DSP-BASED SOLUTION FOR AMBIENT NOISE REDUCTION IN MOBILE PHONES

Jamil CHAOUI, Sébastien de GREGORIO, Guillaume GALLISSIAN, Yves MASSE
Texas Instruments France - Wireless Business Unit
BP5 - 06271 Villeneuve Loubet Cedex FRANCE
j-chaoui1@ti.com, s-de-gregorio@ti.com, g-gallissian@ti.com, y-masse@ti.com

ABSTRACT

Using a mobile handset in noisy environments makes the far-end speech difficult to be perceived from the near-end user perspective. As he is surrounded by a loud background noise, handset user needs to be highly concentrated to understand the far-end speaker conversation. Ambient noise reduction algorithms provide an efficient solution to build a silent zone between the handset loudspeaker and the user ear, thus improving speech understanding for the near-end speaker.

This paper presents a full description of an innovative ambient noise reduction system developed on a TI TMS320C54x DSP. This contribution is combining both theoretical and experimental considerations, raising potential issues that may be encountered when implementing these applications on a real system. It will be shown that the advanced architecture of TI TMS320C54x DSP makes the ambient noise reduction application possible to be executed in the same CPU performing in the same time wireless digital cellular baseband processing.

1. INTRODUCTION

When communicating with a mobile handset in noisy environments such as running car, railway stations or airports, handset user needs to be highly concentrated to understand the far-end speaker conversation. Active noise cancellation algorithms may provide an efficient solution to reduce the ambient noise surrounding the mobile user, by building a small silent zone around the handset speaker, thus improving speech understanding.

In Texas Instruments Wireless Business Unit, an ambient noise reduction system has been developed and implemented on TI TMS320C54x DSP [4]. This DSP family, especially suited for wireless applications, is widely used by major mobile manufacturers.

In this paper, we will first present an overview of the complete system. We will then detail the F-xNLMS algorithm formulation and its similarity to the well-known NLMS algorithm. It will be demonstrated that the formulation of the F-xNLMS algorithm leads to the inversion of 2 filters, which can be done only under a strong quasi-stationary hypothesis. We will then focus on the C filter description for a real GSM handset. The implementation of the algorithm on TI TMS320C54x will also be described. Thanks to the DSP architecture, a low-resource implementation could be achieved. Finally, we will present the performances of the complete system in a real application, demonstrating the real benefit that could be seen from the final user with such an algorithm.

2. SYSTEM DESCRIPTION

The proposed system is described in **Figure 1**.

An adaptive algorithm that will be further described in part 3, running on a TI TMS320C54x DSP generates a canceling noise. This signal is then artificially inverse-phase shifted and added to the far-end speaker's voice, before being played on the speaker. Therefore, the ambient noise will be reduced in a small area around the speaker, thus improving speech understanding

The adaptive algorithm inherently handles the sample misalignment due to A/D-D/A conversion delays.

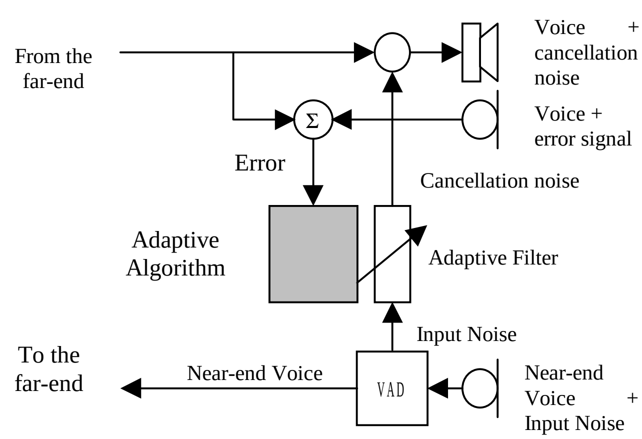

Figure 1
Ambient Noise reduction System Overview

The handset microphone collects the near-end speech embedded in the background noise. A voice activity detection algorithm is then performing the distinction between the near-end speaker voice and the environmental noise. This noise is then filtered to generate the canceling noise that will be played on the handset speaker.

An additional microphone located inside the speaker captures the error signal that is fed into the adaptive filter algorithm to update the filter taps. It should be noticed that this additional microphone must be installed with special care to avoid any mechanical coupling with the handset speaker. This mechanical coupling may cause non-linear effects that may deteriorate the adaptive algorithm convergence, and may even lead to a divergence of the filter adaptation.

3. ALGORITHM DESCRIPTION

Filter taps are adapted with the Filtered-x NLMS algorithm [1].

The Filtered-x NLMS algorithm (see **Figure 2**) derives from the NLMS algorithm, with the introduction into the adaptation process of a filter generally called C in the literature [1], [2], [3], modeling the physical characteristics of the acoustic propagation into the cavity formed by the handset and the user ear. This filter also handles the delay due to the D/A converter located before the handset speaker. It also models the loudspeaker physical characteristics and the propagation of the background noise through the handset body.

The determination of *C* filter will be described in part 4.

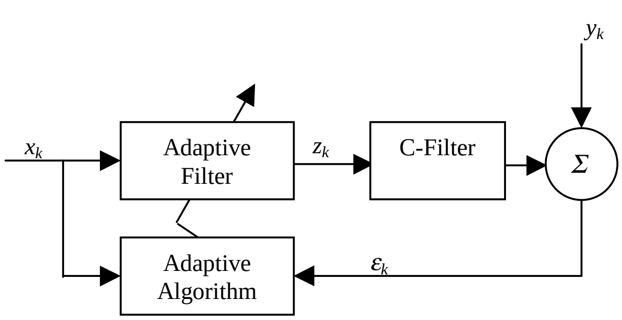

Figure 2
F-xNLMS block diagram

The notations are as follows:
L is the length of the adaptive filter,
X_k is the length-L vector of the L last input samples at time k,
Z_k is the length-L vector of the estimated cancellation noise at time k,
C is the C filter taps,
H_k is the length-L adaptive filter coefficient vector at time k,
y_k is the sample of the noise to be cancelled at time k
ε_k is the error signal at time k,
R_k is the length-L vector of the L last filtered reference samples at time k,
α is the adaptation step.

The error considered in the F-xNLMS algorithm is the sum of the external disturbance and the output of the adaptive filter H, filtered by the C filter [1]. From **Figure 2**, it is easy to see that Z_k could be expressed as (where * denotes convolution):

$$Z_k = H_k * X_k$$

Therefore, the error ε_k could be expressed as:

$$\varepsilon_k = y_k + C * Z_k = y_k + C * H_k * X_k$$

When introducing the filtered reference signal defined as:

$$R_k = C * X_k$$

it is easy to derive that ε_k could be computed as follows:

$$\varepsilon_k = y_k + H_k * R_k$$

This expression appears to be close to the NLMS algorithm [2]. The filtered reference signal R_k is indeed replacing the original signal x_k in the NLMS formulation. The subtraction appearing in the NLMS equation is transformed into an addition to model the two acoustic waves combination into the air.

Therefore, it is now easy to derive the F-xNLMS complete equations from the NLMS equations:

$$\begin{cases} R_k = C^T X_k \\ \varepsilon_k = y_k + H_{k-1}^T R_k \\ H_k = H_{k-1} - \alpha\, \varepsilon_k R_k \end{cases}$$

The equivalent scheme for this new formulation of the F-xNLMS algorithm is represented on **Figure 3**.

It must be pointed out that the input signal is now filtered by the C filter <u>*before*</u> being filtered by H. As mentioned in [1], this filter inversion can only be done under the assumption that these 2 filters show quasi static variations, and therefore vary slowly with time. For our specific application, this assumption may be challenged, as the user may move quickly the handset around his ear. However, real-life tests have shown the validity of the filter quasi-stationarity hypothesis.

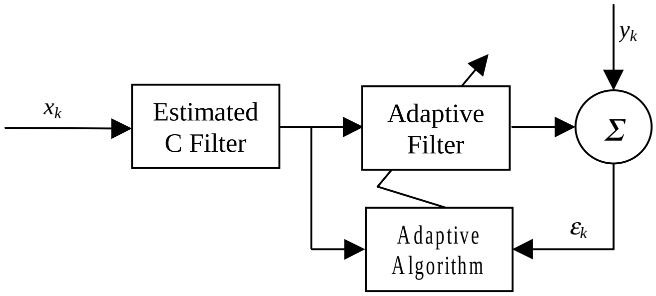

Figure 3
F-xNLMS equivalent block diagram

4. C FILTER DETERMINATION

The experimental set-up described by **Figure 4** was used to determine the *C* filter, with a commercial GSM handset.

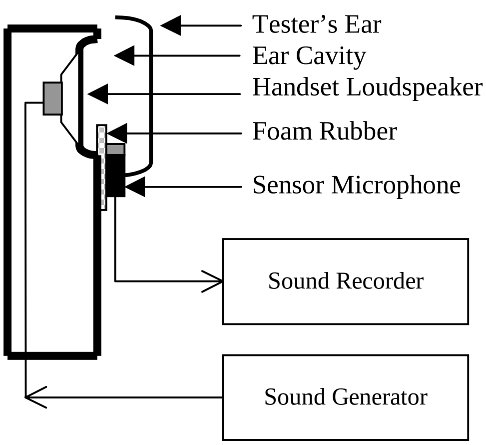

Figure 4
C Filter Measurement Set-up

All signals were sampled at 16kHz. Pure sine waves were generated in order to measure the *C* filter impulse response. **Figure 5** shows the *C* filter impulse response spectrum, while the time domain impulse response is represented on **Figure 6**.

Our measurements have shown the presence of strong non-linearity, mainly due to the mechanical vibrations on the handset body. In order not to overload the computational burden of the F-xNLMS implementation, this non-linearity was neglected.

The *C* filter spectrum shape shows a strong minimum around 1kHz. This is very likely to be due to a destructive resonance of the stationary acoustic waves into the cavity at this particular frequency.

Two local maximums are appearing on the time-domain representation (**Figure 6**), revealing that two acoustic paths are present into the cavity. The first path (direct path) has a length less than 2 cm while the secund path length (complex multipath) is about 10 cm.

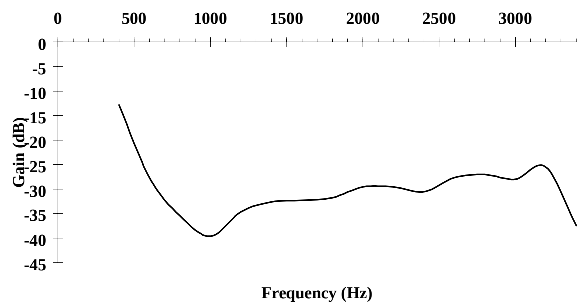

Figure 5
C Filter Impulse Response Spectrum

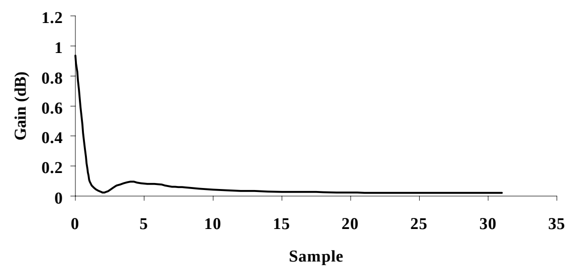

Figure 6
C Filter Impulse Response

5. IMPLEMENTATION ON TI C54x

The algorithm has been implemented on a TI TMS320C54x fixed-point DSP[4]. This DSP combines high-performance (up to 100MIPS at 2.5V CPU core voltage), very low power consumption (0.6 mA/MIPS at 2.5V), a large degree of parallelism, and a specialized instruction set aimed at efficiently implementing a variety of complex algorithms and wireless applications.

This DSP family is based on an advanced Harvard architecture, built around 4 major internal buses: 1 program bus, 2 read data buses and 1 write bus. Therefore, 2 read and 1 write operation can be performed in a single cycle. Some instructions allow to perform memory store or memory loads in parallel with arithmetic computation.

Therefore, advanced signal applications requiring intensive computations, such as active noise cancellation algorithms, can be implemented in a very efficient way, by taking fully advantage of the powerful architecture of these DSPs.

TI TMS320C54x DSP provides a dedicated LMS instruction that allows the computation in only 2L machine cycles of the new filter coefficients and the error at a given time. However, this instruction leads to an adaptation algorithm close to the well-known Delayed-NLMS algorithm, which we have experienced to show high instability in our active noise cancellation application.

The complete ambient noise reduction application is performed in only 2500 cycles. As the signal is sampled at 8KHz, this represents a CPU loading of only 20 MIPS (20% of the total CPU). Data RAM requirement is less than 5K*16 bits. Therefore, these low resource requirements make the ambient noise reduction application possible to be executed on the same DSP performing also wireless digital cellular baseband processing.

6. PERFORMANCE EVALUATION

The ANR (Ambient Noise Reduction) criterion was defined to evaluate the performances of the system. This criterion, similar to the well-known ERLE (Echo Return Loss Enhancement) criterion commonly used in echo cancellation applications, is defined hereafter, using the same notations as in part 3.

$$ANR = 10.\log\left(\frac{E\left\{|y|^2\right\}}{E\left\{|\varepsilon|^2\right\}}\right)$$

On **Figure 7**, the ANR is plotted versus the adaptive filter size, showing that, with a filter length of only 256 samples, an ambient noise reduction greater than 15dB can be expected.

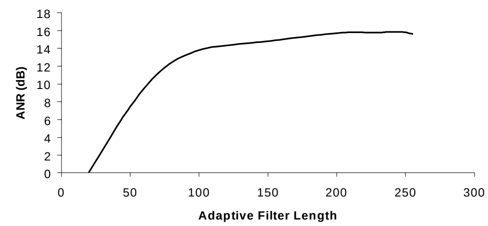

Figure 7
System performances.

The influence of the number of taps in the *C* filter was studied through the limitation of the convolution sum length in R_k calculation, which is equivalent to reset the last taps of the C filter. It was observed that by reducing to C filter to its 2 first coefficients, 97% of the optimal performance can be reached. This last result shows that the secondary acoustic propagation path could be neglected compared to the primary direct path.

7. CONCLUSION

In this paper, we have described a complete DSP-based system for ambient noise reduction in mobile phones. The high performances of the algorithm coupled with an efficient implementation showing very limited resource requirements, make this application possible to be integrated in a wireless baseband platform. We have also shown the very little influence of the C filter on the final performances of the system.

8. REFERENCES

[1] Stephen J. Eliott, P. A. Nelson. "Active Noise Control," *IEEE Signal Processing Magazine*, October 1993, pp 12-33

[2] Hareo Hamada, "Signal Processing for Active Control", *International Symphosium on Avtive Control of Sound and Vibration*, April 9-11, 1991, Tokyo, pp 33-44.

[3] B. Widrow and S.D. Stearns, "Adaptive Signal Processing," Prentice-hall, Englewood-cliffs,1985.

[4] "TMS320C54x User's Guide" - 1996