Echo Cancellation
13/12/01

Level / Version : 001
Date : 12/12/2001
Reference / Référence : WM CUS GEN CTI 005

Level/ Indice	Date/ Date	History of the evolution / Historique des évolutions	Writer / Rédacteur
1.0	31/10/2001	**First issue**	Eric LOGEZ

	Name / Nom	Function / Fonction	Date/ Date	Signature/ Signature
Written by / Rédigé par	E.LOGEZ	Technical Support	12/12/2001	**APPROVED**
Checked by / Vérifié par	L.PARVY	Applications Engineering Manager	12/12/2001	**APPROVED**
Approved by / Approuvé par	F.BIETRIX	Signal Group Leader	12/12/2001	**APPROVED**

1- Echo cancellation

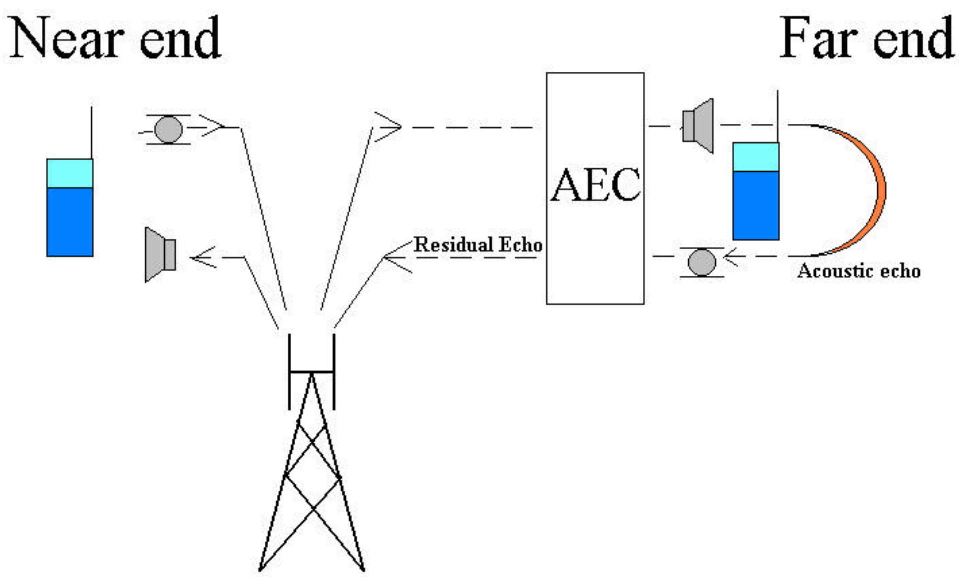
1.1 Introduction

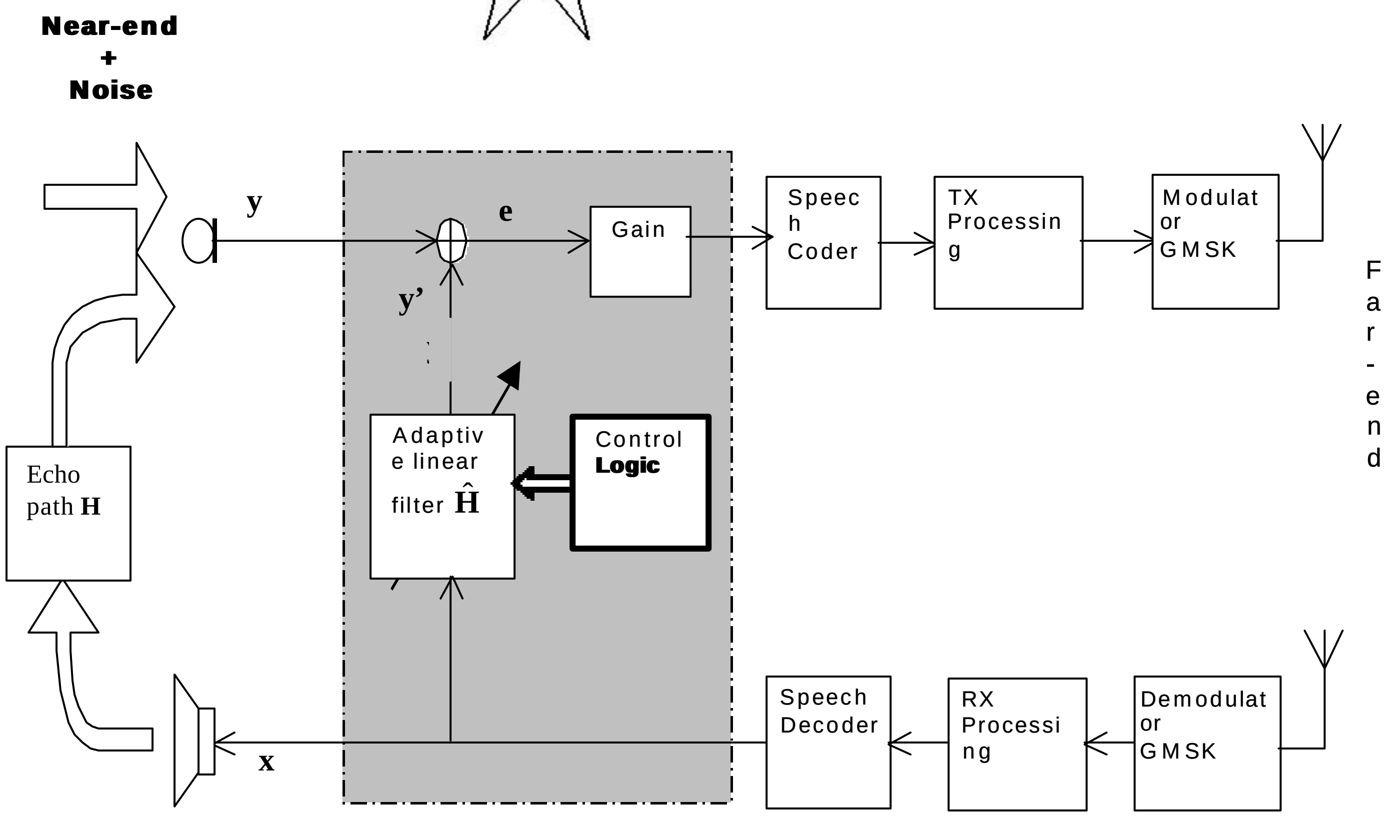
Subscribers use speech quality as the benchmark for assessing the overall quality of a network. The effective removal of hybrid and acoustic echo inherent within the digital cellular infrastructure is the key to maintaining and improving perceived voice quality on a call.

Acoustic echo is generated with analog and digital handsets, with the degree of echo related to the type and quality of equipment used. The form of echo is produced by poor voice coupling between the earpiece and microphone in handsets and hand-free devices. Further voice degradation is caused as voice-compressing encoding/decoding devices (vocoders) process the voice paths within the handsets and in wireless properties. When compounded with inherent digital transmission delays, call quality is greatly diminished for the wireline caller.
In a typical mobile situations, such as people are driving their cars, sound from a loudspeaker is heard by a listener, as intended. However, this same sound also is picked up by the microphone, both directly and indirectly, after bouncing off the roof, windows and seats of the car. The result of this reflection is the creation of multipath echo and multiple harmonics of echo, which, unless eliminated, are transmitted back to the distant end and are heard by the talker as echo.

1.2 Wavecom approach

In hands-free mode with a telephone or in a car, acoustic echo cancellation can be used as a means of suppressing acoustic feedback and echoes or increasing the received speech level by simulating the loudspeaker-room-microphone (LRM) system as an FIR filter (Finite impulse response) and by compensating it subsequently.
This filter makes a replica , or estimate, of the echo path. Passing the input signal through the filter generates an estimated echo signal that it subtracted from the received signal, in order to reduce the echo.
The more accurately the echo canceller identifies and simulates the LRM system, the more successful the compensation will be.

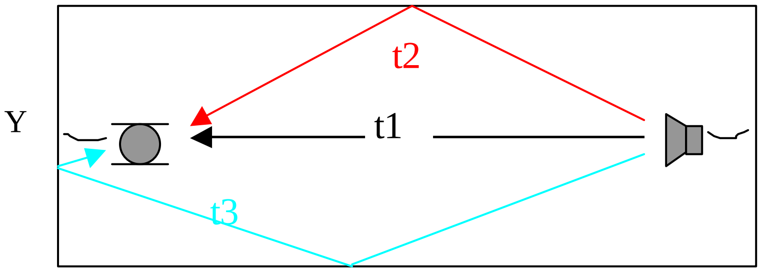




x : sample sent to PCM_OUT
y : sample received from PCM_IN
y': replica of the echo obtained via the linear filter
e : residual echo
H: impulse response of the room
H':Estimated impulse response

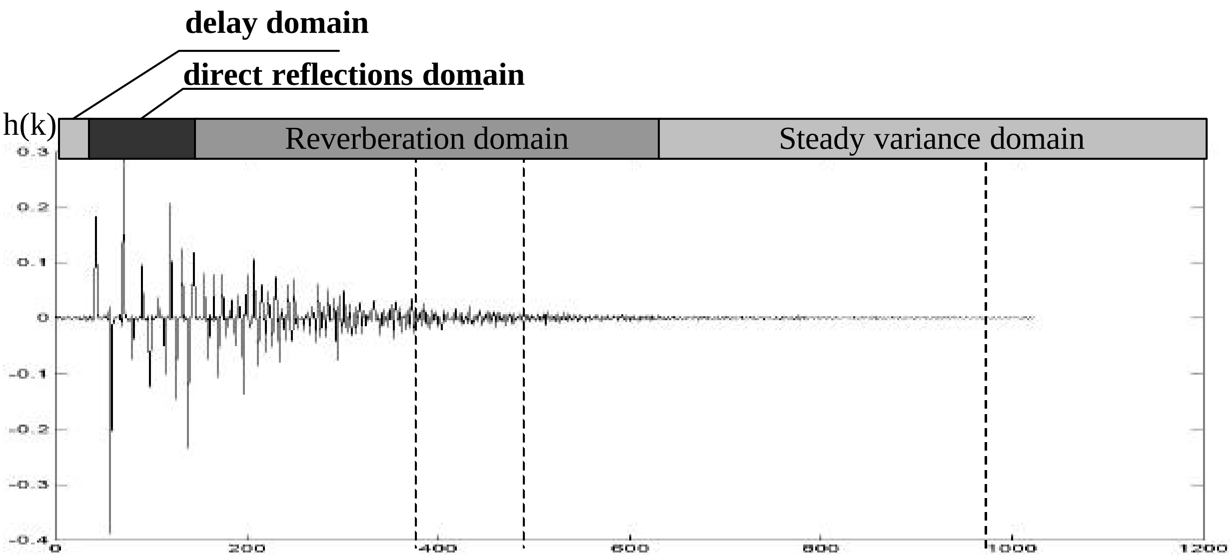
1.3 Acoustic reflections

Room Impulse Response



From the loudspeaker to the microphone : an infinity of acoustic reflections (t1, t2, t3, ...) because of the sides, the obstacles....
The loudspeaker-room-microphone (LRM) system can be simulated as a filter:

$$y(t) = \sum_i h_i x_{t-i} \quad : \quad \text{linear dependance}$$



- In a long acoustic reflection, the level of the reflection is low but the signal is very distorted. So you notice a very damaged signal, the quality of the speech is affected.

- In a short acoustic reflection, the level of the reflection is high but the signal is near the same as the direct signal. So you can't hear a distorted signal.

Adaptive linear filter
Most of today's echo cancellers use an adaptive linear filter with continuous updating of the filter coefficients. The speech from the far-end speaker is used in place of the test signal, and the filter coefficients are updated in response to the correlation between the echo and the speech. But this solution is very complexity.
Because of the complexity constraints on the echo canceller, the most widely used algorithm for adjusting the filter coefficients is the normalised least mean squares (NLMS) method. The NLMS method consist of adjusting the filter coefficients in order to minimise the power of the error between the true and estimated echo.
The algorithm is: H(n+1)=H(n) + fct[(y-y')+coeff]
This algorithm, so the update of the filter, is only used for a normally talking one at a time.

Single-talk
In case of single talk, the filter coefficients is adjusted.

Double-talk
For situations when the parties are talking at the same time, referred to as **double-talk** situations. The adaptation of the filter should then be inhibited, otherwise an erroneous estimate of the echo path is obtained, which results in poor echo cancellation.
The control logic function that inhibits adaptation in double-talk situation has to allow the echo canceller to converge.

Supplementary filter: switch attenuation
The main problem of echo cancellation is the adaptative of the filter for two situations, which demand different actions. So the adaptative filter is not exactly equal to the real filter. The output of the filter will not be an exact replica of the echo.
It's why after the echo cancellation, an attenuator is added to reduce the residual noise.

1.4 AT command:

AT+ECHO = <*mode*> , <*AlgoId*> , <*AlgoParam*> , <*NoiseThres*> , <*NmbTaps*>

With the AT command, it's possible to configure the echo and switch attenuation parameter with the third algorithm.

<mode>
0: Deactivate Echo
1: Activate Echo

<AlgoId>
3: Echo cancellation 3

<AlgoParam> switch attenuation parameter

this parameter allows to reduce the residual noise after the echo cancellation. So you modify the gain of the block just before the speech coder (see block diagram). Thus the signal is multiply by a coefficient between 0 and 1
The allowed range is [0 ; 63]. (**30** default)
0 means low attenuation of residual echo, so you don't modify the signal
63 means high attenuation of residual echo.

<NoiseThres> : noise reduction parameter.

<NmbTaps> : **echo cancellation parameter**
indicate the Number of Taps of the Adaptive Filter H()
The allowed range is [64 ;256]. (**256** default)
In all the case, we choose 256 but usually, to optimise the charge of the CPU, it's better to adapt the value depend on the environment:
- 64 taps is for short Echo, for example for a handset.
- 256 taps is for long Echo: for example in a car.

2. Explanation of the Noise reduction

2.1 Introduction

The noise reduction algorithm was designed to be implemented directly on the base band chip. So it only need a new software version and the most important thing is: it doesn't need an external DSP.
The choice of the algorithm had to satisfy some drastic criterions:
MIPS consumption,
memory space,
acceptable quality
The algorithm is based on spectral amplitude estimation using an overlap-add FFT filter bank system.

Description of the algorithm
The basis of the system is the well-known *filter bank overlap addition method* with an FFT length of N=256. The optimal weighting of spectral magnitudes is computed using a noise power estimate and a subtraction rule.

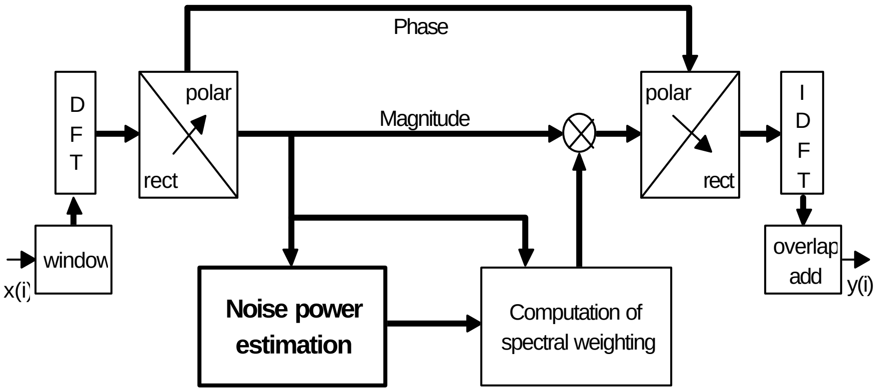


Figure 1: Block diagram of spectral processing

Windowing
The use of a smooth data-tapering window is used to reduce edge effects in the data and to obtain superior spectral properties. The choice is a 'modified' Hann window (also known as the Hanning window), it's very efficient implementation in the frequency domain.

DFT: Discret Fourier Transformer
To modify every sample of the speech signal, we need to transform the signal to the frequency domain. After the DFT, it's necessary to convert every sample in polar coordinate. In this way, the speech signal is separate between the spectral magnitudes and the phases. We only modify the magnitude because it's easier and need less MIPS consumption. We call this spectral amplitude: $|X_k(m)|$.

Transformation of axis
We have 160 samples of the speech signal every frame so every 20ms. The discret spectral magnitude of the audio signal is sampling between –4KHz to +4KHz.

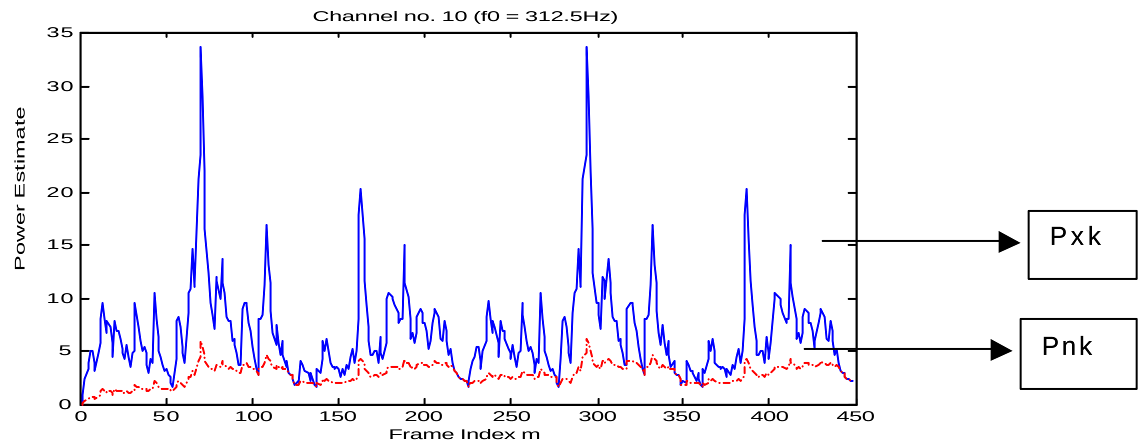
So we transform this trace to another axis. We trace in a graph the power signal of every frame for each frequency. We obtain the Pxk (see below).
We do this operation every 31,25Hz (=8kHz/256) between 0Hz and 4KHz.

Short-time noise power $P_{nk}(m)$ estimation
When we have the power signal (Pxk) for each frequency, we can estimate the noise spectral power (Pnk) by a non-linear estimator :

If $P_{nk}(m-1) < P_{xk}(m)$
then Pnk(m) = fct(the actual sample, the slope, and the previous sample).

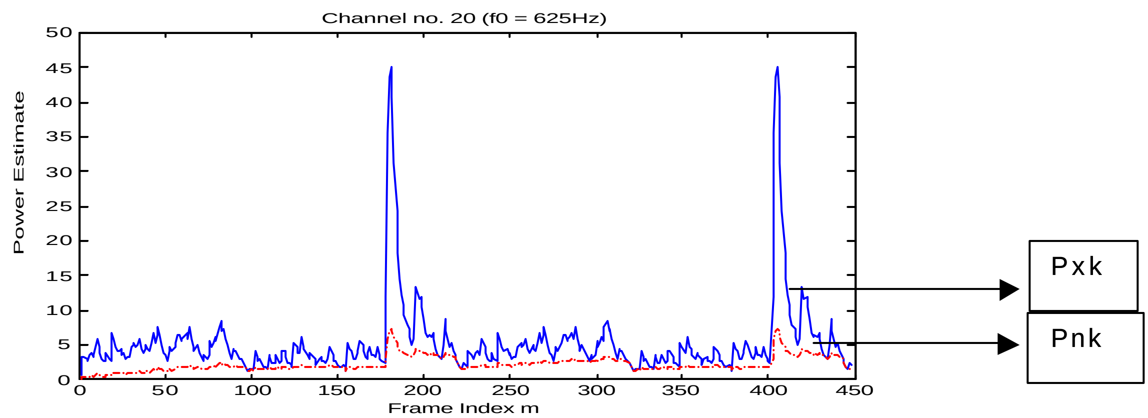
Else $P_{nk}(m) = P_{xk}(m)$

here is a example of the noise spectrum estimation.



(a)

(b)
Examples of spectral power estimation using Eq.1-3 (solid lines for Px, dashed lines for Pn) for a noisy speech signal with SNR=0dB, plotted at different subbands with center frequencies f0=312.5Hz (a) and f0=625Hz (b).



Spectral subtraction rule

When we know an estimation of the noise power, we can calculate a gain factore $g_k(m)$ to modify the spectral amplitudes $|X_k(m)|$.
A spectral subtraction rule is applied :

$$g_k(m)=\begin{cases}1-fct(P_{nk}(m),k) & g_k(m)\geq \beta_f \\ \beta_f & otherwise\end{cases} \quad (4)$$

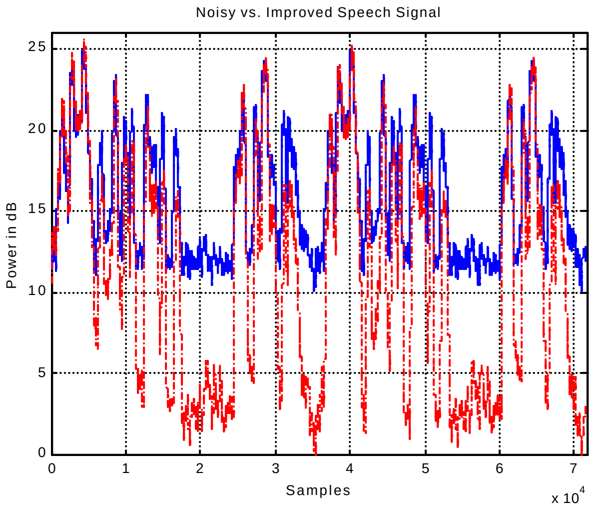
β_f is a so-called spectral-floor. It limits the attenuation of the noise reduction filter to a positive value to leave some noise in the output signal. If the noise were reduced completely, the quality of the output signal would not be acceptable, even though the absolute SNR enhancement of the noise reduction would increase.

Then, we multiply the spectral amplitudes $|X_k(m)|$ by the corresponding gain factors $g_k(m)$. It gives the improved signal subbands $|Y_k(m)|$:

$$|Y_k(m)| = g_k(m).|X_k(m)| \quad k = 1...256 \quad (5)$$

Experimental Results

For simulation purposes, the input signal of the spectral subtraction was created by adding a car noise recorded in a real car on the road. The level of noise was set in order to have the SNR=5dB. The level of noise subtraction is adjustable with the β_f parameter.



The SNR of the noisy speech signal is about 5 dB (solid lines for noisy signal, dashed lines for improved signal). The improvement in the SNR in around 10

2.2 Conclusion

We have presented an efficient enhancement algorithm. The principal advantage of this noise reduction system is the easily integration in the internal DSP of the module, so you don't need an external DSP for the audio.
The noise estimation performs spectral minimal tracking in subbands, and thus allows for suppression of short-time stationary noise disturbances.

2.3 AT command

AT+ECHO = <*mode*> , <*AlgoId*> , <*AlgoParam*> , <*NoiseThres*> , <*NmbTaps*>

With the AT command, it's possible to configure the noise reduction parameter with the third algorithm.

Defined values:

<mode>
0: Deactivate Echo
1: Activate Echo

<AlgoId>
3: Echo cancellation 3

<AlgoParam> switch cancellation parameter

<NoiseThres> noise reduction parameter
indicate the noise threshold.
Low value leads to high noise attenuation (but can modify little bit the audio signal).
High value means no noise attenuation (so no modification of the audio signal).
The allowed range is [0 ;32767]. (**8000** default)

<NmbTaps> echo cancellation parameter

the NoiseThres correspond at β_f = NoiseThres / 32767.

Example:

To activate the AEC and NR:
AT+ECHO=1,3,63,8000,256

To deactivate the AEC and NR :
AT+ECHO=0