TEXAS INSTRUMENTS

Acoustic-Echo Cancellation Software for Hands-Free Wireless Systems

Application Report

1997 Digital Signal Processing Solutions

Printed in U.S.A., July 1997

SPRA162

Acoustic-Echo Cancellation Software for Hands-Free Wireless Systems

SPRA162
July 1997

TEXAS INSTRUMENTS

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

Contents

List of Figures

Acoustic-Echo Cancellation Software for Hands-Free Wireless Systems

ABSTRACT

This note describes the DSP implementation of a full-duplex acoustic-echo cancellation (AEC) software. This software is based on the normalized least mean square (NLMS) algorithm. The algorithm includes active-channel and double-talk detection. This software was implemented on a TMS320C54x digital signal processor (DSP) in assembly language. It requires few resources in terms of million instructions per second (MIPS), random-access memory (RAM), and read-only memory (ROM). AEC achieves 30 dB of attenuation in an automobile environment. The NLMS algorithm cancels more or less echo depending on the noise level, and responds well to variations of the signal level.

1 Introduction

New safety regulations are leading the field of radio-communications towards hands-free radio/telephones. With such a system, the speaker (operator) can talk freely and still concentrate on his driving task. However, one of the drawbacks of this system is that acoustic echo is perceived by the far-end speaker. The European Telecommunication Standards Institute (ETSI) is currently specifying AEC systems.

The echo phenomenon is caused by the coupling between the loudspeaker and the microphone (see Figure 1). In full-duplex communications, the far-end speaker hears his own voice with a delay (echo) depending on the automobile interior and the global system for mobile communications (GSM) delay. The length of the echo path is a key parameter for the AEC. The software described in this application note focuses on the automobile hands-free environment.

Adaptive filtering (and more precisely, the NLMS algorithm) is one of the most common solutions to the problems of AEC. The NLMS algorithm offers a good tradeoff between computational load and performances.

Other problems with AEC are double-talk (DT) conditions, where both operators are speaking at the same time. If not detected, DT can cause divergence of the adaptive algorithm.

The AEC software utilizes the NLMS algorithm to cancel the echo, and is implemented in ’C54x DSP assembler. The software is described in Section 2, while its test results are shown in Section 3.

2 AEC Algorithm

This section contains six detailed instances of algorithm discovery from the system description up to, and including, the benchmarks that describe the firmware, time, and cost elements.

Discoveries include the following descriptions:

- System description
- NLMS algorithm matrix
- Active-channel detection
- Double-talk detection
- Software architecture
- Benchmarks

2.1 System Description

Figure 1 shows an overview of the AEC system. The system includes the AEC software implemented on a TMS320C54x DSP, a loudspeaker, and a microphone.

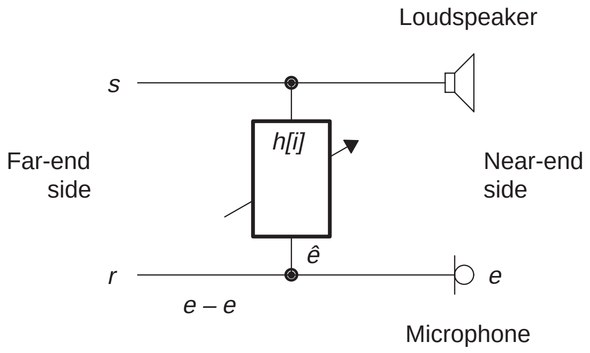

Figure 1. Overview of an AEC System

2.2 NLMS Algorithm

The NLMS algorithm updates the coefficients of an adaptive finite-impulse response (FIR) filter. This filter is used to predict the echo. This prediction is subtracted later from the real echo and gives a residual echo.

The length of the filter matches the length of the echo path (typical value is 32 ms for an average car). For a sampling frequency of 8 kHz, the number of taps (filter coefficients) is, therefore, equal to 256.

- s is the signal matrix.
- e is the echo matrix.
- h is the filtering coefficient matrix.

The result of the adaptive filtering is given by the formula:

$$\hat{e}[n] = \sum_{i=0}^{255} h\ [i].\ s[n-i] \tag{1}$$

The residual error ***r*** is given by the formula:

$$r[n] = e[n] - \hat{e}[n] = e[n] - \sum_{i=0}^{255} h\ [i].\ s[n-i] \tag{2}$$

The NLMS coefficients at the nth step of the adaptation are updated with the formula:

$$h[j] = h[j] + \frac{\mu.r[n].\ s[n-j]}{\sigma_n} \tag{3}$$

where μ is the adaptation step and σ_n is the signal energy.

$$\sigma_n = \sum_{k=0}^{255} s^2[n-k] \tag{4}$$

Formulas (2) and (3) are computed in the same cycle with the LMS instruction provided by the C54x assembler. But it will delay the NLMS algorithm because the *h[j]* coefficients used to compute the residual ***r*** are those computed for the previous sample. Example 1 shows the main loop of the source code.

Example 1. Main Loop of the NLMS Algorithm Source Code

The main loop computes the echo model, updates the algorithm, and stores the filter coefficients in two cycles (instead of three cycles, without the LMS instruction).

AR3 is a pointer to the filter coefficients.

AR2 is a pointer to the input sample matrix.

AR4 is a pointer to the previous input sample matrix.

These three pointer registers are used in circular addressing.

AR0 = 1

T register is loaded with $\frac{\mu . r[n-1]}{\sigma_n}$

```
LMS    *AR3, *AR2+0%
          ; A = T*s[n–1–i] + h[i] ; B = B + h[i]*s[n–i]
ST     A, *AR3+0% || MPY *AR4+0%, A
          ; h[i] = T*s[n–1–i] + h[i] ; A =T*s[n–2–i]
```

2.3 Active-Channel Detection

A key feature of an AEC algorithm is active-channel detection. Indeed, when the far-end operator is silent and the near-end operator is talking, the filter must not be adapted because the near-end operator is no longer an echo.

Active-channel detection is achieved by computing the energy of the signal, and then comparing this energy to an adaptive threshold. Using SQUR and 32-bit instructions, the energy is computed as shown in Example 2.

Example 2. Source Code Energy Computing

The smoothing coefficient value is $2^{–6}$.

A is loaded with the last input sample s[n].

```
SQUR      A, B            ; B = s[n]^2
DSUB      @PWR_SIG, B     ; B = s[n]^2 – Energy
SFTA      B, –6           ; B = (s[n]^2 – Energy)*2^–6
DADD      @PWR_SIG, B     ; B = (1–2^–6)Energy + 2^–6 s[n]^2
DST       B, @PWR_SIG     ; Store signal Energy
```

When the receive channel is inactive, the AEC software is no longer processing, which prevents the adaptation of the filter during periods of silence; in this case, *r = e.*

When the send channel is inactive (silent) and the echo is too low, the filter adaptation ceases in order to reduce the computation; again, in this case, *r* = *e*.

2.4 Double-Talk (DT) Detection

During a DT condition, the near-end signal on the microphone contains echo (*e*) and near-end speech (that is, double-talk)***.*** The residual error used to update the filter coefficients includes the near-end speech, and if the algorithm is still adapting, the algorithm may start to diverge. This situation must be prevented.

DT detection uses an energy-based algorithm with a variable threshold.

During the DT periods, the software algorithm processes as follows:

- The filter coefficients are frozen (no adaptation).
- The far-end signal *s* is still filtered.
- The residual echo is *r = e – h*s* (* represents the convolution operator).

2.5 Software Architecture Description

Figure 2 shows the flow of the software, which is composed of the following three main procedures:

- *AECmain()* Computes channel-active and double-talk detection
- *NLMSloop()* Computes the NLMS algorithm
- *FilterLoop ()* Computes the algorithm described in DT detection

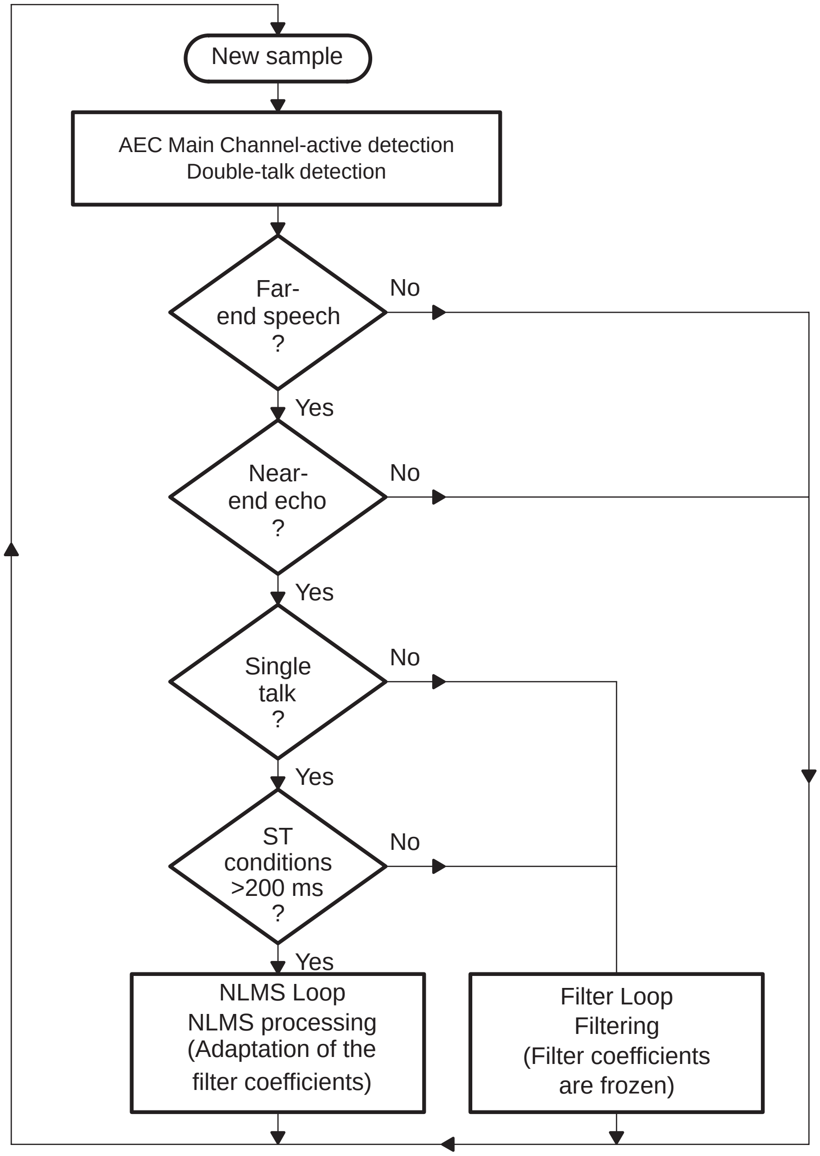

Figure 2. Architecture of the AEC Software

2.6 Benchmarks

The benchmarks (expressed in 16-bit words) of the software are:

- Code size: 154 words
- Static RAM: 527 words
- Scratch RAM: 2 words
- The maximum computational-time cost is 4.7 MIPS

The computational-time cost is maximum during ST periods; during DT periods, it is reduced to 2.4 MIPS. ST periods represent the main part of a conversation, whereas DT periods are present only during brief and limited instances.

3 Performance Results

This section contains six detailed instances of performance results from performance assessment up to and including DT-condition test results.

The instances include:

- Performance assessment
- Test results in a quiet environment
- Response to input-level-signal scaling
- Response to echo-level scaling
- Echo versus noise-test results
- Tests during DT conditions

3.1 Performance Assessment

Various test signals recorded in real automobile environments were used to assess the performance of the algorithm. This assessment is quite similar to the one recommended by the ETSI.

As specified by the ETSI, the measurements were taken after 10 seconds of speech to allow the algorithm to reach a steady state, then the terminal coupling loss (TCL) was computed as follows:

$$TCL = 10 \log \left(\frac{\sum_{n} s^2[n]}{\sum_{n} r^2 \, [n]} \right) \tag{5}$$

The TCL computes the total attenuation between the receiving port and the sending port of the far-end side.

3.2 Test Results in a Quiet Environment

The TCL was tested in actual automobile environments on different signals (male voice, female voice). The mean value of these tests is roughly:

$$TCL = 30 \text{ dB}$$

The initial convergence time is good. Indeed, the TCL nearly attains its maximum value after two seconds (TCL = 28.8 dB).

3.3 Response to Input Level Signal Scaling

The level of the far-end signal varies significantly due to the various network gains. The objective of this test is to reveal the response of the algorithm when the input-level signal varies.

Figure 3 shows that the performance of the algorithm is steady (TCL ≈ 30 dB) for an input-signal level between 10 bits and 15 bits (13 bits is the nominal value). Above 15 bits, the signal is saturated; therefore, the TCL measurement is meaningless. When the signal level is under 7 bits, the TCL reaches the coupling loss.

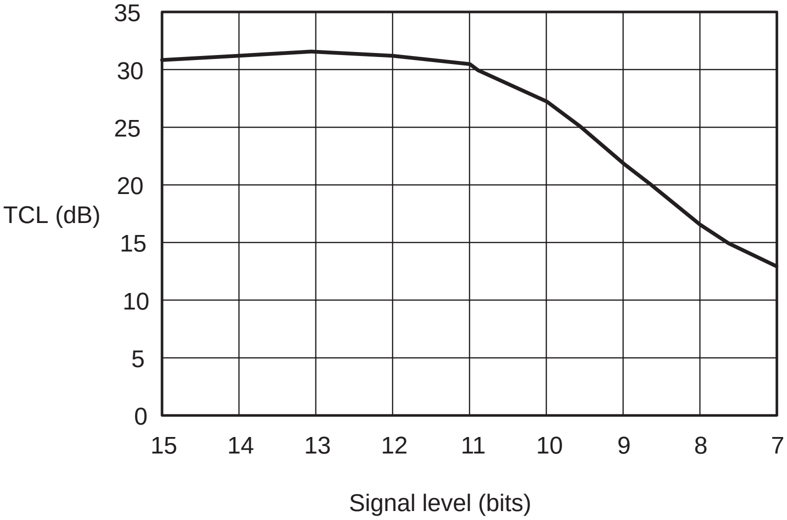

Figure 3. TCL / Input Signal Scaling

3.4 Response to Echo-Level Scaling

These tests evaluate the echo attenuation when the echo level is raised without raising the signal level (i.e., when using the volume control of the loudspeaker).

Nominal echo-level value is 11-bits. Figure 4 shows that the TCL is low for an echo level of 13 bits at a level of 13 dB, which indicates that the echo and the signal have the same level. This is unlikely to happen; indeed, if the microphone and the loudspeaker are properly located in the automobile, a minimum of 6 dB attenuation between the signal and the echo can be expected.

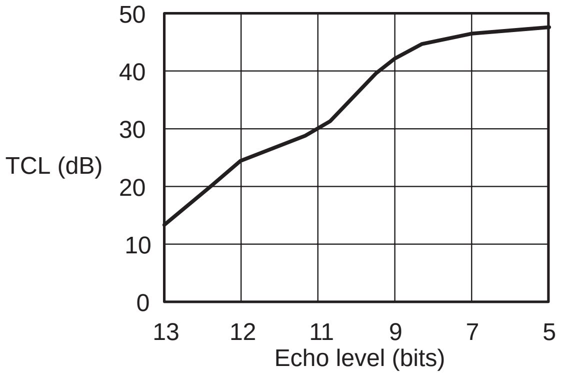

Figure 4. TCL / Echo Scaling

3.5 Echo Versus Noise Test Results

To test the behavior of the algorithm in noisy environments, artificial noise was added to offer a near-end signal of *e* + *noise.*

These tests were made using two new parameters:

- Echo-to-noise ratio (ENR)

$$ENR = 10 \log \left(\frac{\sum_n e^2 [n]}{\sum_n noise^2 \ [n]} \right) \tag{6}$$

- The TCL in the performance assessment section is no longer significant and is computed with this formula:

$$TCL = 10 \log \left(\frac{\sum_n (s{-}noise^2)[n]}{\sum_n (r{-}noise^2) \ [n]} \right) \tag{7}$$

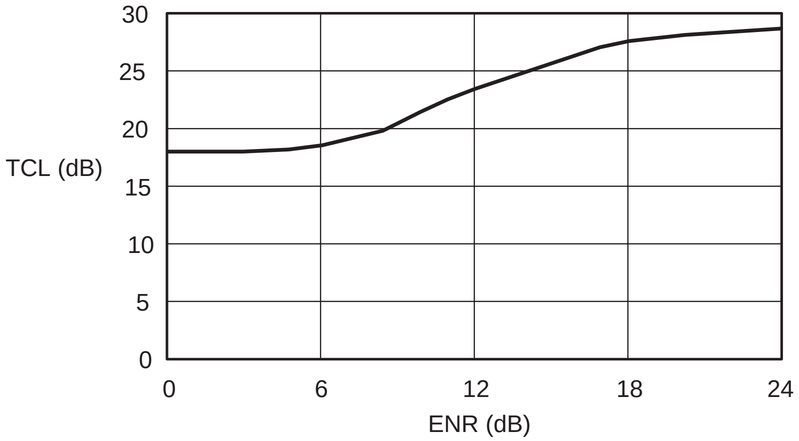

Figure 5. TCL in Noisy Conditions

When the ENR is high (about 24 dB), the TCL is the same as the TCL in noiseless conditions.

When the ENR decreases below 0, the TCL equals approximately the attenuation in the echo path. However, this situation is not annoying because the noise masks the echo.

3.6 Tests During Double-Talk Conditions

The file used for the test was composed of a single talk (ST) period of 14 seconds, followed by DT for 3 seconds, then the ST condition returned.

When DT occurred, the TCL computation was initialized and was computed with the formula:

$$TCL = 10 \log \left(\frac{\sum_{n} (s\text{–}dt)^2 [n]}{\sum_{n} (r\text{–}dt)^2 \; [n]} \right) \qquad (8)$$

Figure 6 shows that the filter is less and less adapted to the signal in the DT period as the TCL slowly decreases. Assuming that DT periods are short, the loss of efficiency of the algorithm is not critical during these periods.

An excellent recovery time is realized after double talk. Following DT, the TCL resumes its maximum value (30 dB) within two seconds.

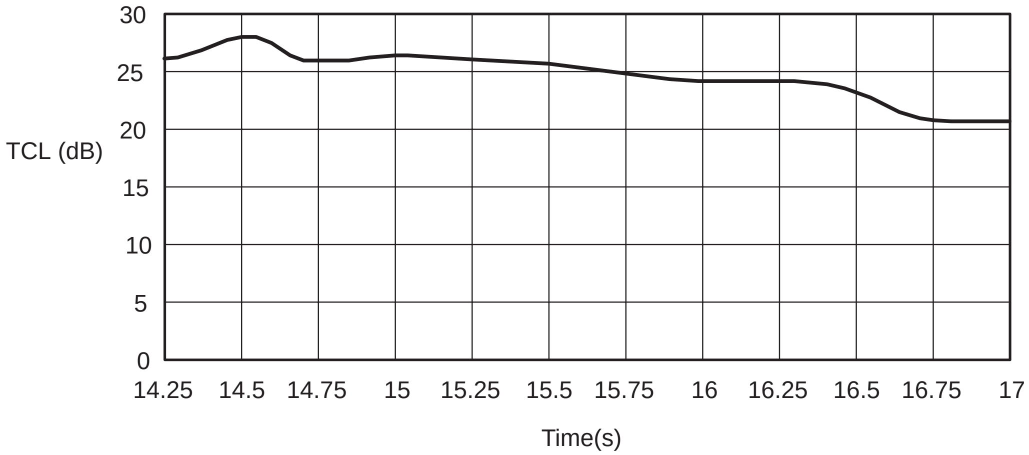

Figure 6. TCL in DT Conditions

4 Conclusion

The AEC algorithm achieved a low computational cost of about 4.7 MIPS and 30 dB of TCL in ST, while displaying acceptable control of DT periods and noisy conditions.

This software was tested with various “ETSI-like” tests. Even though these tests were not the ones specified by the ETSI, they are analogous to the ETSI specifications. Nevertheless, some improvements are required to duplicate the ETSI specifications [1] and the ITU-T recommendation [2].

References

1. “Test Methods and Quality Assessment for Hands-Free Mobile Stations” ETSI – GSM Technical report – Draft – DTR/SMG, September 1996.
2. “General Characteristics of International Telephone Connections and International Telephone Circuits”, “Acoustic Echo Controllers” ITU-T Recommendation G.167, March 1993.