,9	224,4	309,9	265,4	249,9	312,5	310,3

Table 3. Mean values of estimated delays

[ms]	SCC	ROTH	SCOT	PHAT	CPS-2	HT	ECKART	HB	WIENER
5	4,9	5,1	5,2	3,7	5,2	4,4	4,2	5,2	5,2
10	9,7	10,3	10,3	7,5	10,3	8,8	8,3	10,4	10,3
20	19,5	20,6	20,7	15,0	20,7	17,7	16,7	20,8	20,7
30	29,2	30,9	31,0	22,4	31,0	26,5	25,0	31,2	31,0
50	48,7	51,4	51,6	37,4	51,6	44,2	41,7	52,1	51,7
100	97,3	102,9	103,3	74,8	103,3	88,5	83,3	104,2	103,4
200	194,6	205,7	206,6	149,6	206,6	177,0	166,6	208,3	206,8
300	292,0	308,6	309,9	224,4	309,9	265,4	249,9	312,5	310,3

Table 4. Root mean square deviation of estimated delays

The abscissa of the largest peak value is the estimated delay. Note that 50 trial speech records for each processor were evaluated to obtain the mean value and the Root Mean Square Deviation (RMSD) parameter (Anderson & Woessner, 1992). Not only different speech signals, but various hybrid impulse response models have been used. The results for delays from 5 to 300 ms are presented in the corresponding tables. Table 3 contents the mean values, whether Table 4 illustrates the estimated RMSD values.

3. Echo delay estimation using adaptive filters

In this section, we introduce methods for extracting an echo delay between speech signals using adaptive filtering algorithms. We know that time delay estimation is an initial step for many speech processing applications. Conventional techniques that estimate a time difference of arrival between two signals are based on the peak determination of the generalized cross-correlation between these signals. To achieve a good precision and stability in estimation, the input sequences have to be multiplied by an appropriate weighting function. Regularly, the weighting functions are dependent on the signals power spectra. The spectra are generally unknown and have to be estimated in advance. An implementation of the time delay estimation via the adaptive least mean squares is analogous to estimating the Roth generalized cross-correlation weighting function. The estimated parameters using the adaptive filter have a smaller variance, because it avoids the need for the spectrum estimation. In the following, we discuss proportionate and partial-update adaptive techniques and consider their performance in term of delay estimation.

Time delay estimation (TDE) has always been and remains a popular research topic. It is known, that the main problem associated with IP-based networks is the round-trip delay, which can never be reduced below its fundamental limit. A number of efforts were made in order to improve the TDE precision. Various methods based on the Generalized Cross-Correlation (GCC) were proposed. The GCC algorithms arrange a pre-filter to obtain the modified signal spectrum for optimal time delay estimation. To specify the filter's characteristic, it requires a priori knowledge of the statistics of the received signals. However, the efficiency of the algorithms decreases considerably when little or no prior knowledge about the signal statistics is known. From the time when Widrow proposed an adaptive filtering technique based on Least Mean Squares (LMS) (Widrow, 2005; Haykin, 2001), an adaptive theory also found an application to delay estimation. An adaptive implementation of the time delay estimation via Widrow's LMS algorithm is usually referred to as TDLMS. Comparing to the GCC algorithms, the adaptive filtering techniques do not require a priori information of the signal statistics, because the estimation of the signal spectrum is no longer needed. The adaptive filtering algorithms determine the time delay in an iterative manner. There are comparative studies, which provide comparison of the LMS versus the generalized cross-correlation (Zetterberg et al., 2005). Generally, the time domain implementation of any adaptive filter is associated with high computational complexity. It directly depends on the length of the adaptive filter. In order to reduce the computational load of the TDLMS (Emadzadeh et al., 2008), we offer using adaptive filtering algorithms with reduced computational complexity. They provide savings comparing to the conventional adaptive algorithms. In the following, we discuss each of the algorithms in greater details. First, a general scenario for the adaptive time delay estimation using a simple Normalized Least Mean Squares (NLMS) adaptive filtering algorithm is presented. Afterwards, we introduce the proportionate and partial-updated algorithms proceeding in the time domain. A new partial-updated proportionate NLMS algorithm is outlined. A comparison between the TDE algorithms is made in context of the network echo delay estimation.

3.1 General scenario of delay estimation using adaptive filters

Traditionally, the NLMS algorithm is used for the echo canceller implementation. It applies a Finite Impulse Response (FIR) adaptive filter with adjustable weights for modelling the unknown echo path's impulse response. The NLMS algorithm minimizes the least mean squares difference between two signals: the reference incoming signal and the filtered original (outgoing) signal (see Fig. 7).

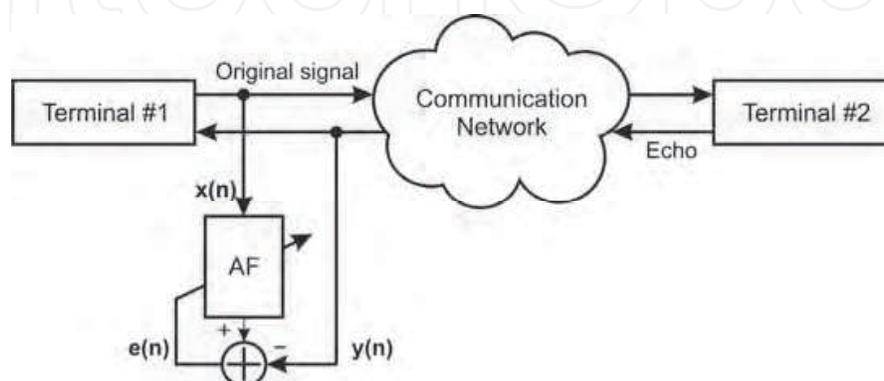


Fig. 7. Time delay estimation using adaptive filter

Basically, the NLMS algorithm is a simple extension of the Widrow’s LMS algorithm. According to Fig. 7, the adaptive filter weights are tuned iteratively using a special rule. The important parameter controlling the whole adaptation process is referred to as a step-size parameter. It should be varied in time in order the algorithm to be able to track non-stationary changes in the echo path’s impulse response. Unlike the LMS algorithm, the NLMS algorithm’s step-size is adjusted according to the instantaneous short-time energy of the input signal. The adaptive filtering process using the NLMS algorithm can be described by the following set of equations

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{1}{\mathbf{x}^T(n) \cdot \mathbf{x}(n)} \mu_0 \cdot e(n) \cdot \mathbf{x}(n) \tag{13}$$

$$\begin{aligned} e(n) &= y(n) - \mathbf{w}^T(n) \cdot \mathbf{x}(n) = y(n) - \sum_{i=0}^{L-1} w_i(n) \cdot x(n-i) \\ \mathbf{w}(n) &= [w_1(n), w_2(n), \dots, w_L(n)]^T \\ \mathbf{x}(n) &= [x(n), x(n-1), \dots, x(n-L+1)]^T \\ \boldsymbol{\mu}(n) &= [\mu_1(n), \mu_2(n), \dots, \mu_L(n)]^T \end{aligned} \tag{14}$$

where \mathbf{w} is a vector containing L coefficients of the adaptive filter; \mathbf{x} is a vector consisting of L samples of the input signal $x(n)$; $e(n)$ is a difference between the reference signal, $y(n)$, and the adaptive filter output during the n^{th} iteration; μ_0 is the fixed NLMS step-size parameter from the interval (0;1). Fig. 8 illustrates the basic principle of the NLMS adaptive algorithm. While the signal $x(n)$ corresponding to the outgoing voice signal (that notation is used in the previous section) is referred to as the far-end signal, the signal $y(n)$ corresponding to the incoming voice signal is referred to as the near-end signal. Basically, the near-end signal $y(n)$ is the delayed and attenuated version of the far-end signal $x(n)$.

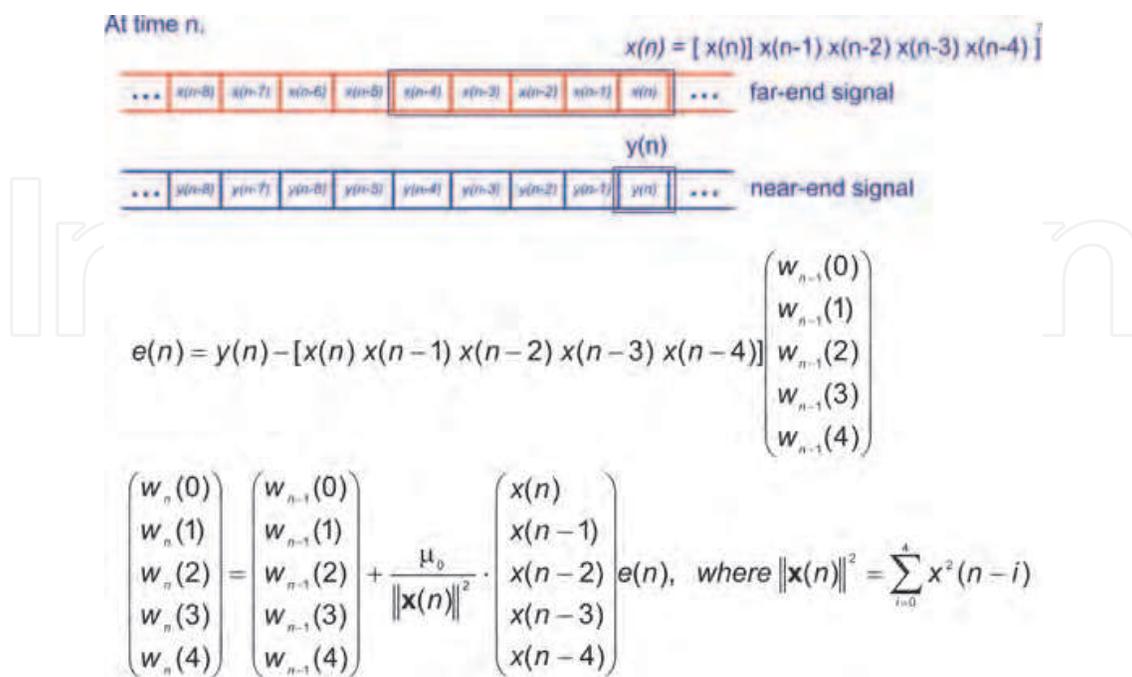


Fig. 8. Principle of the normalized least mean squares algorithm

3.2 Time-domain adaptive algorithms

Knowing the adaptive theory, it is trivial that the delay estimation can be achieved by selecting the largest value from the adaptive filter weights vector, w . There is only one issue that has to be taken into account. The adaptive filter needs some time in order to converge to the optimal performance. The existing adaptive algorithms differ from each other with different convergence properties and computational memory requirements. The robust fast converging algorithms are primarily used in the acoustical echo cancellation applications. They take a lot of computational resources. In our case, it is not necessary to apply the complex algorithms, because the adaptive filter is not directly used for the purpose of echo cancellation, but for the delay estimation. Therefore, the reduced complexity adaptive filtering algorithms became the subject of our interest.

3.2.1 Proportionate adaptive filtering algorithms

In the following, we provide a reader along with a principal of the proportionate adaptive technique. The proportionate normalized least mean squares (PNLMS) algorithm proposed in (Duttweiler, 2000) has been developed for use especially in the telephone network environment. For hybrid echo cancellers, it is reasonable to assume that the echo path has a sparse character (i.e., many IR's (Impulse Response) coefficients are close to zero). Although there are studies and research on the multiple reflection echo paths, a typical echo path impulse response in the practical communication networks has only one reflection, which means all the active coefficients are occupied in a continuous area of the whole echo span. Therefore, we do not need to update all the filter weights to achieve a nominal performance. One can adjust and operate only with non-zero active coefficients. Basically, this is a main idea behind the PNLMS algorithm and other subsequently derived proportionate adaptive filters (Hongyang & Doroslovacki, 2006; Paleologu et al., 2006). Proportionate approaches achieve their higher convergence rate by using the fact that the active part of network echo path is usually much smaller (4-8ms) compared to 64-128 ms of the whole echo path that has to be covered by the adaptive filter. In case of voice transmission over the packet-switching network, these numbers may be more considerable. In the PNLMS algorithm, the adaptive step-size parameters are assigned to all the filter coefficients. They are calculated from the last estimate of the filter weights in such a way that a larger coefficient receives a larger increment. As a result, the convergence rate can be increased the fact that the active taps are adjusted faster than non-active coefficients. Therefore for the sparse IR, the PNLMS algorithm converges much faster comparing to the NLMS. This feature is an advantage especially when it is necessary to estimate the long echo delays. The PNLMS algorithm can be described using the following equations:

$$w(n+1) = w(n) + \frac{1}{x^T(n) \cdot G(n-1) \cdot x(n)} \mu_0 \cdot G(n-1) \cdot e(n) \cdot x(n) \quad (15)$$

$$G(n-1) = \text{diag}\{g_0(n-1), \dots, g_{L-1}(n-1)\} \quad (16)$$

where $G(n-1)$ is a diagonal matrix adjusting the step-size parameters, μ_0 is an overall step-size parameter. The diagonal elements of $G(n)$ are estimated as follows:

$$\gamma_i(n) = \max\{\rho \max[\delta_p, |w_0(n)|, \dots, |w_{L-1}(n)|], |w_i(n)|\} \quad (17)$$

$$g_l(n) = \frac{\gamma_l(n)}{\sum_{i=0}^{L-1} \gamma_i(n)}, \quad 0 \leq l \leq L-1 \quad (18)$$

Parameters δ_p and ρ are positive numbers with typical values for $\delta_p = 0.01$ and for $\rho = 5/L$. The first term in (17), ρ , prevents $w_l(n)$ from stalling when it is much smaller than the largest coefficient and δ_p regularizes the updating when all coefficients have zero values at initialization. In spite of the sparse system identification, which is a vital requirement for the fast converging adaptive filters, there is another requirement. It is directly addressed to the adaptive filter implementation. The algorithm should have reasonable power concerns. Unfortunately, the PNLMS algorithm has drawbacks. One of them is an increase in the computational complexity by 50% compared to the NLMS algorithm. Furthermore, the PNLMS algorithm shows the slow convergence rate after the fast initial start. It is because of the slow convergence rate dedicated to the small coefficients (Gay, 1998). The increased computational complexity can be reduced by the way of selective partial-updating. In turn, the slow convergence of the PNLMS in the stable state can be improved by switching from the PNLMS to NLMS equations after the fast initial convergence has been achieved (Benesty & Gay, 2002). Recently researchers proposed the partial-update techniques for using along with the adaptive algorithms. The partial-update algorithms reduce computational complexity by updating only a subset of the filter weights per iteration. They can be relevant for applications requiring fast real-time processing. Unfortunately, there is another side to the coin. The fewer coefficients you are going to update per iteration, there is more misalignment presented in the algorithm. Therefore, a certain trade-off should be made when selecting the number of coefficients to update. The following subsection presents several partial-update algorithms along with the proposed modification of the partial-update PNLMS algorithm.

3.2.2 Partial-update adaptive filtering algorithms

The partial-update algorithms can be seen to exploit the sparseness of the echo path in two different ways. It is known that when the unknown system's impulse response is sparse, many of the adaptive filter's weights can be approximated to zero. Alternatively, the sparseness may be present in the weight update vector as a consequence of the distribution of the input samples in the $(L \times 1)$ input vector, $x_n = [x(n), x(n-1), \dots, x(n-L+1)]^T$. In both these cases, exploiting the sparseness properties can reduce complexity and improve performance of the adaptive algorithm (Fevrier & Gelfand, 1999). Some of the first work on the partial-update algorithms was done by Douglas (Douglas, 1997). It presents the periodic and the sequential updating schemes for the Max-NLMS algorithm. However, these partial-update algorithms show slow convergence 2properties compared to the full-update algorithms. The reason is inconsistent updating schemes. More recently, the partial-updating concept was developed by Aboulnasr (Aboulnasr, 1999). It leads to the M-Max NLMS algorithm and supporting convergence analysis (Aboulnasr & Mayyas, 1998). Another block-updating scheme for the NLMS algorithm was studied by Schertler (Schertler, 1998). The latter work was published by Dogancay and Tanrikulu. They consider approaches for more robust Affine Projection Algorithm (APA) (Dogancay & Tarinkulu, 2001, 2002). Further, we give a summary of the general partial-update algorithms along with the proposed one.

M-Max NLMS

The algorithm selects a specified number of the coefficients providing the largest reduction in the mean squared error per iteration (Naylor & Sherliker, 2003). Only M out of the total L filter coefficients are updated. Those M coefficients are the ones associated with the M largest values within the following vector $|x(n - i + 1)|; i = 1; \dots; L$. The update equations for this algorithm are

$$w_i(n+1) = w_i(n) + \frac{1}{x^T(n) \cdot x(n)} \mu_0 \cdot e(n) \cdot x_i(n) \quad (19)$$

$$w_i(n) = \begin{cases} 1, & i \in \{M \text{ maxima of } |x(n - i + 1)|\} \\ 0, & \text{otherwise} \end{cases}, \quad 0 \leq i \leq L \quad (20)$$

One of the features of the M-MAX-NLMS algorithm is that it reduces the complexity of the adaptive filter by selectively updating the coefficients while maintaining the closest performance to the full-update NLMS algorithm. We present misalignment curves for the algorithm in the up-following section. The misalignment value was calculated using the equation formula

$$\text{Misalignment}(n) = 10 \cdot \log_{10} \left(\frac{\sum_{j=1}^L (h_j(n) - w_j(n))^2}{\sum_{j=1}^L (h_j(n))^2} \right) \quad (21)$$

Here, w_j is the j -th coefficient of the adaptive filter, while h_j is the j -th coefficient of the simulated echo path's IR.

Selective-partial-update NLMS

This algorithm opposed to the M-Max NLMS has a block structure. An objective behind the latter is the same: it reduces computational costs by updating a subset of the filter coefficients. But first, the vector $x(n)$ and the coefficient vector $w(n)$ are arranged into K blocks of length $M = L/K$, where L is an integer as in

$$x(n) = [x_1^T(n) \quad x_2^T(n) \quad \dots \quad x_k^T(n)]^T \quad (22)$$

$$w(n) = [w_1^T(n) \quad w_2^T(n) \quad \dots \quad w_k^T(n)]^T \quad (23)$$

The coefficient vector's blocks $w_1(n), w_2(n), \dots, w_k(n)$ represent candidate subsets that can be updated during the current iteration. For a single-block updating scheme, the constrained minimization problem, which is solved by the NLMS algorithm, can be written as

$$w_i(n+1) = w_i(n) + \frac{1}{\|x_i(n)\|^2} \mu_0 \cdot e(n) \cdot x_i(n) \quad (24)$$

The selection of the block that has to be updated is made by determining the block with the smallest squared-Euclidian-norm update. According to Eq. (24), that justification can be described by the following terms

$$i = \arg_{\min_{1 \leq j \leq M}} 1/\|x_j(n)\|^2 = \arg_{\max_{1 \leq j \leq M}} \|x_j(n)\|^2 \quad (25)$$

Generalization from the single-block to the multiple-block updating scheme is done through the following. Suppose that only the B ($B < K$) blocks with the largest magnitudes are selected to be updated. Let a vector $I_B = \{i_1, i_2, \dots, i_B\}$ denote the subset of B blocks out of $\{1, 2, \dots, K\}$ to be updated. Thus, the equation for the B -block updating scheme is

$$w_{I_B}(n+1) = w_{I_B}(n) + \frac{1}{\|x_{I_B}(n)\|^2} \mu_0 \cdot e(n) \cdot x_{I_B}(n) \quad (26)$$

where x_{I_B} and w_{I_B} are defined as follows

$$x_{I_B}(n) = [x_{i_1}^T(n) \ x_{i_2}^T(n) \ \dots \ x_{i_B}^T(n)]^T \quad (27)$$

$$w_{I_B}(n) = [w_{i_1}^T(n) \ w_{i_2}^T(n) \ \dots \ w_{i_B}^T(n)]^T \quad (28)$$

The computational and memory requirements of the selective-partial-update NLMS algorithm are almost identical to those of the selective-block-update algorithm proposed in (Aboulnasr & Mayyas, 1998). Nevertheless, simulation results illustrated in the next section shows that this approach does not lead to the reasonable trade-off between performance and simplicity. The algorithm's efficiency is weaker than the one of the M-Max NLMS algorithm. As an alternative approach, a sparse-partial-update NLMS algorithm applies more relevant selection criterion.

Sparse-partial-update NLMS

This algorithm utilize a so-called sparse-partial (SP) weight selection criterion (Jinhong & Doroslovacki, 2008). The adaptive filter weights are updated based on the largest product of the multiplication of $x(n)$ and $w(n)$. The SP-NLMS single-block update equations are given by

$$w_i(n+1) = w_i(n) + \frac{1}{x^T(n) \cdot x(n)} \mu_0 \cdot e(n) \cdot x_i(n) \quad (29)$$

$$w_i(n) = \begin{cases} 1, & i \in \{M \text{ maxima of } |x(n-i+1)| \cdot w_i(n)\}, \\ 0, & \text{otherwise} \end{cases}, \quad 0 \leq i \leq L \quad (30)$$

Hongyang and Dyba recently suggested a generalization for updating B blocks out of K , i.e.

$$w_{I_B}(n+1) = w_{I_B}(n) + \frac{1}{x_{I_B}^T(n) \cdot x_{I_B}(n)} \mu_0 \cdot e(n) \cdot x_{I_B}(n) \quad (31)$$

$$I_B = \{ i : \text{is one of the } B \text{ largest among } w_{i_1}^T(n) \cdot x_{i_1}(n), \dots, w_{i_B}^T(n) \cdot x_{i_B}(n) \} \quad (32)$$

Simple-partial-update PNLMS

This proposed algorithm exploits the sparseness of the communication channel to speed up the initial convergence and employs the partially updating scheme to reduce the

computational complexity. A selection procedure is performed in accordance with the estimated magnitude of the channel's impulse response. The S-PNLMS algorithm for single-block update is defined as follows. Arrange $x(n)$ and $w(n)$ into K blocks of length $M = L/K$ in the same way as it is done in Eq. (22, 23). Then let $G_i(n)$ denote the corresponding $M \times M$ block of the diagonal weighing matrix, $G(n)$. The recursion for updating adaptive filter weights is given by

$$w_i(n+1) = w_i(n) + \frac{1}{x_i^T(n) \cdot G_i(n-1) \cdot x_i(n)} \mu_0 \cdot G_i(n-1) \cdot e(n) \cdot x_i(n) \quad (33)$$

where the block selection is done according to the following

$$i = \arg \min_{1 \leq j \leq M} G_j(n) \quad (34)$$

It is different to

$$i = \arg \min_{1 \leq j \leq M} x_j \cdot G_j(n) \cdot x_j \quad (35)$$

which is used with the SPU-PNLMS algorithm. It is apparently from the simulations that the S-PNLMS has similar performance to the SP-NLMS and outperforms the SPU-PNLMS algorithm. The S-PNLMS for updating B blocks out of M has these update equations

$$w_{i_B}(n+1) = w_{i_B}(n) + \frac{1}{x_{i_B}^T(n) \cdot G_{i_B}(n-1) \cdot x_{i_B}(n)} \mu_0 \cdot G_{i_B}(n-1) \cdot e(n) \cdot x_{i_B}(n) \quad (36)$$

$$I_B = \{ i : G_i(n) \text{ is one of the } B \text{ largest among } G_1(n), \dots, G_M(n) \} \quad (37)$$

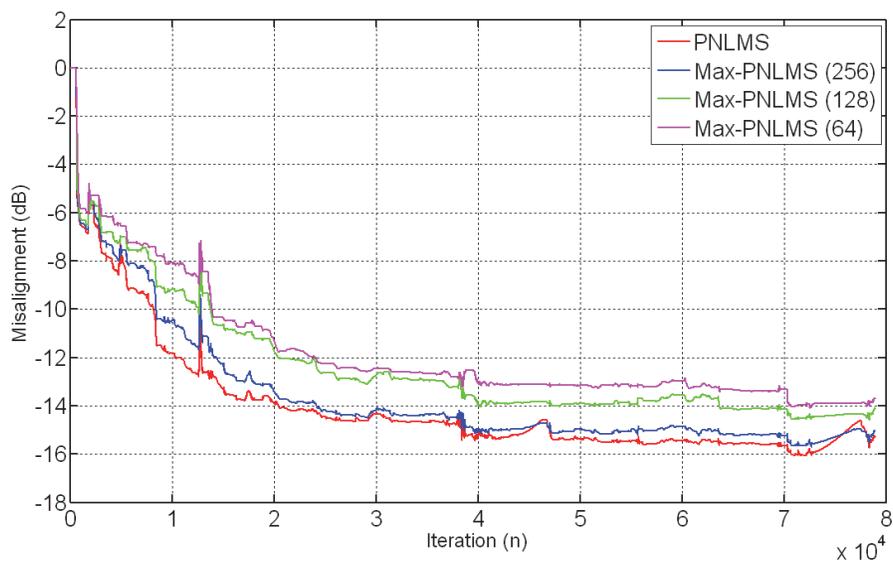
Further, we provide the comparison results for the presented algorithms and demonstrate their performance while estimating the predefined echo delay. Table 5 illustrates the computational complexity of the full-update algorithms and shows saving achieved by the partially updating schemes.

ALG.	MULT.	ADD.	DIV.
NLMS	$3L+1$	$3L-1$	1
M-Max-NLMS	$3M+1$	$3M-1$	1
SPU-NLMS	$3M^*B+1$	$3M^*B-1$	1
SP-NLMS	$3M^*B+1$	$3M^*B-1$	1
PNLMS	$6L+1$	$4L-2$	$L+1$
M-Max-PNLMS	$6M+1$	$4M-2$	$M+1$
SPU-PNLMS	$6M^*B+1$	$4M^*B-2$	M^*B+1
SP-PNLMS	$6M^*B+1$	$4M^*B-2$	M^*B+1
S-PNLMS	$6M^*B+1$	$4M^*B-2$	M^*B+1

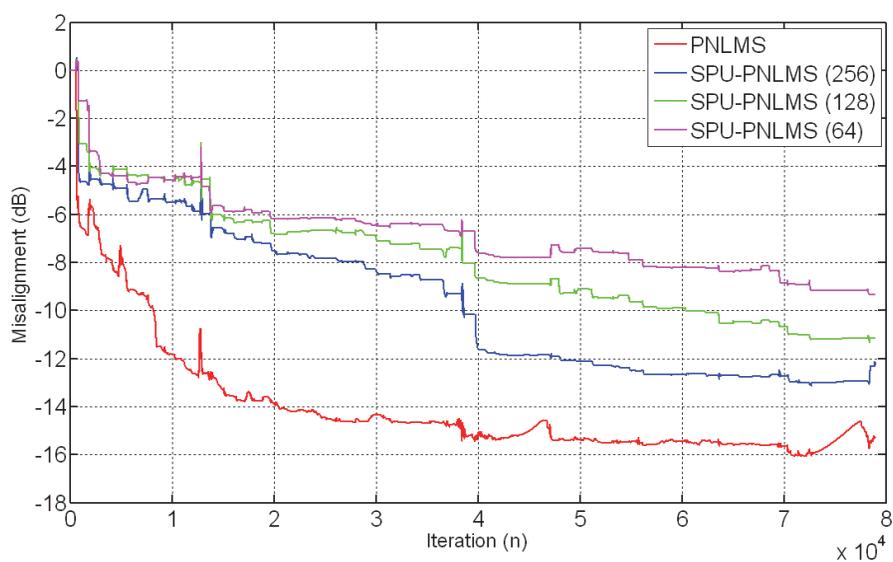
Table 5. Comparison of computational complexity

3.3 Discussion over experimental results

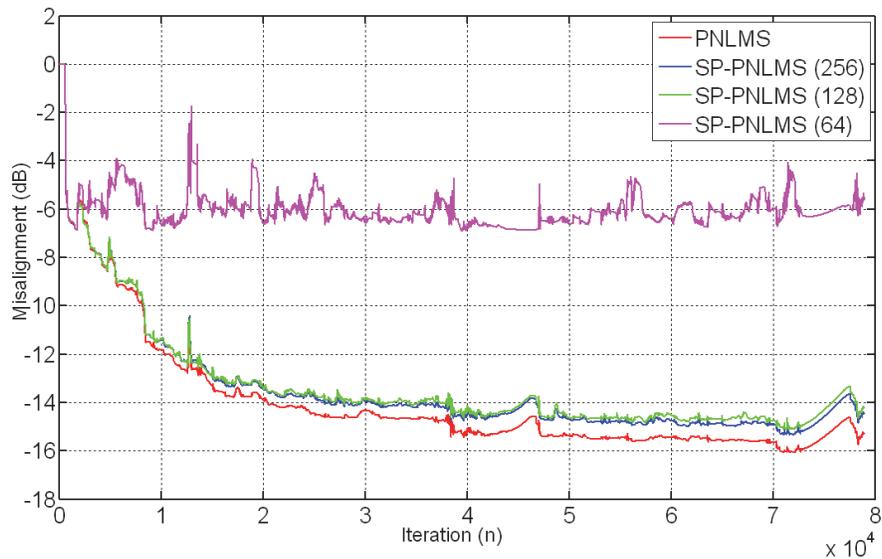
To evaluate the performance of the adaptive filtering algorithms, we implemented them in MATLAB environment. The working scenario is as follows. The filters have to estimate the predefined echo path's impulse responses specified in the ITU-T Recommendation G.168 (ITU-T G.168, 2002). They consist of 1024 taps. The overall step-size parameter, μ_0 , is chosen to be 0.1. The control parameters ρ and δ_p are chosen to be 0.001 and 0.01 respectively. In the first part of the experiment, we look at the misalignment curves of the M-Max-, SPU-, SP- and S-PNLMS algorithms. They are illustrated in Fig. 9 below. The SPU-updating scheme produces the worst results. The proposed S-criterion considerably outperforms it especially in terms of the initial convergence speed. The rest of the algorithms have nearly the same convergence and tracking performance.



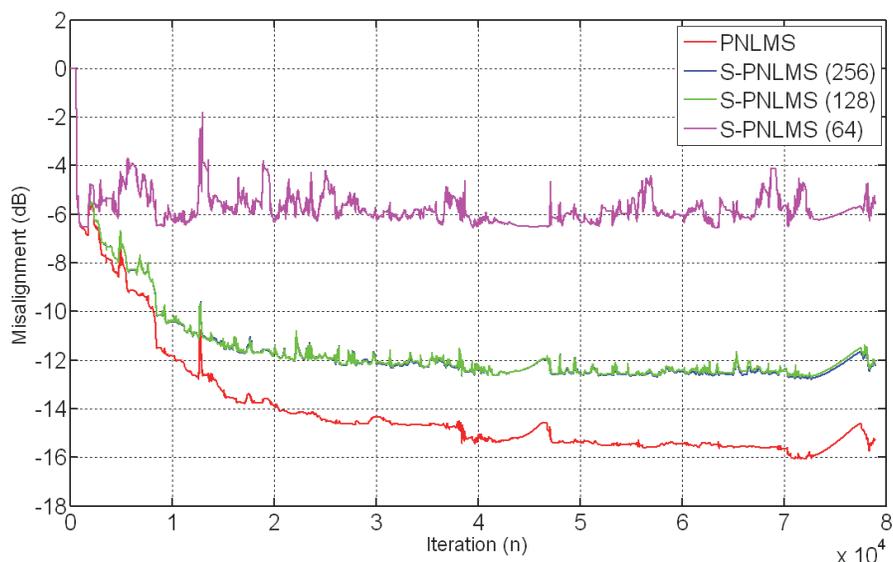
(a) Max-PNLMS algorithm



(b) SPU-PNLMS algorithm



(c) SP-PNLMS algorithm

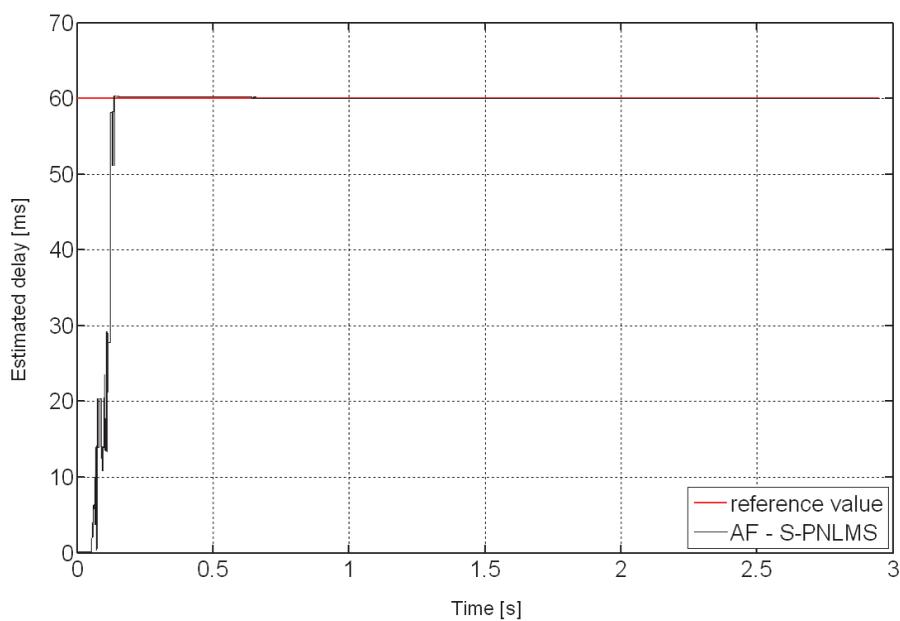


(d) S-PNLMS algorithm

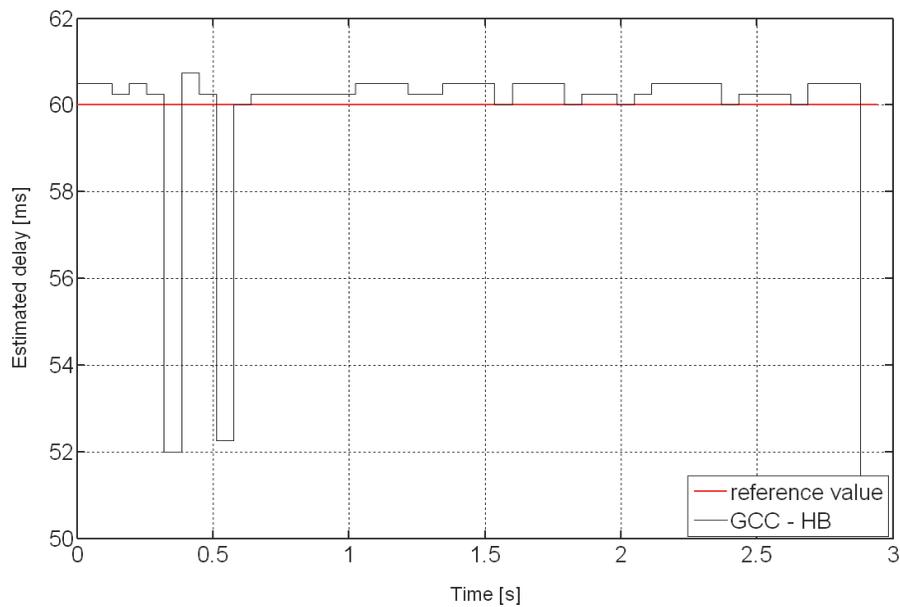
Fig. 9. Misalignment curves of the partial-update algorithms

All the algorithms, except the M-Max-PNLMS, show poor results when the M value equals 64. It can be explained by the fact that the active part of the IR is approximately 16ms long. This value corresponds to 128 samples for sampling frequency of 8kHz, therefore, 64 samples are not enough to cover the active region completely. Regarding to the dissimilar selection criterion, the M-Max-PNLMS algorithm can deal relatively well with that problem. The Max-updating formula does not count with the sparse character of the IR. It performs selection according to the distribution of the values of the input vector. Otherwise, its drawback is lower initial convergence speed comparing to the SP-PNLMS algorithm. The second part of our experiment concerns the performance of the adaptive algorithms versus the ones based on the generalized cross-correlation function. They are compared in the context of the time delay estimation. The time difference between time when the outgoing

signal leaves the voice terminal and the moment when the incoming signal containing the echo of the original signal arrives back from the network is referred to as a true echo delay. Table 6 concludes the results for the AF and GCC algorithms for different delay values. Fig. 10 shows the estimated echo delay diagram for the echo path that has a 60 ms pure delay. It can be observed that the outputs from the adaptive filter in the steady state have much smaller variance than the results obtained from the algorithm based on the generalized cross-correlation function.



(a) TDE using S-PNLMS algorithm



(b) TDE using S-PNLMS algorithm

Fig. 10. Estimated echo delay curves

[ms]	ROTH	SCOT	CPS-2	HB	M-Max	SPU	SP	S	samples
5	5,1	5,2	5,2	5,2	5,1	5,1	5,0	5,0	40
10	10,3	10,3	10,3	10,4	10,3	10,3	10,4	10,1	80
20	19,6	19,7	20,0	20,0	20,5	20,5	20,1	20,3	160
30	29,3	29,6	30,1	30,0	30,8	30,8	30,1	30,2	240
50	49,0	49,7	49,4	49,8	51,3	51,3	50,3	50,1	400
100	98,1	99,4	98,8	99,6	102,5	102,5	100,5	100,6	800
200	196,2	198,8	197,6	199,1	205,1	205,1	201,3	201,2	1600
300	294,2	298,1	296,4	298,7	307,6	307,6	301,8	301,9	2400

Table 6. Mean values of estimated echo delays

4. Partial, proportionate and sparse control for multi-delay filters

Taking into the account the fact that the generalized cross-correlation algorithms operate in the frequency domain and use advantages of the fast Fourier transform, reasonable computational savings for adaptive filtering algorithms can be achieved as well. The basic operation underlying frequency domain adaptive filters is the transformation of the input signal into a more desirable wave form before its adaptive processing. This is accomplished by the Discrete Fourier Transform (DFT) whereby the input signal is transformed to the frequency domain as shown in Fig. 11.

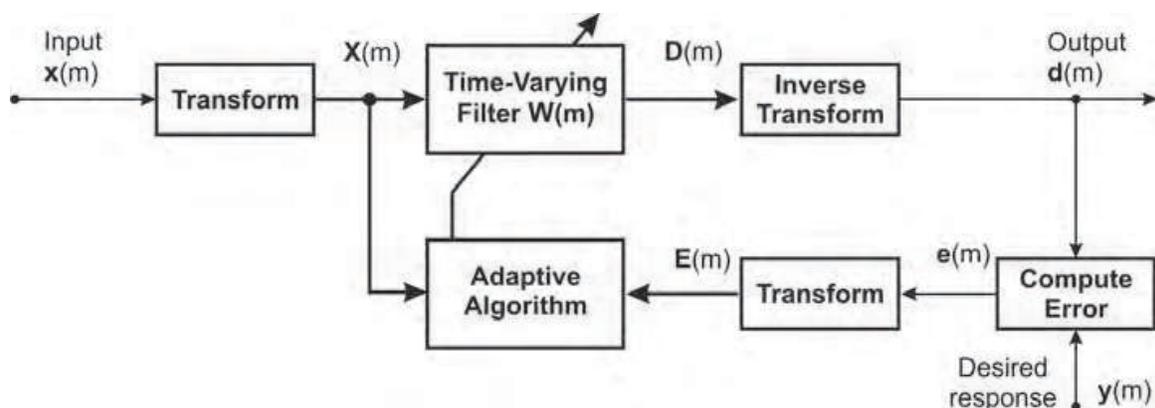


Fig. 11. Frequency domain adaptive filter configuration

Frequency Domain (FD) adaptive filters have primarily two advantages compared to time domain implementations (Dohnal, 1995). The first advantage is the potentially large savings in the computational complexity. The Fast Fourier Transform (FFT) is an efficient implementation of the DFT, which provides these savings. A second advantage is that the DFT structures generate signals that are approximately uncorrelated (orthogonal) (Shynk, 1992). As a result, a time-varying step-size parameter is used for each adaptive weight, thereby allowing a more uniform convergence rate across the adaptive filter. Thus, the overall convergence rate of the algorithm may be improved, sometimes approaching that achievable with Recursive Least Squares (RLS) algorithms without a similar increase in the computational complexity (Widrow & Stearns, 1985). Many other attractive features and variation of the FD adaptive filters can be found in the literature (Khong et al., 2007).

4.1 Sparse partial-update proportionate multi-delay filter

The multi-delay frequency domain adaptation algorithm generates its individual step-size control factors that are mutually orthogonal. This advantage put the Multi-Delay block Frequency domain (MDF) algorithm closer to the analytical RLS algorithm. It differs from the algorithms processing in the time domain with better convergence properties and lower computational complexity. The MDF algorithm decreases the processing delay associated with frequency domain adaptive filters. The delay value is directly proportional to the number of the filter coefficients. The multi-delay approach solves this problem by partitioning the time-domain filter of length L into K sub-filters each of length N with $L=KN$. Consequently, the delay value is reduced by a factor of L/N compared to the full length approach. The diagram of the MDF algorithm is shown in Fig. 12. This algorithm uses the FFT of size N_{FFT} . It equals to the smallest power of two integers larger than or equal to $2L/K=2KN/K=2N$. The MDF algorithm works as follows. Let us first define m , as the frame index. The speech signals are operated on the frame-by-frame basis.

The first step of the MDF algorithm is to convert the most recent overlapped input samples to the frequency domain as follows

$$X_f(K, m) = \text{diag} \left\{ \text{FFT} \left[x_0(m-1), x_1(m-1), \dots, x_{N-1}(m-1), x_0(m), x_1(m), \dots, x_{N-1}(m) \right]^T \right\} \quad (38)$$

The input vectors incoming to each of the sub-filter blocks can be obtained via the frame index shifting without invoking any computation as follows:

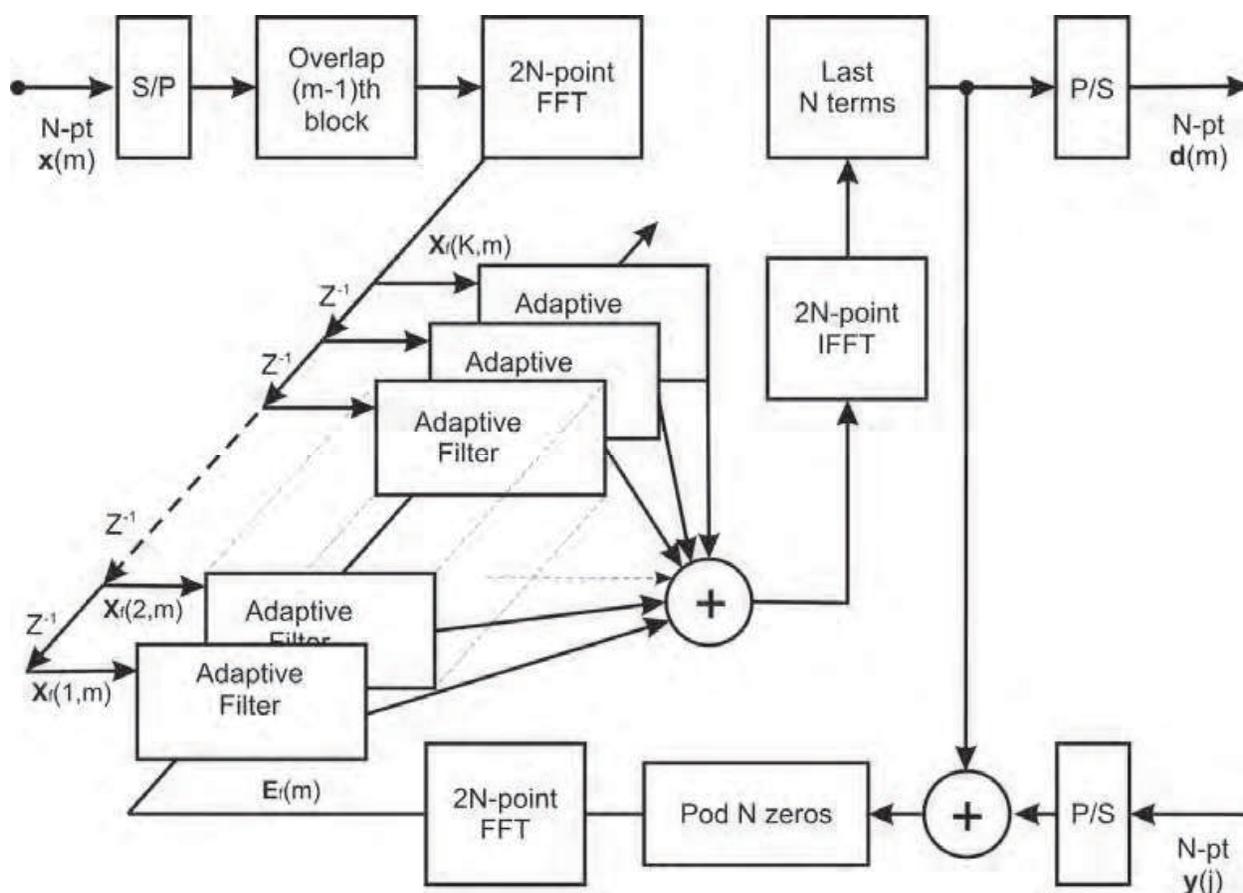


Fig. 12. Block diagram of the MDF algorithm

$$\mathbf{X}_f(k, m) = \mathbf{X}_f(k+1, m-1), \quad k = 1, \dots, K-1 \quad (39)$$

The described approach suggests that only one FFT is needed per frame iteration in order to transform the input vector into the frequency domain. It implies a significant computation saving. The output and the error frequency-domain vectors can be expressed and calculated as

$$\mathbf{d}(m) = \text{last } N \text{ terms of } \left\{ FFT^{-1} \left[\sum_{k=1}^K \mathbf{X}_f(k, m) \cdot \mathbf{W}_f(k, m) \right] \right\} \quad (40)$$

$$\mathbf{E}_f(m) = FFT \left\{ \underbrace{0, 0, \dots, 0}_{N \text{ zeros}}, \underbrace{\mathbf{y}(m) - \mathbf{d}(m)}_{N \text{ terms}} \right\} \quad (41)$$

where $\mathbf{W}_f(k, m)$ is the k^{th} coefficient vector and $\mathbf{d}(m)$ is the desired vector. The coefficient update equations to minimize the Mean Square Error (MSE) are the following

$$\boldsymbol{\varphi}_i(k, m) = \text{first half of } \left\{ FFT^{-1} \left[\mathbf{X}_f(k, m)^* \cdot [S_{MDF}(m) + \delta]^{-1} \cdot \mathbf{E}_f(m) \right] \right\} \quad (42)$$

$$S_{MDF}(m) = \lambda \cdot S_{MDF}(m-1) + (1-\lambda) \cdot \mathbf{X}_f(k, m)^* \cdot \mathbf{X}_f(k, m) \quad (43)$$

$$\boldsymbol{\Phi}_f(k, m) = FFT \left[\boldsymbol{\varphi}_i(k, m), \underbrace{0, 0, \dots, 0}_{N \text{ zeros}} \right]^T \quad (44)$$

$$\mathbf{W}_f(k, m+1) = \mathbf{W}_f(k, m) + K \cdot \mu_k \cdot \boldsymbol{\Phi}_f(k, m) \quad (45)$$

where $k=1,2,\dots, K$ and μ_k is the block step-size control parameter, $0 < \lambda < 1$ is the forgetting factor. Further, we examine how the partial update technique may be incorporated into the MDF algorithm. For frequency-domain selection, the known approach is to select frequency bins corresponding to the largest magnitude Fourier transform of the tap-input over all the sub-filter blocks $k=1,2,\dots, K$ (Khong et al,2008). However, the selection is better to perform on the coefficient-block basis (Deng & Dyba, 2008). It is faster and takes less computation power. Therefore, first we need to calculate metrics for each of the sub-filter in order to perform selection procedure. The number of metrics equals the number of sub-filter blocks K . One particular frequency-domain metric, Q_k , may, for example, represent the sum of the frequency bins in the corresponding sub-filter block

$$Q_f(k, m) = \sum_{j=1}^{2N} |X_{f_j}(k, m)| \quad (46)$$

There is a little bit different selection approach, which shows better convergence performance compared to the last one. Define the $2L \times 1$ vector $\boldsymbol{\chi}(m)$, which consists of the concatenated Fourier transform of the input signal across all sub-filters, as

$$\boldsymbol{\chi}(m) = [\mathbf{X}_f^T(0, m), \dots, \mathbf{X}_f^T(K-1, m)] = [\chi_1(m), \dots, \chi_{2L}(m)]^T \quad (47)$$

Each element in the $2L \times 2L$ diagonal selection matrix $\mathbf{P}(m)$

$$p_l(m) = \begin{cases} 1, & \text{if } l \text{ corresponds to } M_f \text{ maxima of } |\chi_l(m)|, l = 0, \dots, 2L-1 \\ 0, & \text{otherwise} \end{cases} \quad (48)$$

The tap-selection routine is done by multiplying this matrix with the corresponding input in every k^{th} sub-filter, as it is shown below

$$\tilde{\mathbf{X}}_f(k, m) = P_k(m) \cdot \mathbf{X}_f(k, m) \quad (49)$$

Otherwise, if it is supposed that the algorithm models a sparse network impulse response, the sparse partial updating scheme should be involved. The control metrics are calculated as follows ($k=1, 2, \dots, K$)

$$\mathbf{Q}_f(k, m) = \mathbf{X}_f(k, m)^* \cdot \mathbf{W}_f(k, m) \quad (50)$$

Here, we suggest using a new type of metric, in order to choose sub-filter blocks for updating. Note, that it should be calculated in the time-domain as follows

$$\mathbf{Q}_i(k, m) = \frac{\sum_{i=0}^{N-1} |w_i(k, m)|}{\sqrt{L} \cdot \sqrt{\sum_{j=0}^{L-1} (w_j(m))^2} + \varepsilon} \quad (51)$$

We have to remember that the minimum number, M , of the sub-filter blocks to be updated must equal or be greater than two ($2 \leq M < K$). This is because the pure delay of the echo path is never known. Thus, if only one sub-filter is chosen, it can not cover all the possibilities of the variation of echo path. Another important question for frequency domain adaptive filters is the FFT length? – From our point of view, it is reasonable using the FFT of 256 samples long. This is an optimal length, which can provide satisfactory results for the high range of MDF algorithms. We have tried to apply different values, but they did not bring sufficient improvements. For example, if the echo path consists of 1024 taps, the adaptive filter should be divided in $1024 / (256 / 2)$ sub-filters. The number K of sub-filters will equal 8.

Before we present a proposed frequency domain algorithm, let us resume the following. The partial update frequency-domain algorithms may use the values of the calculated block metrics as selection criteria. Each type of metrics has its own features. In case of (13, 18, 19), only K ($K \ll 2L$) metrics are sorted in order to select M ($2 \leq M < K$) active subfilters. Since K is a small number compared to the entire number $2L$ of subfilters coefficients in (14, 15, 16, 17), the sorting overhead may be reduced by factor $2L/N$. We have spoken about the features of sparse echo paths and the possibility of improving their convergence rate by assigning the individual step-size parameters proportionally to the coefficient magnitudes. In the art, there is a big family of various MDF algorithms. Researchers are trying to improve algorithms performance from one hand and to reach computational savings from the other hand. Further we present our approach how to deal with this contradictory problem. First, we describe the basic idea of sparseness measure, and show how to incorporate it into the MDF framework. The concept of sparseness refers to a representational scheme where only a few units (out of a large massive) are effectively used to represent typical data vectors. In effect, this implies most units taking values close

to zero while only few take significantly non-zero values. Numerous sparseness measures have been proposed and used in the literature to date (Benesty et al, 2006). Such measures quantify how much energy of a vector is packed into only a few components. On a normalized scale, the sparsest possible vector (only a single component is non-zero) should have a sparseness of one, whereas a vector with all elements equal should have a sparseness of zero. In our approach, we use a sparseness measure based on the relationship between the L_1 norm and the L_2 norm:

$$\xi(m) = 1 - \frac{\|w_j(m)\|_1}{\sqrt{L} \cdot \|w_j(m)\|_2 + \varepsilon} = 1 - \frac{\sum_{j=0}^{L-1} |w_j(m)|}{\sqrt{L} \cdot \sqrt{\sum_{j=0}^{L-1} (w_j(m))^2} + \varepsilon} \quad (52)$$

Here, L is the length of the adaptive filter in the time domain. This function evaluates to unity if and only if w contains only a single non-zero component, and takes a value of zero if and only if all components are equal (up to signs), interpolating smoothly between the two extremes. We use this sparseness measure ξ of the time domain estimated filter coefficients to adaptively control the number of taps that need to be updated

$$M_t = L \cdot (1 - \xi(m)) \quad (53)$$

Note that the total number of corresponding frequency domain coefficients is $2L$ for an L -long time domain adaptive filter, therefore

$$M_f = 2 \cdot M_t = 2 \cdot L \cdot (1 - \xi(m)) \quad (54)$$

If selection is performed on the block-based basis, then the corresponding number of sub-filters in frequency domain equals

$$M = \text{round} \left\{ \frac{2 \cdot L \cdot (1 - \xi(m))}{2 \cdot N} \right\} = \text{round} \{ K \cdot (1 - \xi(m)) \} \quad (55)$$

Fig. 13 shows sparseness measure for sparse and dispersive impulse responses. It is also shown the number of adaptive filter coefficients that should be updated, if partial updating scheme is considered.

The convergence process, by itself, can be divided into two stages, i.e. before and after the convergence of the impulse response (or some time, before and after the convergence of the considerable/large coefficients). It has been showed in the time domain adaptive filtering that it is better using PNLMS/IPNLMS at the initial stage for fast convergence, and afterwards switching to NLMS for assuring better misalignment. Here, for recognizing between two stages in adaptation process, we propose to use the following parameter

$$\eta(m) = 10 \cdot \log_{10} \left(\frac{\sum_{j=0}^{L-1} (w_j(m))^2}{L \cdot 0,001} \right) \quad (56)$$

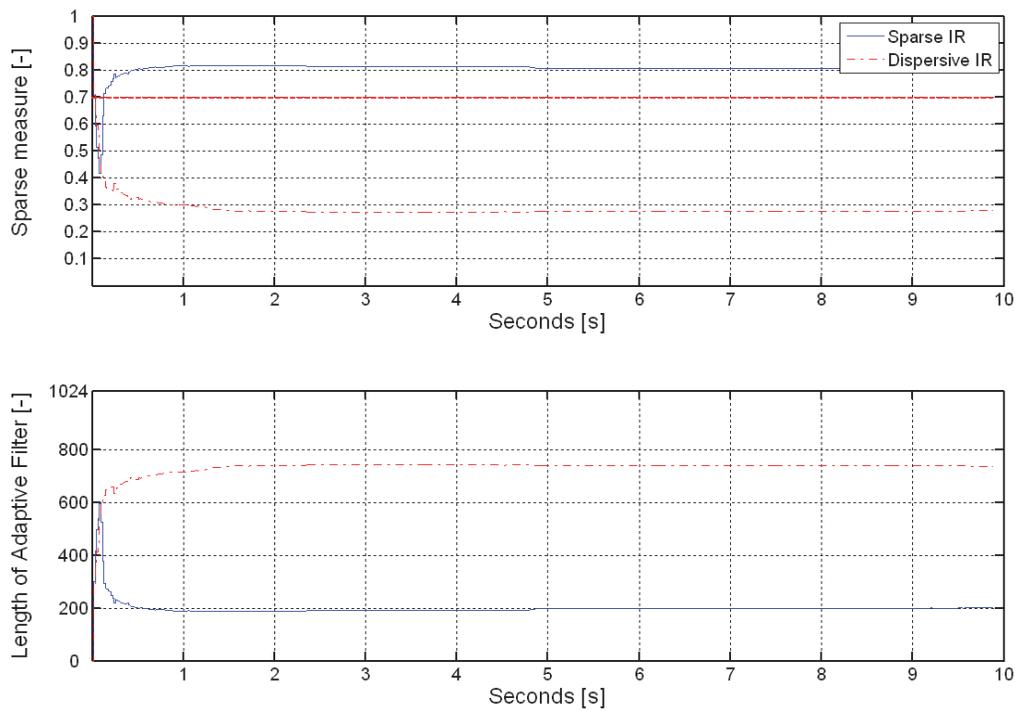


Fig. 13. Sparseness measure and its impact on length of adaptive filter

The quantity in Eq. (56) represents an energy measure in [dBm] within an estimated impulse response. Fig. 14 demonstrates the curve for this parameter for two frequency domain algorithms. It can be observed that after 0.5 seconds estimation of IR energy measure for IPMDF stops fluctuating. Consequently, this fact can be used for switching between different adaptation schemes.

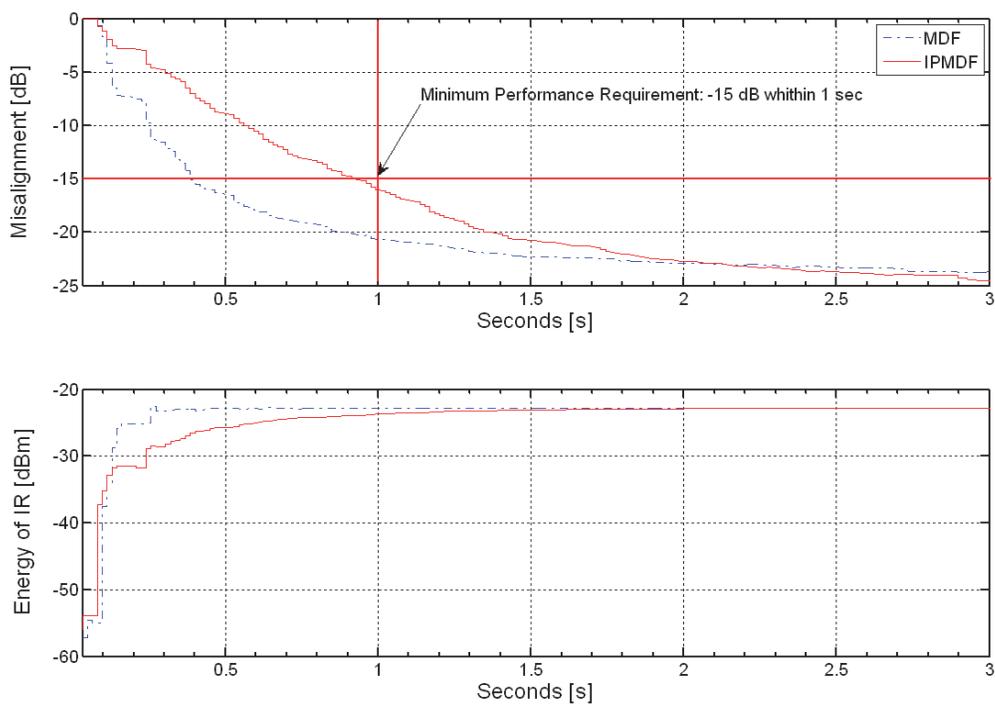


Fig. 14. Estimated misalignment and energy measure for MDF and IPMDF algorithms

Before representing the proposed MDF scheme, declare the following statements:

I. Utilization of sparseness measure, $0 < \xi(m) < 1$ (for real IR)

- a. $\xi(m) < 0.7$
 - i. IR is considered dispersive
 - ii. $M_t = L$, if fully updated scheme is chosen (mostly during initial period)
 - iii. $L/K \leq M_t < L$, if partial updated scheme is chosen
- b. $\xi(m) \geq 0.7$
 - i. IR is considered sparse
 - ii. $M_t = L \cdot (1 - \xi(m))$

II. Utilized updating schemas

- a. non-partial
- b. χ - based selection (coefficient-based), $2L$ -vector
- c. μ - based selection (block-based), K -vector

III. Utilization of IR energy measure, η

- a. $|\eta(m) - \eta(m-1)| \leq \Delta_\eta$, switching to the MDF algorithm
- b. $|\eta(m) - \eta(m-1)| > \Delta_\eta$, switching to the IPMDF (SC-IPMDF) algorithm

Finally, our proposed algorithm can be described as:

$$\begin{aligned}
 & \text{1st stage: } |\eta(m) - \eta(m-1)| \leq \Delta_\eta \\
 & \text{if } \xi(m) < 0.7, \\
 & \quad M_f = 2 \cdot L \quad \text{or} \quad M = K \\
 & \quad \mathbf{w}(k, m+1) = \mathbf{w}(k, m) + L \cdot \mu_k \cdot \mathbf{G}_t(k, m) \cdot \boldsymbol{\varphi}_t(k, m) \\
 & \quad \mathbf{G}_t(k, m) = \text{diag}\{g_{kN}(m), g_{kN+1}(m), \dots, g_{kN+N-1}(m)\} \\
 & \quad g_{kN+l} = \frac{1-\alpha}{2 \cdot L} + \frac{(1+\alpha) \cdot |w_{kN+l}(m)|}{2 \cdot \sum_{j=0}^{L-1} |w_j(m)| + \varepsilon} \\
 & \quad \mu_k = \text{const} \\
 & \quad \text{if } m \cdot N \leq L, \\
 & \quad \quad \alpha = \text{const} \\
 & \quad \text{else } m \cdot N > L, \\
 & \quad \alpha = 1 - 2 \cdot \frac{L}{L - \sqrt{L}} \cdot \left(1 - \frac{\|w_j(m)\|_1}{\sqrt{L} \cdot \|w_j(m)\|_2 + \varepsilon} \right) \\
 & \quad \text{else } \xi(m) \geq 0.7, \\
 & \quad M_f = 2 \cdot L \cdot (1 - \xi(m)) \quad \text{or} \quad M = \text{round}\{K \cdot (1 - \xi(m))\} \\
 & \quad \text{if } \text{mod}(m, T) = 0, \\
 & \quad p_l(m) = \begin{cases} 1, & \text{if } l \text{ corresponds to } M_f \text{ maxima of } |\chi_l(m)|, l = 0, \dots, 2L-1 \\ 0, & \text{otherwise} \end{cases} \\
 & \quad \mathbf{X}_f(k, m) = P_k(m) \cdot \mathbf{X}_f(k, m) \\
 & \quad \boldsymbol{\varphi}_t(k, m) = \text{first half of } \left\{ FFT^{-1} \left[\mathbf{X}_f(k, m)^* \cdot [S_{MDF}(m) + \delta]^{-1} \cdot \mathbf{E}_f(m) \right] \right\} \\
 & \quad \mathbf{w}(k, m+1) = \mathbf{w}(k, m) + L \cdot \mu_k \cdot \mathbf{G}_t(k, m) \cdot \boldsymbol{\varphi}_t(k, m) \\
 & \quad \mu_k = \text{const}
 \end{aligned}$$

$$\begin{aligned}
& \text{else } \text{mod}(m, T) \neq 0, \\
\mu_k(m) &= \begin{cases} \frac{\sum_{i=0}^{N-1} |w_i(k, m)|}{\sqrt{L} \cdot \sqrt{\sum_{j=0}^{L-1} (w_j(m))^2 + \varepsilon}}, & \text{if } k \text{ corresponds to } M \text{ maxima of } \mu_k(m), k = 0, \dots, K-1 \\ 0, & \text{otherwise} \end{cases} \\
& \mathbf{w}(k, m+1) = \mathbf{w}(k, m) + L \cdot \mu_k \cdot \mathbf{G}_i(k, m) \cdot \boldsymbol{\varphi}_i(k, m) \\
& \text{2nd stage: } |\eta(m) - \eta(m-1)| < \Delta_\eta \ \& \ \text{Counter}_\eta > T_\eta \\
& \alpha = -1 \\
& \text{if } \xi(m) < 0.7, \\
& \quad M_f = 2 \cdot L \quad \text{or} \quad M = K \\
& \text{else } \xi(m) \geq 0.7, \\
& \quad M_f = 2 \cdot L \cdot (1 - \xi(m)) \quad \text{or} \quad M = \text{round}\{K \cdot (1 - \xi(m))\} \\
& \quad \text{if } \text{mod}(m, T) = 0, \\
p_l(m) &= \begin{cases} 1, & \text{if } l \text{ corresponds to } M_f \text{ maxima of } |\chi_l(m)|, l = 0, \dots, 2L-1 \\ 0, & \text{otherwise} \end{cases} \\
& \mathbf{X}_f(k, m) = P_k(m) \cdot \mathbf{X}_f(k, m) \\
& \boldsymbol{\varphi}_i(k, m) = \text{first half of } \left\{ \text{FFT}^{-1} \left[\mathbf{X}_f(k, m)^* \cdot [S_{MDF}(m) + \delta]^{-1} \cdot \mathbf{E}_f(m) \right] \right\} \\
& \mathbf{\Phi}_f(k, m) = \text{FFT} \left[\mathbf{\varphi}_i(k, m), \underbrace{0, 0, \dots, 0}_{N \text{ zeros}} \right]^T \\
& \mathbf{W}_f(k, m+1) = \mathbf{W}_f(k, m) + K \cdot \mu_k \cdot \mathbf{\Phi}_f(k, m) \\
& \quad \mu_k = \text{const} \\
& \text{else } \text{mod}(m, T) \neq 0, \\
\mu_k(m) &= \begin{cases} \frac{\sum_{i=0}^{N-1} |w_i(k, m)|}{\sqrt{L} \cdot \sqrt{\sum_{j=0}^{L-1} (w_j(m))^2 + \varepsilon}}, & \text{if } k \text{ corresponds to } M \text{ maxima of } \mu_k(m), k = 0, \dots, K-1 \\ 0, & \text{otherwise} \end{cases} \\
& \mathbf{W}_f(k, m+1) = \mathbf{W}_f(k, m) + K \cdot \mu_k \cdot \mathbf{\Phi}_f(k, m)
\end{aligned}$$

4.2 Discussion over experimental results

To compare the performance of the proposed algorithm with reference MDF, IPMDF, SC-IPMDF adaptive filtering algorithms, we implement all four adaptive filters in MATLAB environment. These MDF-based filters estimate the impulse response of the predefined echo paths, which are specified by the ITU-T Recommendation (ITU-T G.168, 2002). The tested impulse responses are 1024 taps long. All experiments are performed using pre-recorded real speech signals. The number K of sub-filters equals 8. The following values for control parameters are used: $\mu_{\text{initial}} = 0.1$, $a_{\text{initial}} = -0.75$, $\varepsilon = 10^{-3}$, $\Delta_\eta = 0.05$ dBm, $T_\eta = 8$. The

performance of each algorithm is studied using the normalized misalignment parameter, which can be estimated as follows

$$MIS = 10 \cdot \log_{10} \left(\frac{\|\mathbf{h} - \mathbf{w}(m)\|_2}{\|\mathbf{h}\|_2} \right) \text{ in } [dB] \quad (57)$$

where \mathbf{h} is a true impulse response of length L . Another criterion is Echo Return Loss Enhancement (ERLE), which is used in real-life environment to evaluate performance

$$ERLE = 10 \cdot \log_{10} \left(\frac{\|\mathbf{y}(m) - \mathbf{d}(m)\|_2}{\|\mathbf{y}(m)\|_2} \right), [dB] \quad (58)$$

where $\mathbf{y}(m)$ is a desired signal (echo) and $\mathbf{d}(m)$ is adaptive filter's output. Note that any reliable adaptive filter with disabled residual echo suppressor has to achieve ERLE of -15dB within 1 second after starting convergence process (ITU-T G.131, 2003). Fig. 15, 16, 17 illustrate an application of μ -based selection metric. This kind of metric is used for sub-filter selection, when estimated sparse measure parameter, ξ_z equals or larger than 0.7. This value was defined experimentally during multiple trials for numerous types of echo path. We suggest using μ -based selection metric as an individual block step-size parameter. It helps accelerating a speed of convergence by allocating larger step-size values for currently updated sub-filters. If you look at the diagram illustrated in Fig. 17 carefully, you will notice that the energy, which is available for adaptation, is concentrated around the sparse region of the echo path. Thus, this fact can be used for selecting sub-filters to be updated along with setting the step-size parameter for these sub-filters. When estimation of sparse measure, ξ_z is smaller than 0.7, we suggest switching to the χ - based selection metric.

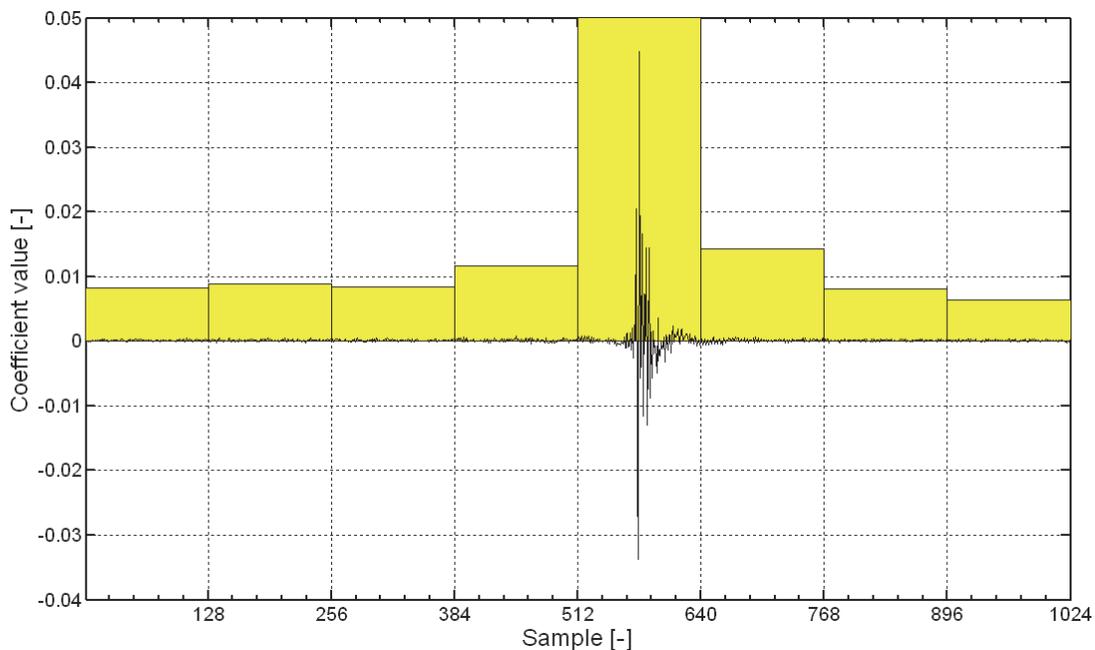


Fig. 15. Sparse impulse response and estimated μ -based selection metrics

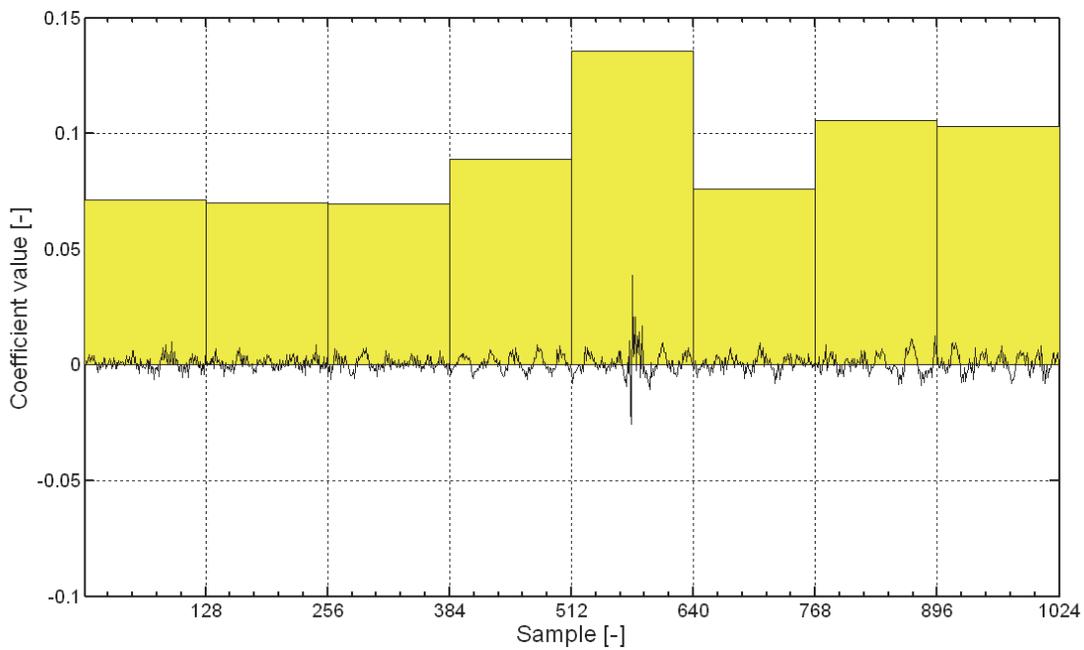


Fig. 16. Dispersive impulse response and estimated μ -based selection metrics

Fig. 18 demonstrates the normalized misalignment and ERLE parameters obtained for real speech signals. The proposed partially updated scheme for MDF shows the similar performance comparing to the other three fully updated frequency domain algorithms. During our future work we are going to enhance the above described algorithm and propose a new class of partial sparse-controlled robust algorithms, which will work reliably, even in double-talk situation. We will apply all the knowledge, which were presented within this particular chapter. Further to conclude the chapter, let us provide summary of material and make several contributions according to the proposed algorithms.

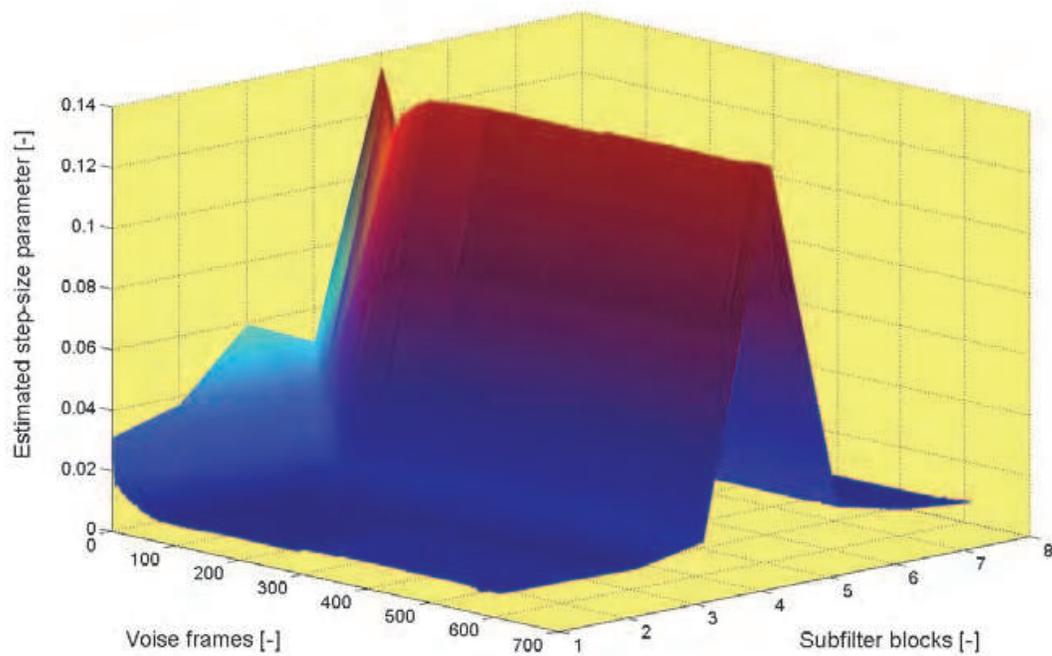


Fig. 17. Step-size parameter estimated using μ -based selection metric

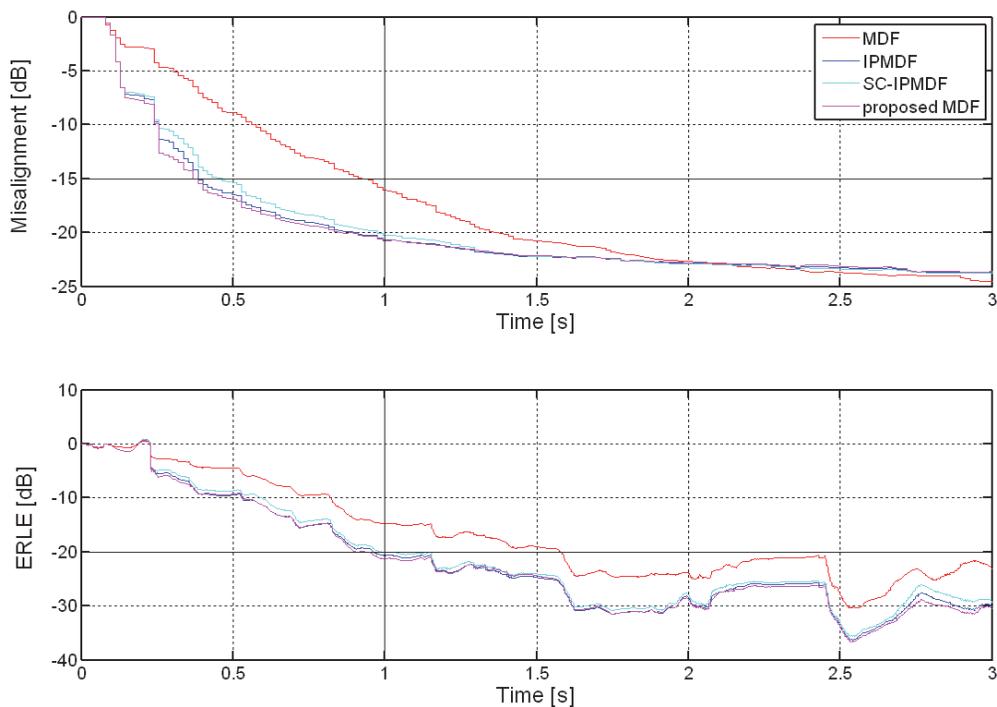


Fig. 18. Misalignment and ERLE curves

5. Conclusion

The first section outlines a basic principle of echo control in packet-based networks. It explains why it is so important to provide monitoring during telephone conversations. When delivering the VoIP service in the packet-switching network, it is important to have the value of the echo delay under control. The increasing transmission delay associated with packet data transmission can make a negligible echo more annoying. Therefore, it is suggested using the echo assessment algorithm. Its purpose is to add an additional attenuation to a particular voice channel (which in terms means to activate an echo canceller), so as to remove the unwanted echo in time. In the second section, we consider an opportunity of using cross-correlation for estimating echo delays. That section provides readers along with up-to-date correlation-based TDE algorithms, which we use to estimate the echo path delays. The problem of long delays taken place in the packet-switching network is considered as a topic of interest. The experiments show that the algorithms precision decreases with increasing transmission delays. The generalized cross-correlation algorithms operating in the frequency domain provide more reliable result comparing to the standard cross-correlation and normalized cross-correlation algorithms. As an alternative to correlation-based methods, techniques, which use adaptive filtering algorithms, can be also applied. Therefore, the third section presents numerous partial-update algorithms and their application to delay estimation. The echo assessment is based on the reduced complexity partial-update adaptive filters. The experiments show a reliable performance of these algorithms. However, their precision suffers during the initial stage of convergence. According to the ITU-T Recommendation G.168, this period should not last more than one second. The Multi-Delay block Frequency domain (MDF) adaptive algorithm can easily

outperform all existing time domain algorithms. Moreover, taking into the account the fact that the generalized cross-correlation algorithms operate in the frequency domain and use advantages of the fast Fourier transform, further computational savings for the adaptive filters are achieved in the frequency domain. Therefore, the fourth section deals with partial, proportionate, and sparse-controlled adaptive filtering algorithms working in the frequency domain. What we claimed, within this section, is: a new metric for performing partial updating; a new approach for designating transitions between MDF and IPMDF-based updating schemas; a method for estimating step-size control parameter; a new partially updated sparseness-controlled improved proportionate multi-delay filter; all the approaches are suitable for implementation whether in time or frequency domains. The proposed algorithm has both a performance compared to the IPMDF and SC-MDF algorithms and reduced computational complexity along with the adjustable step-size parameter. Although the preferred embodiments of the proposed algorithm have been described, it will be understood by those skilled in the art that various changes may be made thereto without departing from the main scope of the invention or the appended claims.

6. Acknowledgment

This work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS 10/275/OHK3/3T/13 and by Grant The Ministry of Education, Youth and Sports No. MSM6840770014.

7. References

- Choi, B.-K.; Moon, S.; Zhi-Li, Z. (2004) Analysis of Point-To-Point Packet Delay In an Operational Network, *Proceedings of INFOCOM 2004, Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp.1797-1807, ISBN 0-7803-8355-9, Hong Kong, March 7-11, 2004
- Gordy, J.D.; Goubran, R.A. (2006) On the Perceptual Performance Limitations of Echo Cancellers in Wideband Telephony, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, issue 1, pp.33-42, ISSN: 1558-7916
- Nisar, K.; Hasbullah, H.; Said, A.M. (2009) Internet Call Delay on Peer to Peer and Phone to Phone VoIP Network, *Proceedings of ICCET '09, International Conference on Computer Engineering and Technology*, vol. 2, pp.517-520, ISBN 978-1-4244-3334-6, Singapore, Januar 22-24, 2009
- Dyba, R.A. (2008) Parallel Structures for Fast Estimation of Echo Path Pure Delay and Their Applications to Sparse Echo Cancellers, *Proceedings of CISS 2008, 42nd Annual Conference on Information Sciences and Systems*, pp.241-245, ISBN 978-1-4244-2246-3, Princeton, NJ, March 19-21, 2008
- Hongyang, D.; Dyba, R.A. (2008) Efficient Partial Update Algorithm Based on Coefficient Block for Sparse Impulse Response Identification, *Proceedings of CISS 2008, 42nd Annual Conference on Information Sciences and Systems*, pp.233-236, ISBN 978-1-4244-2246-3, Princeton, NJ, March 19-21, 2008
- Khong, A.W.H.; Naylor, P.A. (2006) Efficient Use Of Sparse Adaptive Filters, *Proceedings of ACSSC '06, Fortieth Asilomar Conference on Signals, Systems and Computers*, pp.1375-1379, ISBN 1-4244-0784-2, Pacific Grove, CA, October 29 -November 1, 2006

- Hongyang, D.; Dyba, R.A. (2009) Partial Update PNLMS Algorithm for Network Echo Cancellation, *Proceedings of ICASSP 2009, IEEE International Conference on Acoustics, Speech and Signal Processing*, pp.1329-1332, ISBN 978-1-4244-2353-8, Taipei, April 19-24, 2009
- ITU-T Recommendation G.131 (2003) *Talker Echo and its Control*
- ITU-T Recommendation G.168 (2002) *Digital network echo cancellers*
- Carter, G.C. (1976) *Time Delay Estimation*, Ph.D. dissertation, University of Connecticut, Storrs, CT, pp.67-70
- Mueller, M. (1975) Signal Delay, *IEEE Transactions on Communications*, vol. 23, issue 11, pp.1375-1378
- Buchner, H.; Benesty, J.; Gansler, T.; Kellermann, W. (2006) Robust Extended Multidelay Filter and Double-talk Detector for Acoustic Echo Cancellation, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, issue 5, pp.1633-1644, ISSN 1558-7916
- Youn, D.H.; Ahmed, N.; Carter, G.C. (1983) On the Roth and SCOTH Algorithms: Time-Domain Implementations, *In Proceedings of the IEEE*, vol. 71, issue 4, pp.536-538, ISSN 0018-9219
- Zetterberg, V.; Pettersson, M.I.; Claesson, I. (2005) Comparison Between Whitened Generalized Cross-correlation and Adaptive Filter for Time Delay Estimation, *Proceedings of MTS/IEEE, OCEANS*, vol. 3, ISBN 0-933957-34-3, Washington D.C., September 17-23, 2005
- Hertz, D. (1986) Time Delay Estimation by Combining Efficient Algorithms and Generalized Cross-correlation Methods, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, issue 1, pp.1-7, ISSN 0096-3518
- Knapp, C.; Carter, G.C. (1976) The Generalized Correlation Method for Estimation of Time Delay, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 24, issue 4, pp.320-327, ISSN 0096-3518
- Wilson, K.W.; Darrell, T. (2006) 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, pp.2156-2164, ISSN 1558-7916
- Tianshuang, Q.; Hongyu, W. (1996) An Eckart-weighted adaptive time delay estimation method, *IEEE Transactions on Signal Processing*, vol. 44, issue 9, pp.2332-2335, ISSN 1053-587X
- Chen, J.; Benesty, J.; Huang, Y.A. (2006) The SCOT Weighted Adaptive Time Delay Estimation Algorithm Based on Minimum Dispersion Criterion, *Journal of EURASIP on Applied Signal Processing*, vol. 2006, ISBN 978-1-4244-7047-1
- Anderson, M.P.; Woessner, W.W. (1992) *Applied Groundwater Modeling: Simulation of Flow and Advective Transport*, Academic Press (2nd Edition ed.), ISBN 978-0120594856, USA
- Widrow, B. (2005) Thinking about thinking: the discovery of the LMS algorithm, *IEEE Magazine on Signal Processing*, vol. 22, issue 1, pp.100-106, ISSN 1053-5888
- Haykin, S. (2001) *Adaptive Filter Theory, Fourth Edition*, Prentice-Hall, ISBN 0130901261, USA
- Zetterberg, V.; Pettersson, M.I.; Claesson, I. (2005) Comparison Between Whitened Generalized Cross-correlation and Adaptive Filter for Time Delay Estimation, *Proceedings of MTS/IEEE, OCEANS*, vol. 3, ISBN 0-933957-34-3, Washington D.C., September 17-23, 2005

- Emadzadeh, A.A.; Lopes, C.G.; Speyer, J.L. (2008) Online time delay estimation of pulsar signals for relative navigation using adaptive filters, *Proceedings of 2008 IEEE/ION, Position, Location and Navigation Symposium*, pp.714-719, ISBN 978-1-4244-1536-6, Monterey, CA, May 5-8, 2008
- Duttweiler, D.L. (2000) Proportionate normalized least-mean-squares adaptation in echo cancellers, *IEEE Transactions on Speech and Audio Processing*, vol. 8, issue 5, pp.508-518, ISSN 1063-6676
- Hongyang, D.; Doroslovacki, M. (2006) Proportionate adaptive algorithms for network echo cancellation, *IEEE Transactions on Signal Processing*, vol. 54, issue 5, pp.1794-1803, ISSN: 1053-587X
- Paleologu, C.; Benesty, J.; Ciochina, S. (2010) An improved proportionate NLMS algorithm based on the l0 norm, *Proceedings of ICASSP '10, 2010 IEEE International Conference on Acoustics Speech and Signal Processing*, pp.309-312, ISSN 1520-6149, Dallas, TX, March 14-19, 2010
- Gay, S.L. (1998) An efficient, fast converging adaptive filter for network echo cancellation, *Proceedings of the Thirty-Second Asilomar Conference on Signals, Systems & Computers*, vol. 1, pp.394-398, ISBN 0-7803-5148-7, Pacific Grove, CA, November 1-4, 1998
- Benesty, J.; Gay, S.L. (2002) An improved PNLMS algorithm, *Proceedings of ICASSP '02, 2002 IEEE International Conference on Acoustics Speech and Signal Processing*, vol. 2, pp.1881-1884, ISSN 1520-6149, Minneapolis, MN, USA, April 27-30, 2002
- Fevrier, I.J.; Gelfand, S.B.; Fitz, M.P. (1999) Reduced complexity decision feedback equalization for multipath channels with large delay spreads, *IEEE Transactions on Communications*, vol. 47, issue 6, pp.927-937, ISSN 0090-6778
- Douglas, S.C. (1997) Adaptive filters employing partial updates, *IEEE Transactions on Circuits and Systems II*, vol. 44, issue 3, pp.209-216, ISSN 1057-7130
- Aboulnasr, T.; Mayyas, K. (1999) Complexity reduction of the NLMS algorithm via selective coefficient update, *IEEE Transactions on Signal Processing*, vol. 47, issue 5, pp.1421-1424, ISSN 1053-587X
- Aboulnasr, T.; Mayyas, K. (1998) MSE analysis of the M-Max NLMS adaptive algorithm, *Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing*, vol. 3, article ID 10.1109/ICASSP.1998.681776, pp.1669-1672, ISBN 0-7803-4428-6, Seattle, WA, May 12-15, 1998
- Schertler, T (1998) Selective block update of NLMS type algorithms, *In Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing*, vol. 3, pp.1717-1720, ISBN 0-7803-4428-6, Seattle, WA, May 12-15, 1998
- Dogancay, K.; Tanrikulu, O. (2001) Adaptive filtering algorithms with selective partial updates, *IEEE Transactions on Circuits and Systems II*, vol. 48, issue 8, pp.762-769, ISSN 1057-7130
- Naylor, P.A.; Sherliker, W. (2003) A short-sort M-Max NLMS partial-update adaptive filter with applications to echo cancellation, *Proceedings of ICASSP '03, 2003 IEEE International Conference on Acoustics Speech and Signal Processing*, vol. 5, pp.373-376, ISBN 0-7803-7663-3, Hong Kong, April 7-10, 2003
- Jinhong, W.; Doroslovacki, M. (2008) Partial update NLMS algorithm for sparse system identification with switching between coefficient-based and input-based selection, *Proceedings of CISS 2008, 42nd Annual Conference on Information Sciences and Systems*, vol.3, pp.237-240, ISBN 978-1-4244-2246-3, Princeton, NJ, March 19-21, 2008

- Shynk, J. J. (1992) Frequency-Domain and Multirate Adaptive Filtering, *IEEE Transactions on Signal Processing Magazine*, vol. 9, pp. 474–475, ISSN: 1053-5888
- Widrow, B.; Stearns, S.D. (1985) *Adaptive Signal Processing*, Prentice-Hall, ISBN 0130040290, USA
- Khong, A. W. H.; Naylor, P. A.; Benesty, J. (2007) A low delay and fast converging improved proportionate algorithm for sparse system identification, *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2007, Article ID 84376, 8 pages
- Khong, A. W. H.; Xiang, L.; Doroslovacki, M.; Naylor, P. A. (2008) Frequency domain selective tap adaptive algorithms for sparse system identification, *Proceedings of ICASSP 2008, International Conference on Acoustics, Speech and Signal processing*, pp. 229–233, ISBN 978-1-4244-1483-3, Las Vegas, NV, March 31-April 4, 2008
- Deng, H.; Dyba, R. (2008) Efficient Partial Update Algorithm Based on Coefficient Block for Sparse Impulse Response Identification, *Proceedings of CISS 2008, Conference on Information Sciences and Systems*, pp. 233-236, ISBN 978-1-4244-2246-3, Princeton, NJ, March 19-21, 2008
- Benesty, J.; Huang, Y. A.; Chen, J.; Naylor, P. A. (2006) Adaptive algorithms for the identification of sparse impulse responses, In: *Selected Methods for the Cancellation of Acoustical Echoes, the Reduction of Background Noise, and Speech Processing*, Hänslér, E.; Schmidt, G., Springer, pp 125-153, ISBN 978-3-540-33212-1, Berlin

IntechOpen



Adaptive Filtering Applications

Edited by Dr Lino Garcia

ISBN 978-953-307-306-4

Hard cover, 400 pages

Publisher InTech

Published online 24, June, 2011

Published in print edition June, 2011

Adaptive filtering is useful in any application where the signals or the modeled system vary over time. The configuration of the system and, in particular, the position where the adaptive processor is placed generate different areas or application fields such as: prediction, system identification and modeling, equalization, cancellation of interference, etc. which are very important in many disciplines such as control systems, communications, signal processing, acoustics, voice, sound and image, etc. The book consists of noise and echo cancellation, medical applications, communications systems and others hardly joined by their heterogeneity. Each application is a case study with rigor that shows weakness/strength of the method used, assesses its suitability and suggests new forms and areas of use. The problems are becoming increasingly complex and applications must be adapted to solve them. The adaptive filters have proven to be useful in these environments of multiple input/output, variant-time behaviors, and long and complex transfer functions effectively, but fundamentally they still have to evolve. This book is a demonstration of this and a small illustration of everything that is to come.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Kirill Sakhnov, Ekaterina Verteletskaya and Boris Simak (2011). Perceptual Echo Control and Delay Estimation, Adaptive Filtering Applications, Dr Lino Garcia (Ed.), ISBN: 978-953-307-306-4, InTech, Available from: <http://www.intechopen.com/books/adaptive-filtering-applications/perceptual-echo-control-and-delay-estimation>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen