Network Echo Cancellers: Requirements, Applications and Solutions

by Perry P. He, Roman A. Dyba, and Lúcio F.C. Pessoa
Motorola Inc.

Echo in the telephone network is a well-known phenomenon -- long delays and noticeable echo may create significant or even unbearable disturbance in a telephone conversation. Echo cancellation is a technique to reduce the echo to an acceptable level. In fact, network echo canceller is considered as the single most critical element impacting overall voice quality in typical communication networks. The challenge is to maintain the Quality of Service (QoS) at low computational cost. In order to meet the challenge echo canceller vendors have made great efforts to develop carrier-grade echo cancellers aiming at offering high voice quality and high channel density solutions. Key elements in the competitive marketplace include advanced and better tuned echo cancellation algorithms, better processor architectures with more processing capabilities, and better implementation techniques. This article covers the following aspects of the echo cancellation: (1) General market requirements; (2) Typical echo cancellation architectures and applications; (3) Basic echo cancellation algorithms; (4) Echo cancellation implementations; (5) Echo canceller testing and deployment related issues.

Market Requirements

The market requirements for echo cancellers have evolved dramatically over the years. The most important may be simplified as voice quality and cost. To be more specific, the major market requirements can be summarized as follows.

<u>1. ITU-T G.168 Compliance</u>

It is a commonly-established concept in the market place that an echo canceller should be compliant with ITU-T (International Telecommunication Union -- Telecommunication Standardization Sector) G.168 Recommendation [1]. G.168 outlines some general performance requirements for a network echo canceller, and defines 14 sets of tests (some are optional and/or still under study) for ensuring a certain level of echo cancellation performance, including convergence rate and depth, double-talk detection, nonlinear processing, comfort noise generation, tone handling, stability, etc. ITU-T G.168 Recommendation has evolved over the last decade yet it is still a subject of further study and discussion. The major revisions of ITU-T G.168 are the ones published in 1997, 2000 and 2002. There is an on-going discussion whether to enhance G.168 further for one or more revisions, or to issue a new recommendation.

ITU-T G.168, unlike some other ITU-T recommendations (voice compression algorithms, such as G.729, G.723.1, etc.), does not specify any algorithms to use for echo cancellation. The label of "ITU-T G.168 compliance" is a vague statement because some echo canceller vendors test their solutions against some, not all, of the G.168 tests due to specific application needs or some other reasons, and a well-tuned echo canceller may marginally fail some G.168 tests due to specific design trade-offs to enhance other aspects of the echo canceller performance.

Some other ITU-T recommendations, besides G.168, may also be relevant for a network echo canceller performance evaluation. These recommendations may include ITU-T G.165 [2], a predecessor and a sub-set of G.168, and several ITU-T P-series recommendations for voice quality assessment (cf. [4]-[7].)

2. Voice Quality

To ensure voice quality of an echo canceller, in addition to performing G.168 tests, it is essential to conduct extensive voice quality assessment. As a matter of fact, echo canceller vendors and users may pay more attention to subjective voice quality evaluation, than to the G.168 tests. Voice quality assessment is focused on the following aspects:

1. Convergence rate: To evaluate the convergence speed of an echo canceller in the initial phase of convergence, i.e. how fast the echo canceller eliminates the echo, or reaches the steady state that residual echo level which is sufficiently low to permit the use of nonlinear processing to eliminate echo
2. Convergence depth: To evaluate the level of residual echo, if any, after the initial phase of convergence
3. Double talk detection: To evaluate the performance during and after the period of double talk (the presence of both the near-end and the far-end signals, see the Echo Canceller Overview section for details)
4. Nonlinear processing: To evaluate the performance of nonlinear processing (i.e. processing to suppress signals below a defined suppression threshold level) and the impact on voice and background noise transmission, especially on the onset and low level of near-end talker signalsComfort noise generation: To evaluate the impact of audible artifacts when comfort noise (computer-generated pseudo-random noise to be inserted as background or idle channel noise when the nonlinear processing suppresses sub-threshold signals) switches on and off, in different background noise environments
5. Mid-call convergence: To evaluate the re-convergence capability of an echo canceller when hybrid characteristics change during a call.

Voice quality evaluation can be done either subjectively, or objectively. In the subjective evaluation, human subjects evaluate the perceived audio quality of specific test signals (speech, tone, noise or their combinations) under controlled environments, such as a quiet and acoustically controlled room. The evaluation can be done using real-time phone-to-phone conversation, or using preprocessed test signals. Human subjects may include a small group of experts with well-trained ears (called golden ears) for echo cancellation performance evaluation, and/or a relatively large group of listeners without special training in the area of sound/voice quality. The evaluation results are generally represented as Mean Opinion Scores (MOS.)
The subjective evaluation is comprehensive but expensive in terms of resource usage (human and time.) There has been a very significant effort in developing objective evaluation methods of estimating the subjective voice quality, in order to minimize the resource usage. Many algorithms or methods have been proposed over the years, and some of them have been adopted as ITU-T recommendations, such as PAMS (Perceptual Analysis Measurement System), PSQM

(Perceptual Speech Quality Measure), and PESQ (Perceptual Evaluation of Speech Quality) [4]. These objective methods, however, cannot replace the subjective listening completely, simply because nothing has been as effective as "golden ears." It is expected that more R&D effort will be spent in this area.

3. Echo Path Coverage

It is believed that echo path coverage of 24 to 32 ms is normally sufficient for most, if not all, of PSTN and packet telephony applications. Nevertheless, there is a trend to expand echo path coverage to 128 ms mainly dictated by market requirements and competition. Most echo canceller vendors nowadays offer echo cancellers with 64 to 128 ms echo path coverage.

4. Processing Load

Processing load of an echo canceller is commonly expressed in MIPS (Million Instructions Per Second, for a single ALU processor) or MCPS (Million Cycles Per Second, for a multiple ALU processor.) The processing load impacts directly on voice channel density per device to be supported or unit cost per voice channel.

5. Other Factors

In addition to the above four market requirements for an echo canceller there are some other requirements that may be on the list, such as memory usage, integration flexibility, control functionality, debugging capability, etc. depending upon the applications.

Echo Canceller Overview

This section will provide an overview of echo canceller functionality, as well as typical applications.

1. The Sources Of Echo

The basic telephone network consists of two types of wire segments: A four-wire central network and a two-wire local network (see Fig. 1.) The two-wire network includes a subscriber loop and some portion of the local network with the choice of two wires mainly for economics. The four-wire central network separates the two directions of signal transmission, using one pair of wires for each. The two-wire local network, on the other hand, carries signal transmission in both directions in the same pair of wires. A converting device, called hybrid, is needed at the junction of the two-wire to four-wire segments, the impedance mismatch of which is one source of echo. The echo canceller (Fig. 1, again) is a 4-port device in which two inputs are R_{in} (Receiving-path input, or far-end signal) and S_{in} (Sending-path input, a mixture of near-end signal S_{gen}, and the echo of far-end signal.) Two outputs are R_{out} (Receiving-path output, same as R_{in} for a G.168 compliant echo canceller), and S_{out} (Sending-path output, the output from the echo canceller.) When a far-end user talks, the speech signal travels through the near-end echo path, and a portion of the signal is reflected back to the far-end listener, due to the impedance mismatch in the hybrid. This type of the echo is called electric echo or circuit echo.

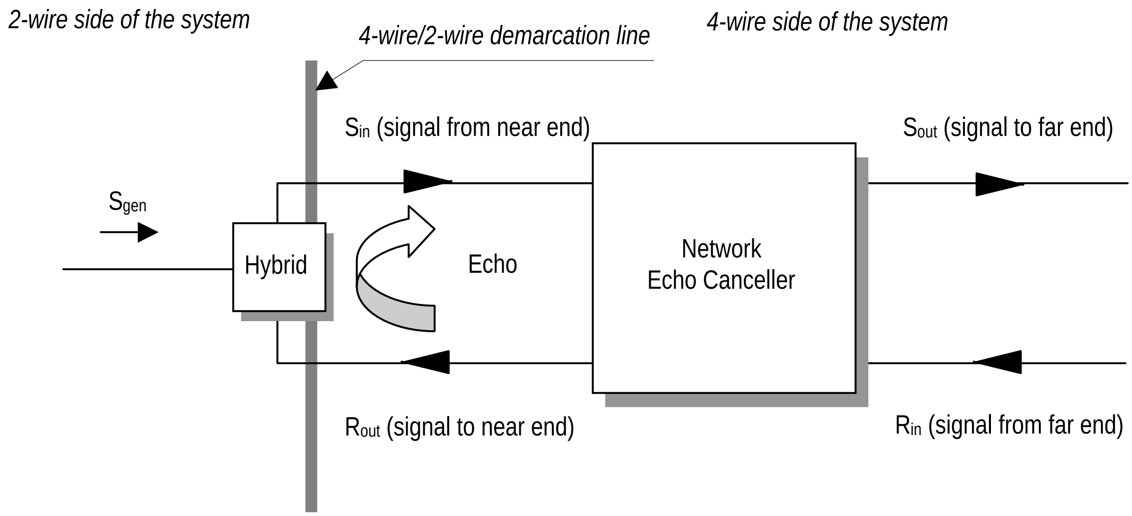

Fig. 1: Network Echo Canceller As A Four-Port Device

Acoustic echo is another type, where the source is acoustic environments, in which an echo path is the acoustic path from earphone to microphone. The scope of this article is limited to the discussion of electric echo cancellation related to the impedance mismatch of the hybrid. Readers may find discussions on acoustic echo cancellation in various references, for example, in ITU-T G.167 [3], which standardizes the basic requirements for an acoustic echo canceller.

2. Major Components Of An Echo Canceller

The basic functionality of an echo canceller is to use two inputs (R_{in} and S_{in}), using sophisticated algorithms/controls to generate S_{out}, in which the echo of R_{in} is minimized while S_{gen} is preserved with minimum degradation. Even though algorithms vary greatly among different echo cancellation solutions, major components should include:

- Adaptive filter
- Double-talk detector or near-end signal activity detector
- Nonlinear processor
- Comfort noise generator
- Signaling tone detector (external tone detector may be used for some echo cancellers)
- Echo canceller controller

A simplified layout of an echo canceller (Fig. 2) includes the above mentioned major components, except for the signaling tone detector, as some designs may use external tone detectors. Two optional high-pass filters (HPFs) are also shown in the diagram.

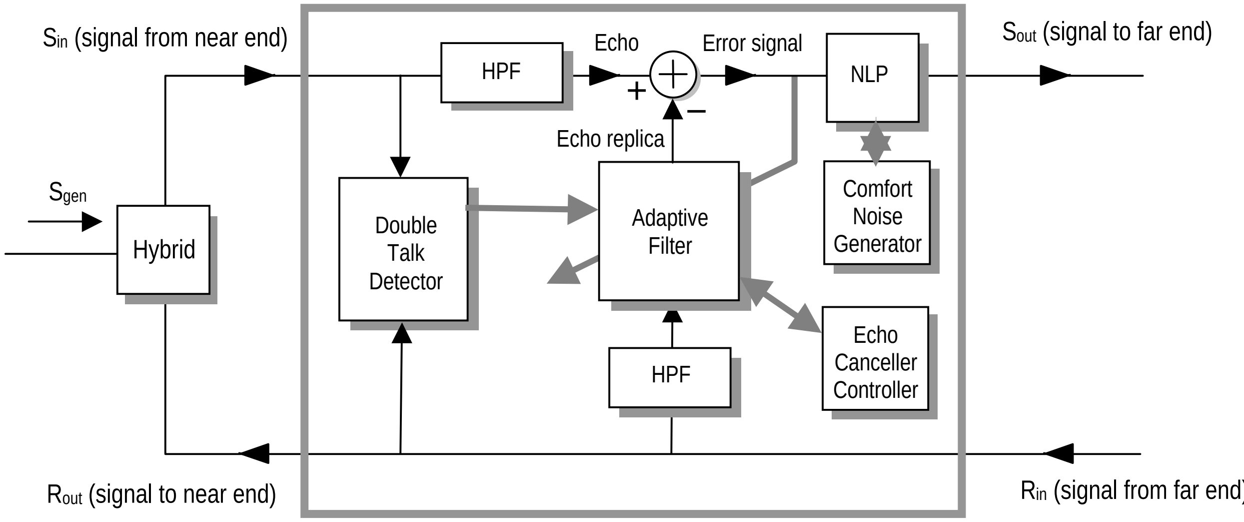

Fig. 2: Simplified View Of A Typical Echo Canceller With Its Major Components: NLP = Nonlinear Processor; Solid Lines = Signal Paths, Shaded Lines = Control Paths

<u>3. Network echo canceller applications</u>

Echo canceller performance has substantial impact on QoS for voice networks and it is important to properly place them in networks to achieve the optimal performance. The following two sub-sections briefly describe the proper placement of echo cancellers in PSTN or Packet Telephony.

Network Echo Cancellers In PSTN:
For traditional PSTN network applications, the echo canceller is placed close to the near-end hybrids, one on each side of the network (Fig. 3.)

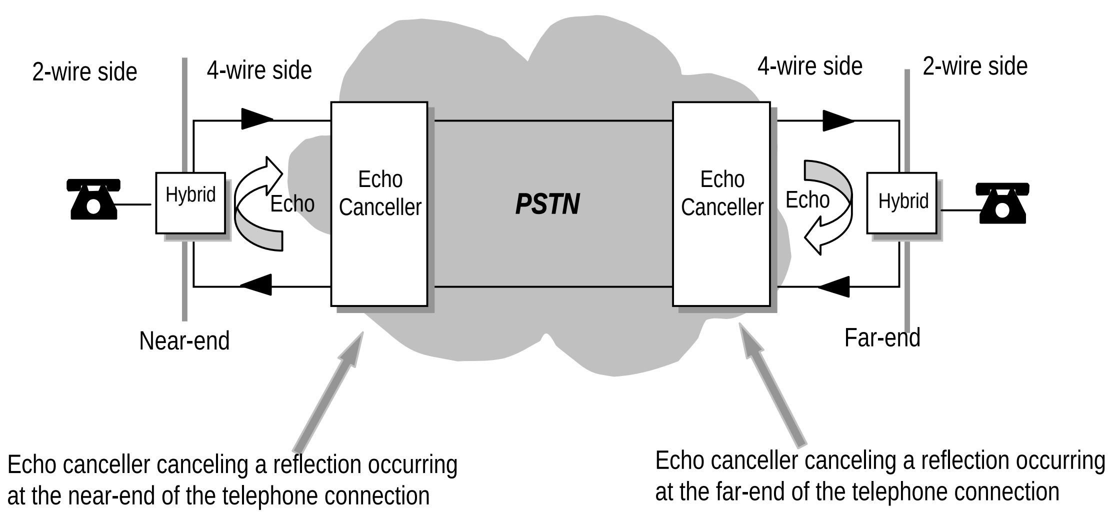

Fig. 3: Network Echo Cancellers Are Deployed At Two Ends Of The Telephone Connection

There are certain minimum requirements for the PSTN applications in order to have the echo canceller functioning properly. The requirements include the minimum value of echo return loss (ERL) of the hybrid (typically it should be at least 6 dB), and the maximum value of echo path delay (which must be within the echo span of the echo canceller.) The echo path span of the echo canceller should cover the pure echo bulk delay and echo dispersion time combined.

Network Echo Cancellers In Packet Telephony Networks
The placement of echo cancellers in Packet Telephony networks is identical to that in PSTN networks. Fig. 4 shows a typical application of echo cancellers in relation to the other components in the network.

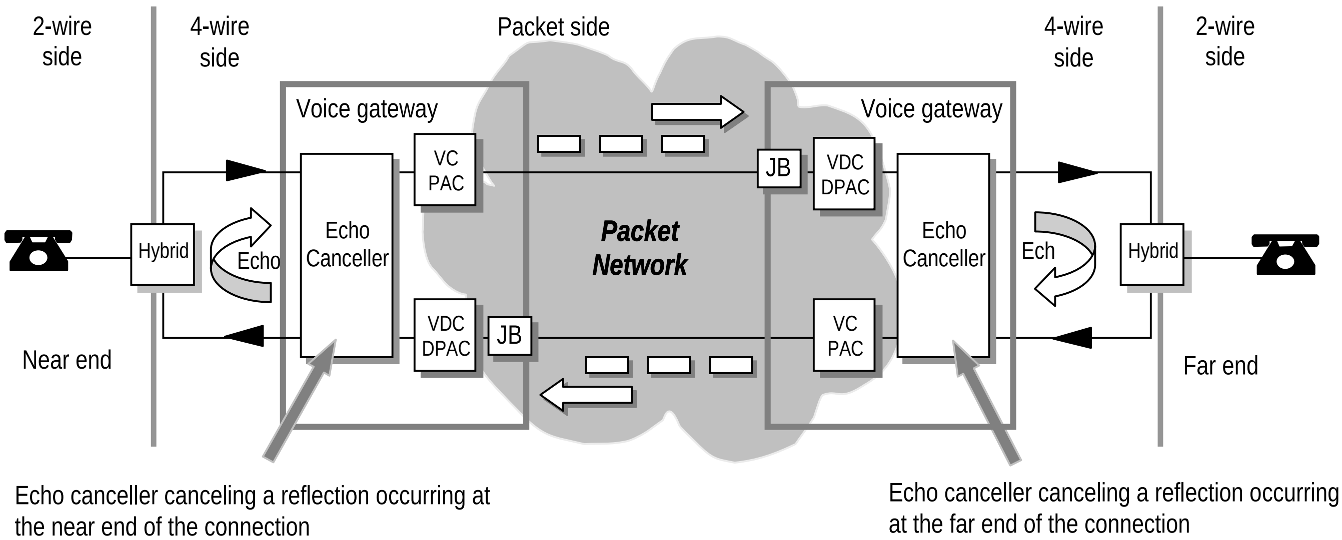

Fig. 4: Simplified Packet Telephony Network With Echo Cancellers Installed At Two Ends; VC PAC = Voice Coder/Packetizer; VDC DPAC = Voice Decoder/Depacketizer; JB = Jitter Buffer

The basic requirements for the minimum ERL of the hybrid and the maximum echo path delay for the echo canceller to operate on are the same as the applications in PSTN networks. The transmission path from Rout to Sin of the echo canceller should not include any codecs (except G.711) and jitter buffers. It is worth mentioning that the round-trip delay for a packet telephony network includes not only the transmission delay, which is similar to a PSTN network, but also the processing delays (such as the delays caused by voice compressing/decompressing, jitter buffering, etc.). Therefore, without an echo canceller, echo may be very noticeable even for a telephony conversation over a relatively short distance.

Echo Cancellation Algorithms Overview

Since the introduction of digital echo cancellation algorithms in the late 1960s, various adaptive filter algorithms have found applications in echo cancellation (see [8] for a summary.) As a brief overview, a few algorithms widely used in the echo cancellation designs are listed as follows.

Least Mean Square (LSM) Adaptive Filter Algorithms

One of the frequently used forms of adaptive filter is a tapped-delay-line finite impulse response (FIR) filter and a reference-matching quality assessment block (see Fig. 5.)

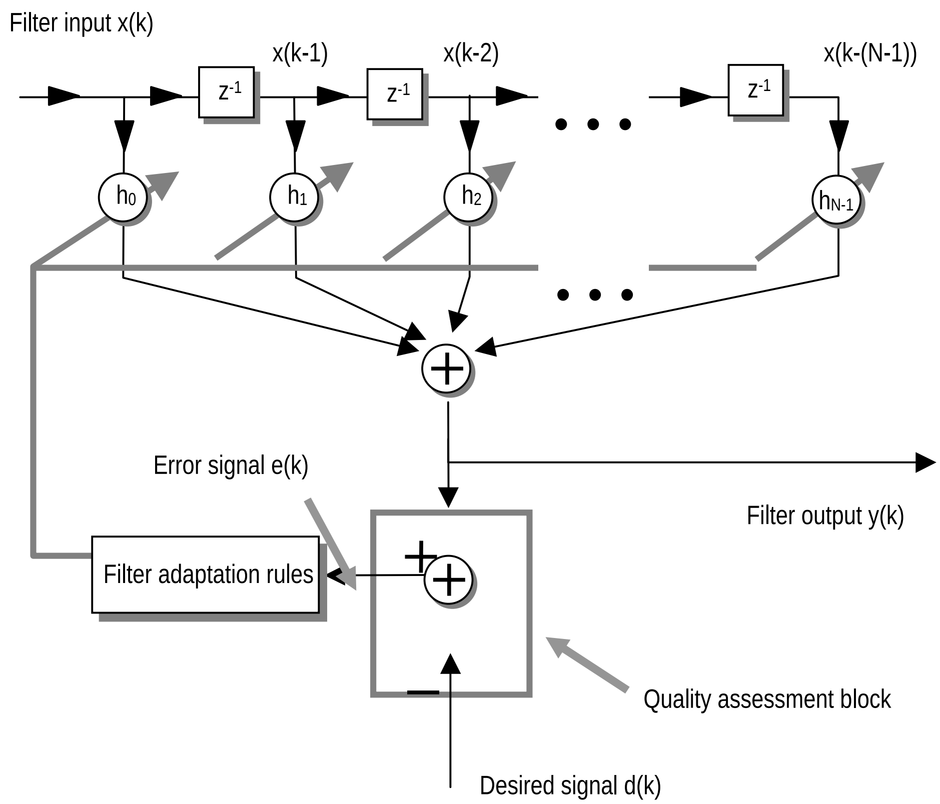

Fig. 5: Configuration Of An Adaptive Filter Controlled By An Error Signal Based On A Classical Architecture Of Linear Optimum Discrete-Time Filters Also Known As Wiener Filters [9].
N = Filter Length; Z^{-1} = One-Sample Delay Block; h_k = Multiplying Factors Or Filter Coefficients

The mapping of generic signals *x(k), y(k), e(k)* and *d(k)* (Fig. 5 to a G.168 echo canceller is:

x(k) = $R_{in}(k)$,
y(k) = Echo replica of $S_{in}(k)$,
e(k) = Signal entering the NLP block,
d(k) = $S_{in}(k)$.

Most algorithms used for echo cancellation employ FIR filters because of their simplicity (less CPU cycle consumption) and stability. IIR filters may also be used but not as commonly.

The output of the FIR filter is given by:

$$y(k) = \sum_{i=0}^{N-1} h_i \cdot x(k-i)$$

To provide a concise description of the LMS algorithm let us introduce, for convenience, the following notation:

Let $\boldsymbol{H} = [h_0, h_1,\ldots,h_{N-1}]^T$

and $\boldsymbol{X}(k)=[x(k), x(k-1),\ldots, x(k-N+1)]^T$

Then, the output signal can be conveniently represented as a product of two vectors, $\boldsymbol{X}$ and $\boldsymbol{H}$:

$$y(k) = \boldsymbol{X}^T(k)\cdot\boldsymbol{H} = \boldsymbol{H}^T\cdot\boldsymbol{X}(k)$$

The core of the adaptation problem is the strategy of finding the "best" impulse response $\boldsymbol{H}$. Typically, the criterion used for defining what is the "best" is the L^2 'distance' between the filter output y(k) and the desired signal d(k), i.e.:

$$J = \sum_k \left|d(k) - y(k)\right|^2$$

When J achieves the minimum value the respective $\boldsymbol{H}$ is said to be generating the best replica of the desired signal. With respect to echo canceller applications, the best $\boldsymbol{H}$ minimizes the echo (or practically makes it unnoticeable.) How to find the "best" impulse response of the hybrid circuit is the object of the given adaptive filter algorithm.

During the adaptation process, the consecutive vectors $\boldsymbol{H}$ are changing and are to approach its best form, $\boldsymbol{H}_{conv}$. Let's denote these consecutive vectors approaching $\boldsymbol{H}_{conv}$ as $\boldsymbol{H}(m)$, where m is a consecutive iteration. Therefore, for a given m, the modified vector expression for $\boldsymbol{H}(m)$ is:

$$\boldsymbol{H}(m) = [h_0(m), h_1(m),\ldots,h_{N-1}(m)]^T$$

It can be demonstrated that under certain assumptions the process of minimizing $\boldsymbol{J}$, through adaptive changes, or iterations $\boldsymbol{H}(m)$, can be practically achieved via the LMS algorithm. The algorithm can be written, for k^{th} iteration, as follows:

$$y(k) = \boldsymbol{H}^T(k)\cdot\boldsymbol{X}(k) \qquad \text{(Filter output)}$$

$$e(k) = d(k) - y(k) \qquad \text{(Error signal)}$$

$$\boldsymbol{H}(k+1) = \boldsymbol{H}(k) + \mu\cdot e(k)\cdot\boldsymbol{X}(k) \qquad \text{(Adaptation formula)}$$

where, $\boldsymbol{\mu}$ is frequently called the step size of the LMS algorithm; $\boldsymbol{k}$ is sample number, or iteration number. Because the sample number coincides with the iteration number, the algorithm is said as performing "per sample" adaptation. Over the years many other versions of the LMS algorithm have been developed. One of them is a "per block" adaptation version. Block-adaptation versions of the LMS algorithms have become popular due to their computational processing efficiency.

As the LMS algorithm governing equations indicate, there are three major computational steps required for implementing the algorithm:

- Computation of the output of the transversal (FIR) filter
- Computation of error
- Computation of updated coefficients of the FIR filter.

Normalized LMS (NLMS) Adaptive Filter Algorithms

The LMS algorithm uses the adaptation constant $\boldsymbol{\mu}$ a small constant that determines, among other things, the speed of convergence of the algorithm. One of many practical problems associated with the choice of $\boldsymbol{\mu}$ is that of finding some way to ensure that $\boldsymbol{\mu}$ does not become large enough (in relation to the input signal energy) to cause the algorithm to diverge.

Normalized LMS (NLMS) adaptation algorithms are straightforward versions of the classical LMS algorithm in which the selection of an adaptation constant is done through normalization of the original constant with respect to the signal power. Because of its simplicity, acceptable convergence speed and depth the NLMS algorithm, although requiring slightly more computational effort than its LMS prototype, has been widely used for echo cancellers, where echo path delays are not excessively large.

The algorithm governing equations are:

$$y(k) = \boldsymbol{H}^T(k) \cdot \boldsymbol{X}(k) \qquad \text{(Filter output)}$$

$$e(k) = d(k) - y(k) \qquad \text{(Error signal)}$$

$$\boldsymbol{H}(k+1) = \boldsymbol{H}(k) + \frac{\mu \cdot e(k) \cdot \boldsymbol{X}(k)}{\gamma + \boldsymbol{X}^T(k) \cdot \boldsymbol{X}(k)} \qquad \text{(Adaptation formula)}$$

where, $\dfrac{\mu}{\gamma + \boldsymbol{X}^T(k) \cdot \boldsymbol{X}(k)}$

is the NLMS algorithm step size; $\boldsymbol{\gamma}$ is a 'protection' term, which ensures that the update term in the adaptation formula does not become excessively large when $\boldsymbol{X}^T(k) \cdot \boldsymbol{X}(k)$ temporarily becomes small.

As the governing equations indicate, there are four major computational steps required for implementing the algorithm:

- Computation of the output of the transversal (FIR) filter
- Computation of error
- Computation of the far-end signal (i.e., input signal) power
- Computation of updated coefficients of the FIR filter

A closer examination of the adaptation formula shows that computation of $\boldsymbol{X}^T(k) \cdot \boldsymbol{X}(k)$ can be significantly speeded up by recursive computations of the signal power.

Proportionate NLMS (PNLMS) Adaptive Filter Algorithms

On typical echo paths the proportionate NLMS (PNLMS) adaptation algorithm leads to faster convergence time than the NLMS algorithm. The convergence depth in both cases remains very similar. The PNLMS algorithm differs from the NLMS algorithm in that the input signal energy is distributed unevenly over the N taps of the adaptive FIR filter.

While the governing equations for the PNLMS remain largely intact, the update term is somewhat modified to include ***G(k)*** factor representing the energy distribution over the filter taps; the PNLMS adaptation formula then becomes:

$$\boldsymbol{H}(k+1) = \boldsymbol{H}(k) + \frac{\mu \cdot \boldsymbol{G}(k) \cdot e(k) \cdot \boldsymbol{X}(k)}{\gamma + \boldsymbol{X}^T(k) \cdot \boldsymbol{X}(k)} \qquad \text{(Adaptation formula)}$$

where, ***G(k)*** is a diagonal matrix for energy distribution, determined via analyzing successive iterations of ***H(k)***. Equations for ***G(k)*** are not elaborated here.

Because of a significant computational penalty imposed by the modified adaptation formula combined with generation of matrix ***G(k)***, the PNLMS algorithm is more suitable for implementation in ASIC technology.

Sub-band (N)LMS Adaptive Filter Algorithms

An example of a general outline of the sub-band (N)LMS method is presented in Fig. 6.

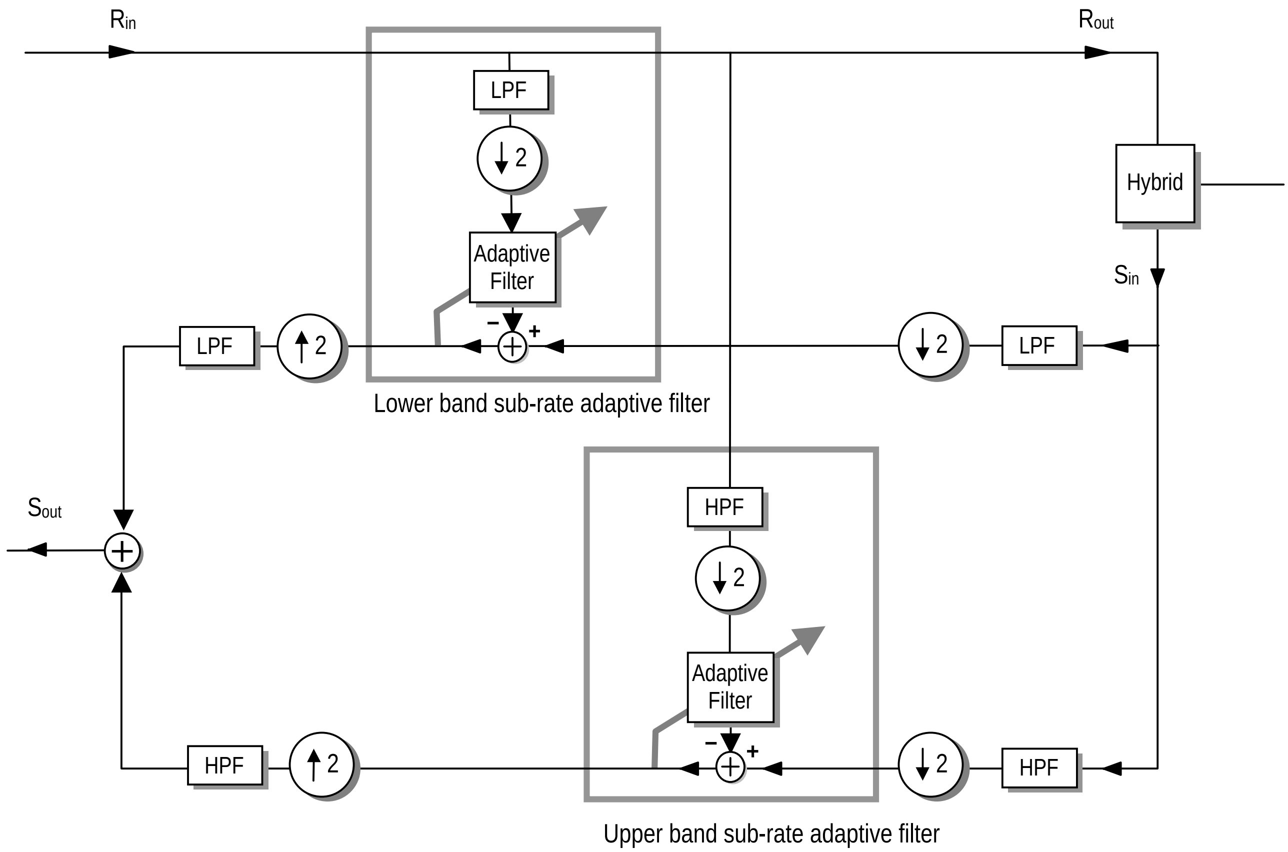

Fig. 6:General Architectural View Of A Sub-Band Echo Canceller
HPF = High-Pass Filter; LPF = Low-Pass Filter;
↑2' = Factor 2 Interpolator; ↓2' = Factor 2 Decimator

Through signal spectrum partitioning individual echo cancellers operating on signals of narrower spectrum converge significantly faster and deeper. Implementation costs of sub-band echo cancellers include, among other things, pass-band filtering and multiple adaptation processes. These costs, however, are partly offset through sub-rate processing.

Affine Projection (AP) Algorithm

The affine projection algorithm is characterized by its moderately higher computational complexity than the LMS algorithm, but significantly faster convergence speed than the NLMS algorithm. The AP algorithm can be considered as a generalization of the NLMS adaptive filtering algorithm.

By using the notation similar to the one in previous paragraphs, one can define the AP as [10]:

$$\underline{e}_n = \underline{s}_n - \boldsymbol{X}_n^T \underline{h}_{n-1}$$
$$\underline{\varepsilon}_n = [\boldsymbol{X}_n^T \boldsymbol{X}_n + \delta \boldsymbol{I}]^{-1} \underline{e}_n$$
$$\underline{h}_n = \underline{h}_{n-1} + \mu X_n \underline{\varepsilon}_n$$

The excitation signal matrix, $\boldsymbol{X}_n$, is $\boldsymbol{L}$ by $\boldsymbol{N}$, and it has the structure of:

$$\boldsymbol{X}_n = [\underline{x}_n, \underline{x}_{n-1},\ldots, \underline{x}_{n-(N-1)}]$$

where, the vectors $\underline{x}_n = [x_n, x_{n-1},\ldots, x_{n-(L-1)}]^T$ are transposed sample vectors of length L.

The adaptive filter tap weight vector is $\underline{h}_n = [h_{0,n}, h_{1,n},\ldots, h_{L-1,n}]^T$, where $h_{i,n}$ is the i^{th} tap at sample period n.

The vector $\underline{e}_n$ is of length $\boldsymbol{N}$ and represents the residual echo (aka adaptation error.) The $\boldsymbol{N}$-length vector, $\underline{s}_n$, is the system output consisting, in general, of the response of the echo path impulse response, $\underline{h}_{EP}$, to the excitation $\boldsymbol{X}_n$ combined with an additive system noise $\underline{n}_n$,:

$$\underline{s}_n = \boldsymbol{X}_n^T \underline{h}_{EP} + \underline{n}_n$$

The scalar $\boldsymbol{\delta}$ is the regularization parameter for the sample auto-correlation matrix inverse used in the formula for $\underline{\varepsilon}_n$. The step-size parameter, $\boldsymbol{\mu}$, is the relaxation factor. Similarly to the NLMS, the AP algorithm is stable for $0 \le \boldsymbol{\mu} < 2$.

If $\boldsymbol{N}$ is set to 1, the above formulas reduce to the ones representing the NLMS algorithm.

The subject of affine projection algorithms is being vigorously researched and several more advanced versions have already been published. They include, among other things, a fast AP (FAP) algorithm and a sub-band FAP [11].

Echo Canceller Implementations

1. Hardware-Based Echo Cancellers

Hardware-based echo cancellers include FPGA-based echo cancellers and ASIC echo cancellers. In addition to criteria regarding the functional performance of these echo cancellers, aspects such as channel density vs. silicon size (which can be indirectly related to the price per channel) vs. dissipated power and other strictly hardware related issues may be factors in applying hardware based echo cancellers to a specific application.

Typically, echo canceller algorithms implemented in the hardware-based systems differ from the ones implemented in software running on digital signal processors or other specialized processors. The chief reason for these differences comes from more relaxed requirements regarding the algorithm computational complexity.

2. Software-Based Echo Cancellers

Software-based echo cancellers are application functions and application modules, which coexist with other applications related to packet telephony within a software framework. Because of the high demand for processing power, particularly in the area of arithmetic computations, these functions or modules mostly run on DSPs, and less frequently are designed to run on RISC or general-purpose processors.

Single Processor DSPs
Most software-based echo cancellers are designed to run on fixed-point DSP processors. The majority of DSP instructions are performed in one CPU cycle per instruction. MIPS, as a measure of CPU workload, has been established as one of the measures of practical algorithm efficiency for Short Instruction Word architecture (i.e. non-pipelined) processors.

SIMD Architectures
A larger portion of the CPU cycle consumption for echo cancellation comes from repetitive multiply-accumulate (MAC) operations. For example, the processes related to FIR and filter coefficient updates may take 40 to 70% of the MIPS of the entire echo cancellation process, depending upon the filter length and the complexity of the surrounding control functions for the adaptive filter and signals/speech patterns. Therefore architectures with multiple processors, or additional hardware accelerators, would be beneficial in reducing the CPU cycles needed for echo cancellation. To measure the processing load required for a given application in a multi-processor architecture, the measure of MCPS (Million Cycles Per Second) is often used.

One kind of parallel processor architecture is Single Instruction-Multiple Data (SIMD) architecture. A SIMD processor can execute multiple instructions within one clock cycle. From the echo cancellation implementation perspective multiple MACs can be executed in one cycle, provided that all the data needed for the processing are ready to use. Data alignment and preparation are the two key elements in the implementation, in order to take advantage of the SIMD processor.

One example of SIMD processor is the Motorola AltiVec technology. The processor can execute 4 MACs, offering significant performance improvement. For a computationally-expensive application, such as FIR and coefficient update in echo cancellation, the CPU cycle consumption can be reduced by 60 - 80%, depending upon the overhead needed for data alignment/preparation and process control.

VLIW Architectures

Another kind of parallel processors employs Very Long Instruction Word (VLIW) architecture. This type of processor is equipped with multiple computational units to be capable of processing multiple instructions (one instruction per processor) in one cycle. One advantage of a VLIW processor over a SIMD one is that each processor can take different instructions (bounded by certain rules.)

One example of a VLIW processor is the Motorola StarCore architecture, which is equipped with 4 ALUs (Arithmetic and Logic Unit) and 2 AAUs (Address Arithmetic Unit) -- Fig. 7.

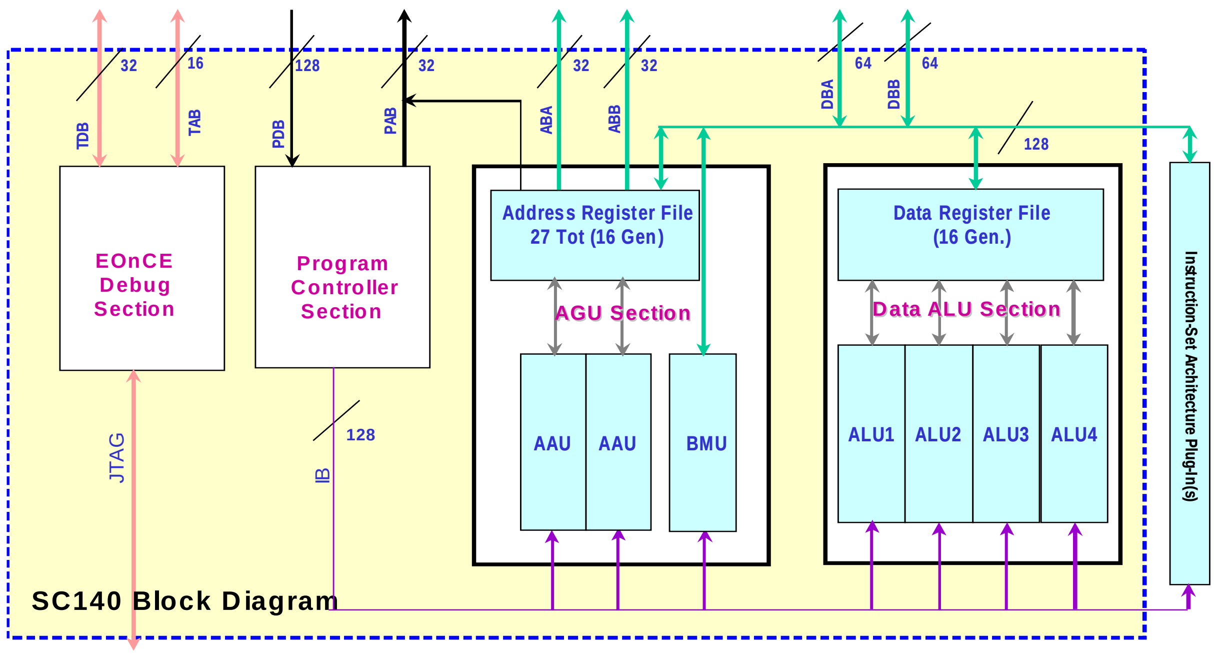

Fig. 7: StarCore SC140 Block Diagram

The architecture is capable of executing 6 instructions, including 4 arithmetic/logic and 2 data movement instructions, in one cycle. A few examples (see Fig. 8) demonstrate the efficiency and flexibility of the architecture. For echo cancellation implementation the architecture can be extremely helpful, especially for the FIR and coefficient updates, because it can perform the following tasks in one cycle: (1) Loading a set of data to be processed for the next cycle; (2) Processing a set of data loaded during last cycle (the instructions to execute can be the same, such as 4 MACs, or different, such as 2 MACs, 1 shift, and 1 logic AND); (3) Restoring a set of data already processed during last cycle.

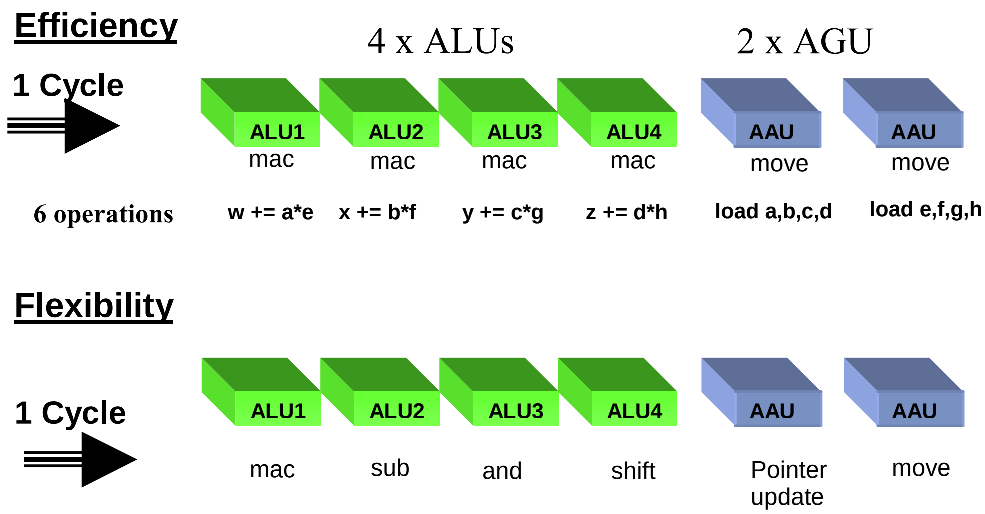

Fig. 8: Examples Of The Efficiency/Flexibility Of The StarCore SC140 Architecture

Motorola has developed a series of DSP devices based on the StarCore SC140 architecture. Examples include the MSC8101, with a SC140 DSP core, programmable Communications Processor Module (CPM) and 60x bus interface. The combination offers advanced signal processing performance, flexible network connectivity and seamless system integration. The MSC8102 includes 4 SC140 DSP cores (see Fig. 9) offering a "DSP-Farm-on-a-Chip" level of performance integration and is well suited for computation-intense DSP applications.

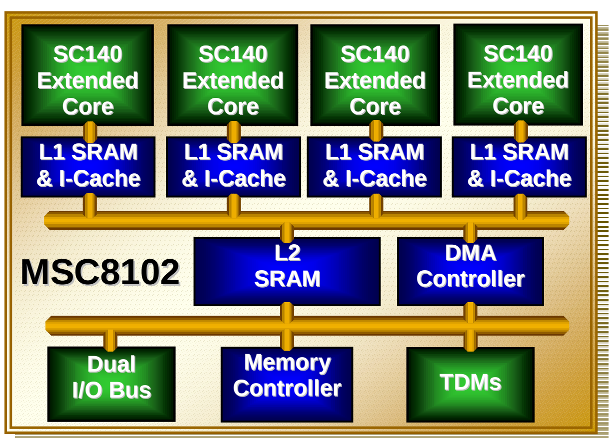

Fig. 9: Block Diagram Of Motorola MSC8102 DSP

Data alignment for VLIW processors, similar to SIMD processors discussed earlier, is also a key element in the implementation, to take full advantage of the enhanced processing capability of the architecture. The performance improvement can be up to a factor of 4, but normally is about 2.5 - 3, because of data alignment, data dependency and other limiting factors related to implementation details.

3. Echo Cancellers With Hardware Accelerators And Co-Processors

Some DSPs take different approach attempting to off-load some processing work from the CPU. One type is to have a co-processor doing repetitive MACs and to have the core do the other operations. EFCOP (enhanced filter coprocessor) is one of them. For example a Motorola DSP56307/56311 DSP comes with a core DSP and an EFCOP, which is designed for FIR and coefficient update processing. The communication between the core and the EFCOP is done by interrupts or DMA. For a typical application the core prepares data for the EFCOP and then enables the EFCOP for FIR and/or coefficient update. While the EFCOP is doing the adaptive filtering related tasks for echo cancellation the core can do other processes, either echo cancellation related or others, such as codecs. Upon the completion of the EFCOP processing, via interrupt request and interrupt service routine, the core will fetch the results from the EFCOP, load new data for the next step processing and send the EFCOP back to work again. The performance improvement through the EFCOP usage can be very significant with proper real-time scheduling and labor division between the core and the EFCOP.

Echo Canceller Testing

Since echo cancellation is the single most critical element affecting voice quality in a communication network, it is extremely important to extensively test the echo canceller. The test coverage for an echo canceller may vary, but the following tests are generally considered as essential.

G.168 Tests

G.168 tests should be the first group of tests conducted for an echo canceller. These tests are objective in nature normally including Tests 2 - 7 (Test 1 has been combined with Test 2 in G.168 2000/2002 versions) and Tests 9 - 10. The remaining tests are optional or under study. Each of the above tests includes some sub-tests. Some of the tests are considered as more important than others with the most important being Test 2 (convergence rate and depth), Test 3 (double talk performance) and Test 9 (comfort noise tests.) Considering the fact that each test should be carried out on all eight G.168 hybrid models, in a combination of different pure echo path delays within the echo span coverage the echo canceller has been designed for, the total number of G.168 tests can readily reach a few thousands.

Voice Quality Tests

Voice quality tests can be done in various ways (as described in the Market Requirement section) and may include real-time phone-to-phone subjective testing, and off-line subjective evaluation on specific testing signals. Expert listening is normally the first step in the evaluation, like echo cancellation development and performance tuning. Such an evaluation is mainly focused on echo cancellation performance under conditions of single talk and double talk with various speech levels, speech patterns and background noises, just to name a few.

Special Voice Quality Tests

In addition to the objective and subjective tests outlined above some special tests are needed for extensive echo canceller testing, which may include lab-generated tone and speech files with various signal levels, double talk patterns and near-end background noises, and field-recorded troubling signals. All those tests will help in minimizing field failures or customer complaints when the echo canceller gets deployed.

Processing Load Measurements

Echo cancellation processing load (CPU cycle consumption) is data-dependent for a given implementation. One example of data dependency is that the filter coefficients are normally not updated every data sample. There may be many reasons for not updating the filter coefficients, such as when the near-end signal is present, or the far-end signal is not strong enough. Another example of data dependency is that some of the operations of echo cancellation are executed every sample, and others are not. Therefore, it is necessary to measure the maximum and the average CPU cycle consumption for system design (such as channel density estimation and real-time scheduler design.) The maximum processing load may be calculated on a per sample basis, or over a frame (e.g. 5 or 10 ms frame size for a given application.) The maximum processing load over a frame may be very helpful in system design, because it provides the lower bound for the number of channels to be supported by the system without causing possible real-time frame loss.

Echo Canceller Deployment Issues

Many echo cancellation related issues may arise after an echo canceller has been deployed, especially in the case of a new echo cancellation design (which has not been extensively exposed to the real world of complicated network operation scenarios.) No matter how carefully the echo canceller has been tested in laboratory conditions it is unrealistic to expect that the echo canceller will be trouble-free in the field. One of the major reasons causing field problems or customer complaints is the wide range of the telephony equipment installed over the years -- the combination of the echo path delay and echo path characteristics has not been tested, or is beyond the capability of the echo canceller to handle.

Dealing with echo canceller field issues can be an ongoing battle, so it is desirable for an echo canceller vendor to be capable of identifying the echo path characteristics causing the troubles for echo cancellation. At a minimum the echo canceller vendor has to be able to identify and duplicate the troubling scenarios in order to provide proper corrective actions. These actions may include functionality enhancements focused on echo cancellation performance improvement. Sometimes, however, echo canceller design limitation (such as algorithm limitations) may be a stumbling factor in providing a satisfactory solution for a given troubling scenario. In such cases it becomes necessary to develop better and more robust echo cancellation algorithms.

Closing Remarks

Over the years, echo cancellation has evolved from a typical arithmetic (digital filter) application in telecommunications to a special discipline involving advanced DSP algorithms, sophisticated DSP architectures, state-of-the-art testing infrastructures, and specialized implementation techniques. Echo canceller vendors invest very significant resources on echo cancellation, simply because of demanding market requirements and fierce competition. Echo cancellation has been an active area for intellectual properties, and the trend continues. In addition to echo cancellation, the demands for voice quality enhancement (VQE) devices are increasing. A VQE device includes line echo cancellation, the main topic of this article, plus acoustic echo cancellation, noise reduction and automatic level control. It is believed that the increasing market requirements and competition will drive the echo cancellation technologies to a new level, where voice enhancement (compliance with all applicable standards, as well as specific customer requirements) and economy of implementation contribute to highly innovative new solutions for voice telecommunication.

References

[1] ITU-T Recommendation G.168 - Digital Network Echo Cancellers, 1997/2000/2002
[2] ITU-T Recommendation G.165 - Echo Cancellers, 1993
[3] ITU-T Recommendation G.167 – Acoustic Echo Controllers, 1993
[4] ITU-T P.862 - perceptual Evaluation of Speech Quality (PESQ), and objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs. 2001
[5] ITU-T P.831 - Subjective performance evaluation of network echo cancellers. 1998
[6] ITU-T P.800 - Methods for subjective determination of transmission quality, 1996
[7] ITU-T P.861 - Objective quality measurement of telephone-band (300-3400 Hz) speech codecs, 1998
[8] Simon Haykin, Adaptive Filter Theory, 3rd Ed, Prentice Hall, 1997
[9] D.L. Duttweiler, Proportionate Normalized Least-Mean-Squares Adaptation in Echo Cancellers, IEEE Trans. On Speech and Audio Processing, Vol.0, No. 5, pp. 508-518, September 2000.
[10] S.L. Gay and S. Tavathia, The Fast Affine Projection Algorithm, International Conference on Acoustics, Speech, and Signal Processing, ICASSP-95, Vol. 5, 9-12, pp. 3023 -3026, May 1995.
[11] Q.G. Liu et al., On the Use of a Modified Fast Affine Projection Algorithm in Subbands for Acoustic Echo Cancellation, Proc. IEEE Digital Signal Processing Workshop, pp.354-357, 1996.

About The Authors

Dr Perry P. He was awarded a PhD in electrical engineering from Rutgers, the State University of New Jersey in 1991, the MS and the BS degrees in electrical engineering from Tianjin University, PRC, in 1984 and 1982, respectively.

Prior to joining Motorola (Austin, Texas) in 1997, Dr He was a scientist at the Center for Research in Earth and Space Technology and an adjunct professor at York University (both in Toronto, Canada) from 1990-1993, a professor at DeVry Institute of Technology (Toronto, Canada) from 1993-1996, and a senior member of the technical staff at Honeywell (Toronto, Canada) from 1996-1997. Over the years, he has been working on speech signal processing, echo cancellation and voice quality enhancement devices, as well as modeling of auditory and visual processing by humans. He has 6 patents pending, and over 30 journal papers, conference proceedings and technical papers to his credit. He is a member of the IEEE and the IEEE Communication Society.
Email: perry.he@motorola.com, Tel: (512)996-4933

Dr. Roman A. Dyba graduated from Wroclaw Technical University (WTU, Wroclaw, Poland). His PhD thesis addressed spectrum evolution of nonlinearly propagated sonic waves. Prior to joining Nortel Networks, Canada, he was pursuing his academic career at the Institute of Telecommunication and Acoustics, WTU and developed numerical models of nonlinear wave propagation.

In 1987 he joined Nortel Networks, Brampton, Ontario as a senior staff engineer and individual contributor working on hardware and firmware projects for Nortel's DMS-100 switching systems. He was lead designer/architect for automatic selection of line card-based balance networks to improve Echo Return Loss enhancing the voice quality of telephone calls. He was lead designer of a multi-channel DSP platform for line/trunk testing and traffic monitoring for the DMS switches. While working for Nortel he taught DSP and Telecommunication Protocols at the University of Toronto. In 1999 he joined California-based Nortel/Micom to develop DSP algorithms for echo cancellation/voice enhancement of access gateways. In 2001 he joined Motorola's Echo Cancellation Group at Austin, Texas as a DSP architect.

Dr. Dyba is an author/co-author of over 80 papers, book chapters, standard contributions, and other technical contributions. He has 7 patents/patents pending in the areas of motionless wave generation, general DSP, and echo cancellation. He is a member of the Acoustical Society of America, IEEE, and Association of Professional Engineers of Ontario.
Email: roman@motorola.com, Tel: (512)996-4973

Lúcio F. C. Pessoa was awarded his Ph.D. degree in Electrical Engineering at the Georgia Institute of Technology in 1997. In 1998 he joined Motorola SPS as a DSP algorithm designer, where he made significant contributions during the final development stage of an ADSL transceiver. Since 2000 he has been focused on the development of various DSP components for packet telephony products, including network echo cancellation, voice quality enhancement

devices and generic tone generation/detection. Dr. Pessoa has 1 US patent issued, 6 patents pending, and over 30 publications on journals, conferences, book chapters and standard contributions. He is member of the Motorola SPS Systems/Software Patent Committee and a Senior Member of the IEEE.
Email: lucio.pessoa@motorola.com, Tel: (512)996-4930