SYSC5603 Project Report:

Real-Time Acoustic Echo Cancellation

Author: Trevor Burton

Submitted to: Professor Miodrag Bolic

Carleton Student ID: 100278854

Date: Monday, April 3, 2006

1. Introduction

In any hands-free communication system an acoustic echo canceller (AEC) is required to ensure that a high quality conversation exists between parties. The AEC accomplishes this by preventing the near end speech of a local talker that is reproduced at the far end from being transmitted back to the near end environment. In a hands-free system the near end speech that is broadcast at the far end via a loudspeaker is also captured by the far end microphone directly and due to inherent acoustical room reflections. This results in an undesired echo signal that the AEC must remove while allowing the local talker signal to be transmitted to the other end of the hands-free communication system. This problem occurs in an identical manner in the near end environment. The reader is directed to [1-4] for a more detailed discussion of the acoustic echo cancellation problem. One half of a typical single microphone hands-free communication system is shown in Figure 1 below.

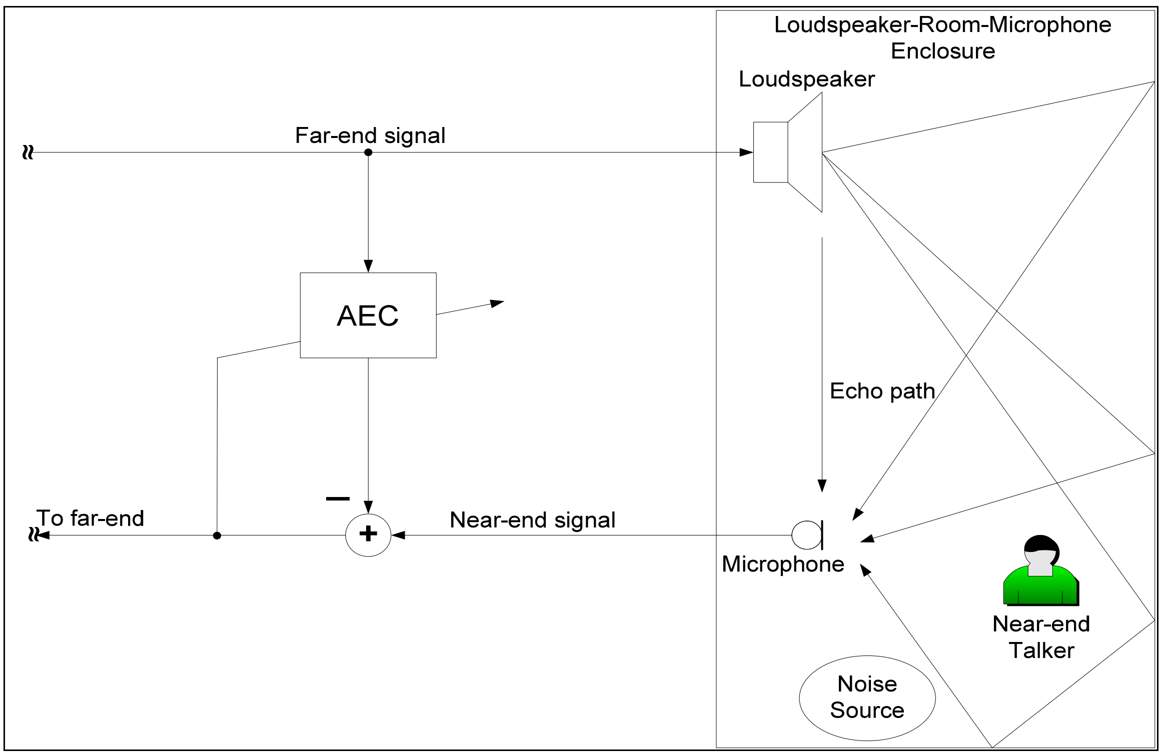

Figure 1 – Single Microphone Hands-Free System

Since the conversations that take place in hands-free communication systems, such as teleconferencing and mobile telephony, occur in real-time the acoustic echo canceller must also be implemented in real-time. One way to implement real-time acoustic echo cancellation is to use digital signal processing techniques employed on a programmable digital signal processor (PDSP).

This report discusses two acoustic echo cancellation algorithms that were implemented in real-time on a PDSP and compares them with previous implementations found in the literature. The algorithms are compared in terms of acoustic echo cancellation performance and computational complexity. The target PDSP that was used is the Texas Instruments TMS320C6713 floating-point device.

1.1. Background

The acoustic echo cancellation problem in hands-free communication is a system identification issue that can be readily solved via linear finite impulse response (FIR) adaptive filtering algorithms. AECs that are employed in hands-free communication systems use adaptive filters to model the associated loudspeaker-room-microphone (LRM) transfer function in order to remove acoustical echo and prevent it from corrupting the hands-free conversation as shown in Figure 1. The performance, complexity, and robustness of an AEC is governed by the adaptive filtering algorithm that is employs. Examples of adaptive algorithms that are used to implement real-time acoustic echo cancellation include, but are not limited to, the least mean squares (LMS) algorithm and its variants, including normalized least mean squares (NLMS) and block least mean squares (BLMS). Other adaptive algorithms used include fast versions of least squares (LS) and affine projection (AP) algorithms.

The authors in [5] implemented a real time AEC based on a modified LMS algorithm using an Analog Devices SHARC PDSP. They reported superior acoustic echo cancellation compared to an AEC using NLMS. However, better acoustic echo cancellation performance may have been achieved if a floating-point PDSP was used instead of a fixed-point PDSP as used by the authors in [5]. In [6] the authors employed a variable step size decorrelation LMS algorithm for their real-time AEC on a TMS320C30 PDSP. The authors in [6] demonstrated greater acoustic echo cancellation performance compared to a standard LMS algorithm for acoustic echo cancellation. However, the complexity of the algorithm used by the authors in [6] is greater than the LMS algorithm due to the variable step size approach they used. In [7] the authors designed a real-time AEC on a TMS320C5x fixed-point PDSP based on a modified FAP algorithm. The authors in [7] reported better acoustic echo cancellation performance compared to the standard NLMS algorithm. However, the improvement in performance occurred at the cost of increased computational complexity. Again, the use of a floating-point PDSP may have further increased the AEC performance over the fixed-point PDSP used by the authors in [7].

Disregarding the nonlinearities in the hands-free communication system, such as loudspeaker distortion, the linear model for one end of the hands-free communication system is depicted by equation (1) below. Also, it should be noted that acoustic echo cancellation is only performed when local talkers are quiet and that it is assumed that a double-talk detector is providing the information on whether or not a local talker is active. The implementation of a double-talk detector is outside the scope of this report. Figure 2, shown below, depicts one half of a hands-free communication system with system signals as indicted in the figure.

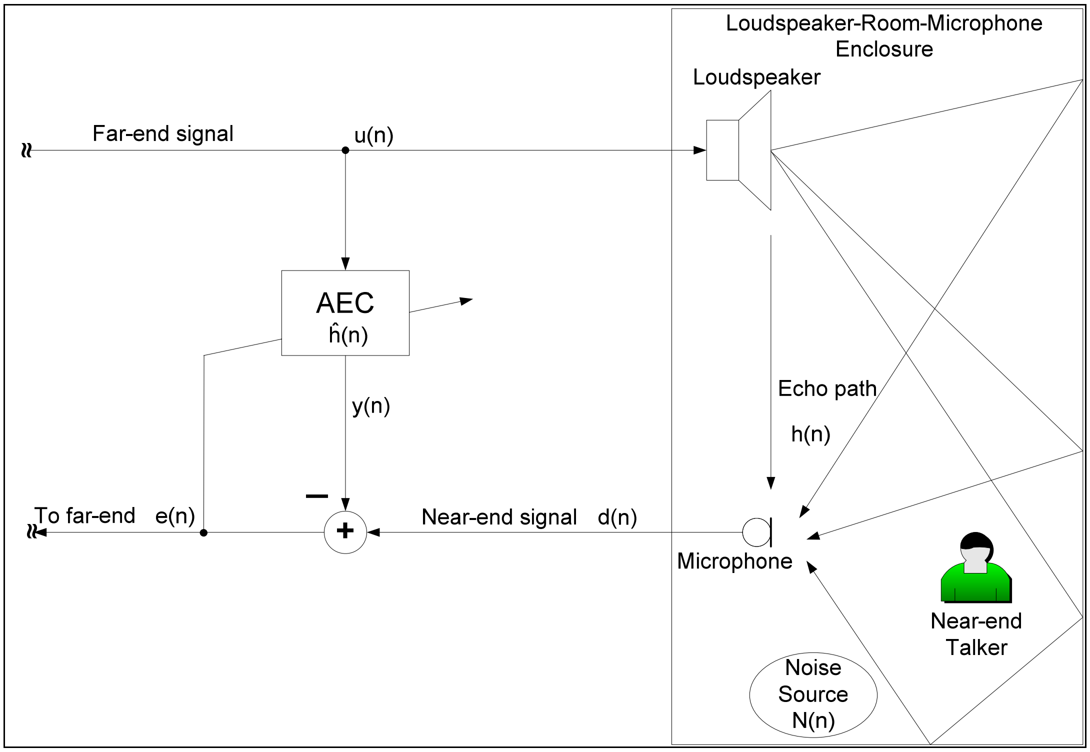

Figure 2 – Linear Model of a Single Microphone Acoustic Echo Canceller

Mathematically the acoustic echo cancellation problem depicted in Figure 2 can be modelled as follows:

$$\begin{aligned} e(n) &= d(n) - y(n) + N(n) \\ &= \boldsymbol{u}^T(n)\boldsymbol{h}(n) - \boldsymbol{u}^T(n)\hat{\boldsymbol{h}}(n) + N(n) \end{aligned} \qquad (1)$$

Where:
e(n) is the echo cancelled signal sent back to the far-end
u*(n)* is the *N*-by-1 far-end signal vector
d(n) is the near-end microphone signal
y(n) is the output signal from the AEC
N(n) is an additive noise signal
h*(n)* is the *N*-by-1 LRM impulse response vector
$\hat{\boldsymbol{h}}(n)$ is the *N*-by-1 estimated LRM impulse response vector
T is the transpose operator

Again, the performance of the acoustic echo cancellation system is determined by the adaptive algorithm that is used to model the LRM transfer function, *h(n)*. In this project the NLMS and BLMS algorithms were used due to their low complexity and ease of implementation. The details of both algorithms are discussed below.

A good theoretical overview of the NLMS algorithm is given in [8]. A summary of the NLMS algorithm, adapted from [8], and how it is applied in the above structure follows.

In the acoustic echo cancellation blocks of all the above structures the error signal, *e(n)*, is determined by the following equation:

$$e(n) = d(n) - \hat{\boldsymbol{h}}^H(n)\boldsymbol{u}(n) \tag{2}$$

Where:
e(n) is the algorithm error signal at time *n*
d(n) is the desired signal at time *n*
$\boldsymbol{u}(n)$ is an *N*-by-1 tap input vector an time *n*
$\hat{\boldsymbol{h}}(n)$ is the *N*-by-1 estimated adaptive filter taps at time *n*
N is the number of adaptive filter taps
H is the conjugate transpose operator

During each iteration of the NLMS algorithm equation (2) is calculated and the adaptive filter taps are updated according to the following equation:

$$\hat{\boldsymbol{h}}(n+1) = \hat{\boldsymbol{h}}(n) + \frac{\tilde{\mu}}{a + \|\boldsymbol{u}(n)\|^2}\boldsymbol{u}(n)e^*(n) \tag{3}$$

Where:
$\tilde{\mu}$ is the adaptation step size constant
a is a small positive constant

For the NLMS algorithm to converge in the mean square sense the following condition must be satisfied:

$$0 < \tilde{\mu} < 2 \tag{4}$$

The constant *a* helps to offset numerical difficulties that may occur when the value of the squared norm of the input vector is very small. Also, if no previous information of the adaptive filter taps is available then the taps are initialized to the zero vector at the beginning of the NLMS algorithm. The computational complexity of the NLMS algorithm is linear with respect to the number of adaptive filter taps, N. The total complexity of the NLMS algorithm per iteration is 2N+1 multiplications and 2N additions.

The BLMS algorithm is identical to the LMS algorithm with the exception that the adaptive filter weights are updated once per block of input data. The BLMS algorithm offers the advantage of faster implementation while maintaining equivalent performance compared to standard LMS. A detailed discussion of the BLMS algorithm is presented in [9]. The BLMS algorithm is shown below.

$$n = kL + i \tag{5}$$

$$\hat{\boldsymbol{h}}(k+1) = \hat{\boldsymbol{h}}(k) + \mu \sum_{i=0}^{L-1} \boldsymbol{u}(kL+i)e^{*}(kL+i) \qquad (6)$$

Where the signals are defined as above with the addition of:
L is the block size
k is the block number
i is the iteration variable in each block

Per block of input samples the computational complexity of the BLMS algorithm is $N(2L+1)$ multiplies and $2LN$ additions [9].

The remainder of this report is organized as follows. Section 2 outlines and discussed the design methodology that was used to implement the acoustic echo cancellation systems, based on the adaptive algorithms discussed above, on the target TMS320C6713 PDSP device. Section 3 analyzes the real-time experimental results obtained and compares them to results found in the literature. Finally, Section 4 concludes the main findings of this report.

2. Design Methodology

In order to facilitate the implementation of the algorithms on the TMS320C6713 Matlab and Simulink were used. This allowed for rapid development of the AEC algorithms along with easy TMS320C6713 assembly code generation and performance monitoring using the embedded target for TI C6000 DSPs and the link for Code Composer Studio (CCS) toolbox. The embedded target for TI C600 DSPs and link for Code Composer Studio integrates Matlab and Simulink with Texas Instruments eXpressDSP tools and allows for the design, simulation, and implementation of signal processing algorithms using blocks from available libraries within Simulink. The Simulink model file for the NLMS based acoustic echo cancellation system is shown below in Figure 3.

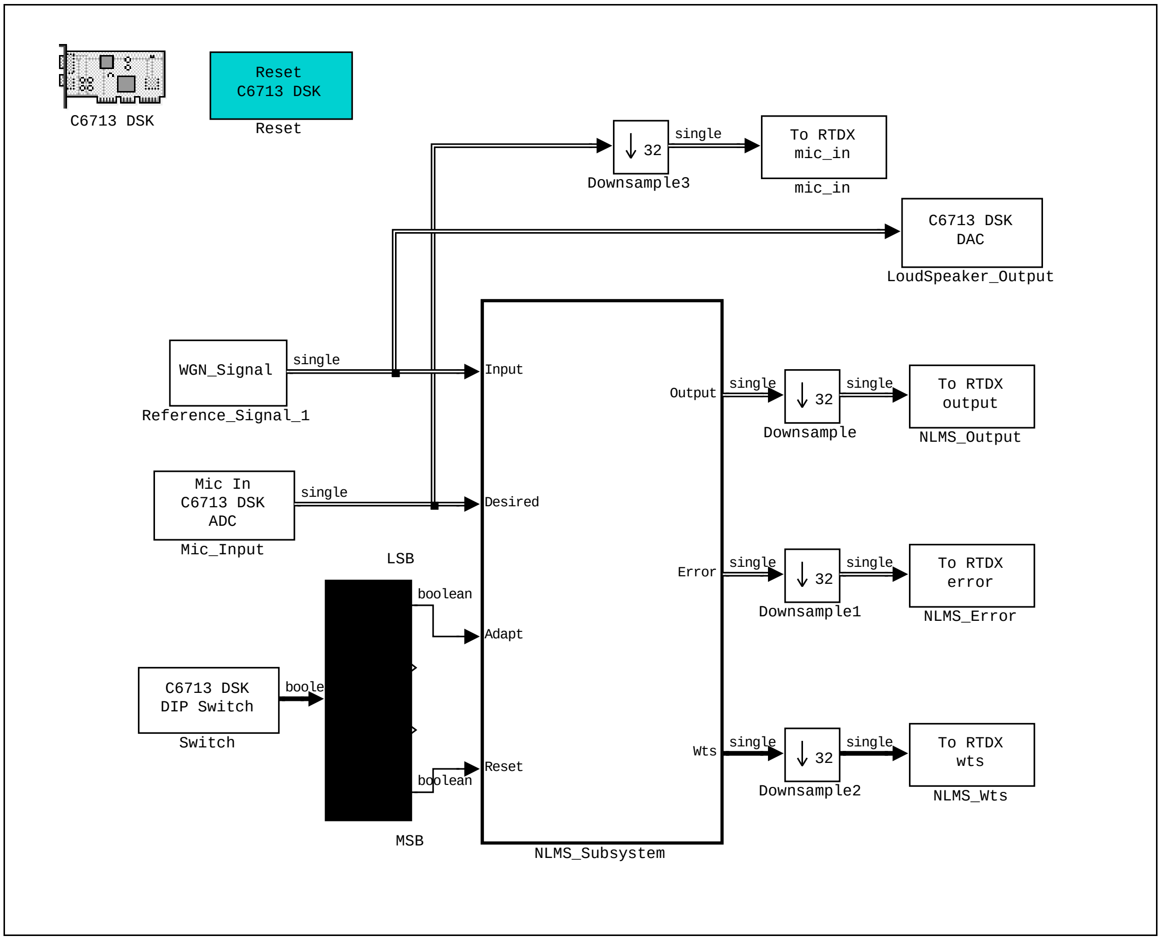

Figure 3 – NLMS Based Acoustic Echo Cancellation Simulink Model

The Simulink model file consisted of standard Simulink blocks along with specific blocks for the C6713 DSP Starter Kit (DSK). The C6713 DSK board houses the target TMS320C6713 PDSP along with analog-to-digital converters (ADC), digital-to-analog converters (DAC), synchronous DRAM, LEDs, switches, and other peripherals. A block diagram of the C6713 DSK board is shown in Figure 4 below. The Simulink NLMS AEC model included a C6713 DSK ADC block for the microphone input (d(n)), a from workspace block that copies the input signal (u(n)) from the Matlab workspace to memory on the C6713 DSK, a DAC block to allow for playback of the input signal into the acoustical environment, an NLMS block for performing the adaptation, a switch block to control the adaptation and reset the adaptive filter, and various signal monitoring blocks. The signal monitoring blocks were real-time data exchange (RTDX) blocks for the C6713 DSK board which allowed data to be read back in real-time into the Matlab environment from the TMS320C6713 PDSP.

Simulink models were created for NLMS and BLMS AEC systems consisting of 128, 512, and 600 filter taps. All of the model files appear the same as Figure 3 with only the underlying adaptive filtering blocks and their associated parameters, such as filter length, step size, block size, etc., differing. Once the model files were created their configuration parameters for C code generation, via Real-Time Workshop (RTW), were set in

Simulink. These C code configuration parameters included specifying the C6713 DSK as a target, requesting that profiling be performed, and setting the compiler optimization level. It should be noted that the compiler optimization was set to the function level for all models. Also, the NLMS and BLMS blocks in all model files were tagged as subsystems in order to allow for automatic profiling code to be generated so that the execution times of these algorithms could be monitored. This is discussed in detail in Section 3.

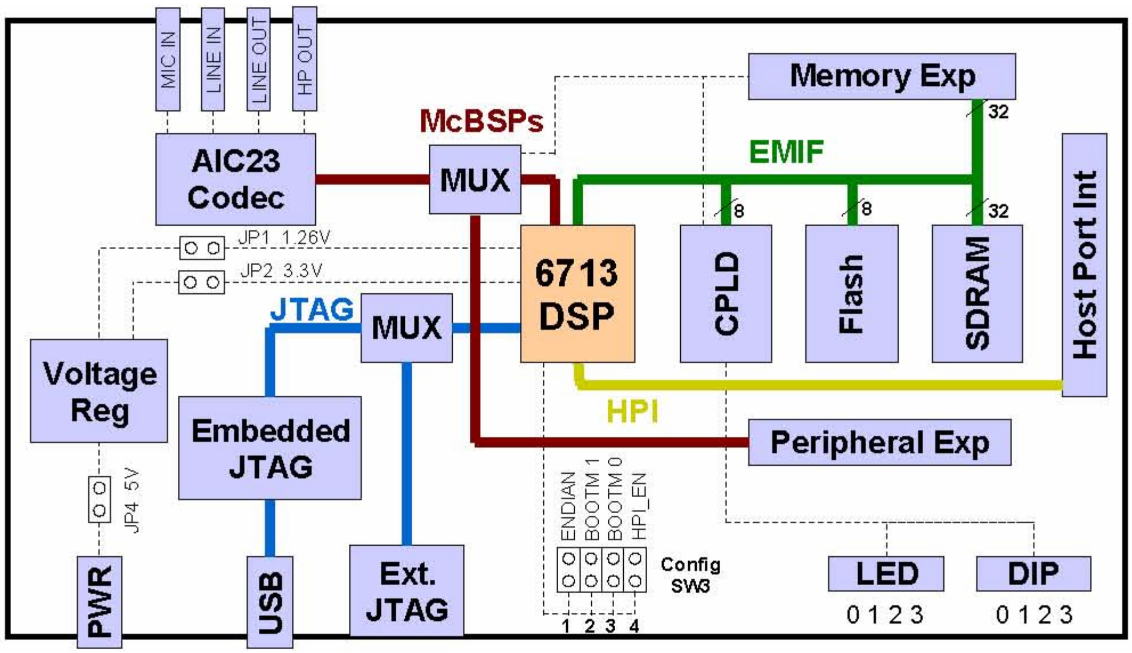

Figure 4 – C6713 DSK Board Block Diagram

After all C code configuration parameters were set the C code for the associated acoustic echo cancellation model was generated with RTW and compiled by CCS via the link for CCS toolbox. The final acoustic echo cancellation machine code was then loaded onto the C6713 DSK via CCS. Once the TMS320C6713 PDSP was programmed a Matlab script was used to monitor the AEC signals in real-time using the link for CCS toolbox.

3. Experimental Results and Analysis

As mentioned above NLMS and BLMS based acoustic echo cancellation systems were designed for 128, 512, and 600 tap filter lengths. These AECs were tested with white Gaussian noise (WGN) and speech signals in order to determine their performance bounds and study their real-world performance. It should be noted that the ADC for all models was set to sample at 8kHz in order to simulate real-world narrowband hands-free systems. Also, all models were set to work under frame based conditions with 64 samples per frame. In other words, the adaptive filtering algorithms must process the 64 sample input frames in 64/8000 seconds or 8 ms in order to avoid an overrun condition. Also, it should be noted that the acoustic echo cancellation system was operated in a typical conference room environment.

The first acoustic echo cancellation system implemented was the 128 tap NLMS based AEC under WGN conditions. The profiling report for this system reported an NLMS subsystem execution time of 7.89 ms. Thus, the 128 tap NLMS AEC was just on the verge of overrunning. However, with only 128 filter taps the NLMS AEC provided little to no actual acoustic echo cancellation. This is shown in Figure 5 below which presents the echo return loss enhancement (ERLE) results for the 128 tap NLMS AEC. ERLE is a measure of how much echo signal energy has been removed by the AEC and is governed by the following equation.

$$ERLE(dB) = 10\log_{10}\frac{E\{d^2(n)\}}{E\{e^2(n)\}} \tag{7}$$

Since the expectation of the signals in equation 7 is not generally known a moving average definition was used instead.

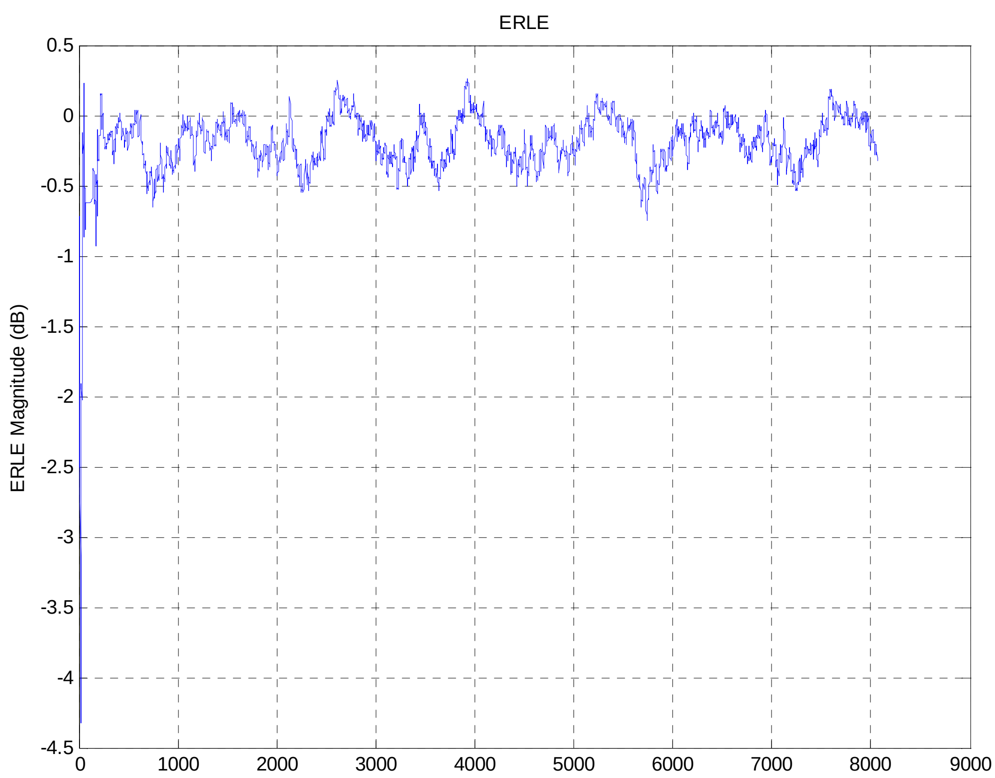

Figure 5 – ERLE of the 128 Tap NLMS AEC under WGN

As shown in the above figure no appreciable echo cancellation took place which meant that the length of the adaptive filter was inadequate at modelling the LRM transfer function. In order to properly model the LRM transfer function the adaptive filter needed to be increased. Increasing the NLMS adaptive filter from 128 to 129 taps resulted in a 64 sample frame execution time of 8.274 ms and thus overrun conditions occurred. In order to reduce the NLMS subsystem execution time the generated C code by RTW was inspected for inefficiencies. As it turned out the C code generated by RTW was indeed very inefficient. For example in the filter tap update term (see equation (3)) of the NLMS algorithm the C code generated by RTW performed the multiplication of the current input sample with the current error sample, the adaptive filter step size and divided by the current input signal frame energy, for each iteration in computing the updated filter taps. However, the step size, input signal frame energy, and current error sample do not change for the N iterations required to update the filter taps. Thus, performing these extra multiplication and division operations is needless and can be performed just once per N iterations in updating the filter taps. The RTW generated C code for the NLMS subsystem was modified to incorporate this change as well as hard coding the frame size and filter length in order to allow the compiler to optimize the generated machine code further. After making these changes the C code for the 128 tap NLMS AEC the system was recompiled and loaded onto the TMS320C6713 PDSP and a resulting frame execution time of 796.3 μs was reported. This was approximately a ten fold improvement in execution time. Thus, the bottleneck in the execution of the RTW generated C code for the NLMS adaptive filter seemed to be the redundant multiplications and divisions during the filter tap updates. Table 1 summarizes the frame execution times for the 128, 512, and 600 tap NLMS AECs after C code adjustments and for the same length BLMS AECs. It should be noted that no C code optimizations were required for the RTW generated C code for the BLMS AEC systems.

	Length of Adaptive Filter		
Adaptive Algorithm	128	512	600
NLMS	796.3 μs	2.824 ms	3.442 ms
BLMS	168.2 μs	578.1 μs	751.7 μs

Table 1 – NLMS and BLMS Frame Execution Times

As shown in Table 1 the execution time of the adaptive algorithms increased with increasing filter length as expected. Also, the BLMS algorithm executed approximately five times faster than the NLMS algorithm for each corresponding length of adaptive filter, which was expected.

With the optimizations made to the NLMS subsystem code, as discussed above, the NLMS AECs for 512 and 600 filter tapes were compiled and loaded onto the TMS320C6713 PDSP and tested under WGN and speech signals. The ERLE results for all filter lengths for the NLMS AEC are shown in Figure 6 below. It should be noted that the NLMS filters were adapted using a step size of 0.1 for all cases.

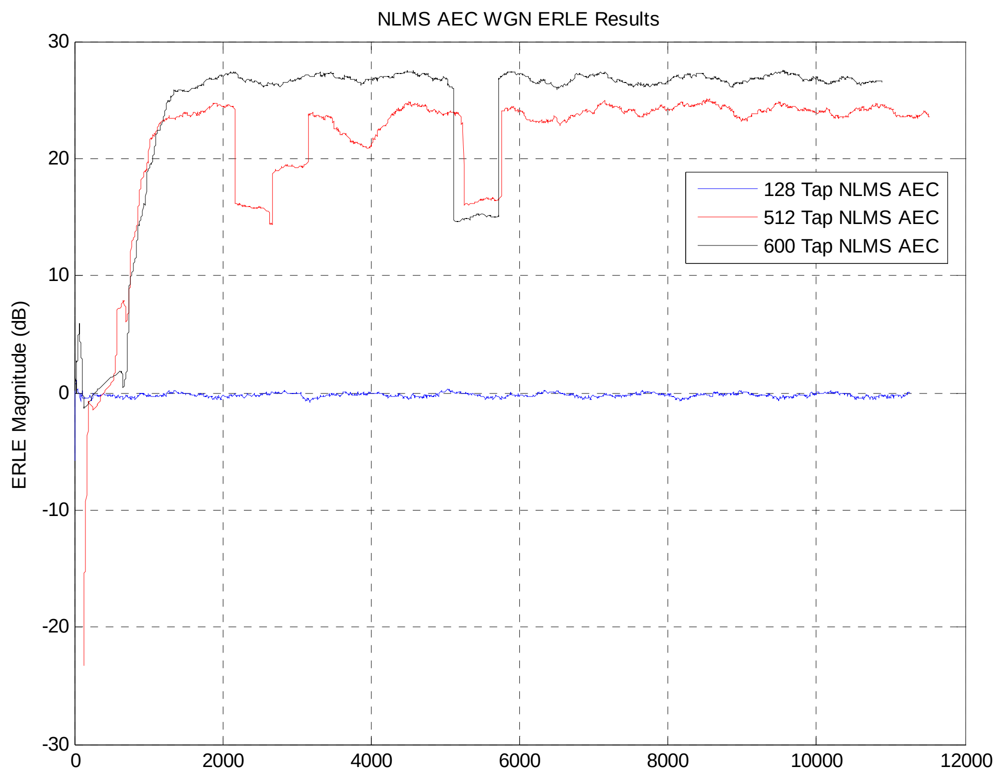

Figure 6 – NLMS AEC WGN ERLE Results

As shown in Figure 6 with no appreciable echo cancellation was obtained using a 128 tap NLMS AEC. This was a result of the severe under modelling of the acoustical environment LRM transfer function. As the number of taps was increased to 512 the NLMS AEC was able to achieve very good acoustic echo cancellation results. The converged 512 tap AEC ERLE performance was approximately 24 dB, increasing the number adaptive filter taps to 600 resulted in approximately 2 dB more of ERLE as shown in Figure 6. Also, these AEC performance results were in good agreement with the real-time results reported in [10]. The drops in the ERLE can be attributed to disturbances in the echo path during the experimentation (e.g. a door slamming).

Figure 7 shows the ERLE results obtained for the NLMS AEC system with speech as the input signal. As expected the NLMS AECs took much longer to converge with a coloured signal such as speech as the input signal [8], [10]. Again the 128 tap filter provided no acoustic echo cancellation; while the 512 and 600 tap AECs achieved approximately 20 dB of ERLE. It should be noted that the 600 tap NLMS AEC did not quite reach full convergence; however at full convergence it would have reached a slightly higher steady state ERLE than the 512 tap AEC as in Figure 6.

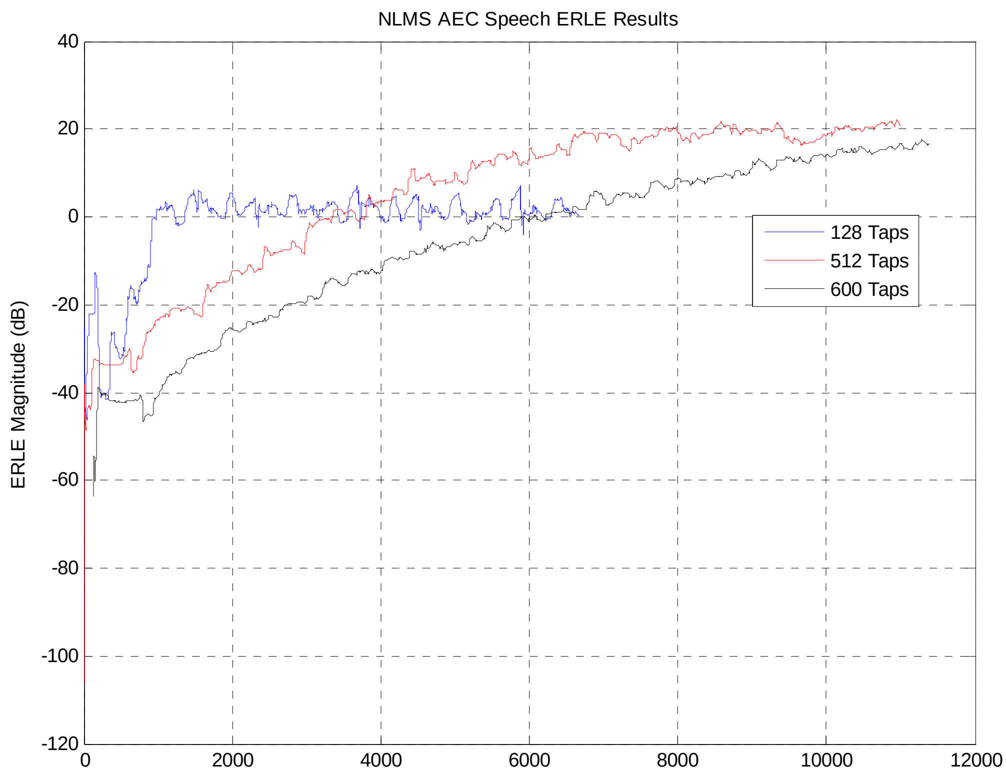

Figure 7 – NLMS AEC Speech ERLE Results

In Figure 8 the ERLE results for the BLMS AEC system with 128, 512, and 600 taps are presented under WGN conditions. Once again the 128 tap AEC provided no acoustic echo cancellation. The 512 tap AEC reached a steady state ERLE of approximately 23 dB while the 600 tap AEC reached a slightly higher steady state ERLE of approximately 25.5 dB. These steady state or converged ERLE results are in agreement with those reported above for the NLMS AEC of the same length which was as expected. Also, the drops in the ERLE for the 512 and 600 tap AECs can be attributed to fluctuations in the echo path as explained above for the NLMS case.

Figure 8 – BLMS AEC WGN ERLE Results

Figure 9 shows the ERLE results for the BLMS AEC system with speech as the input signal. The same speech sequence was used as in the NLMS AEC experiments. Again, in the presence of a coloured signal such as speech the BLMS AECs were slow to converge just as their NLMS counterparts, which was expected. The 512 tap and 600 tap BLMS AECs reached a steady state ERLE of approximately 20 dB which was similar to the NLMS case.

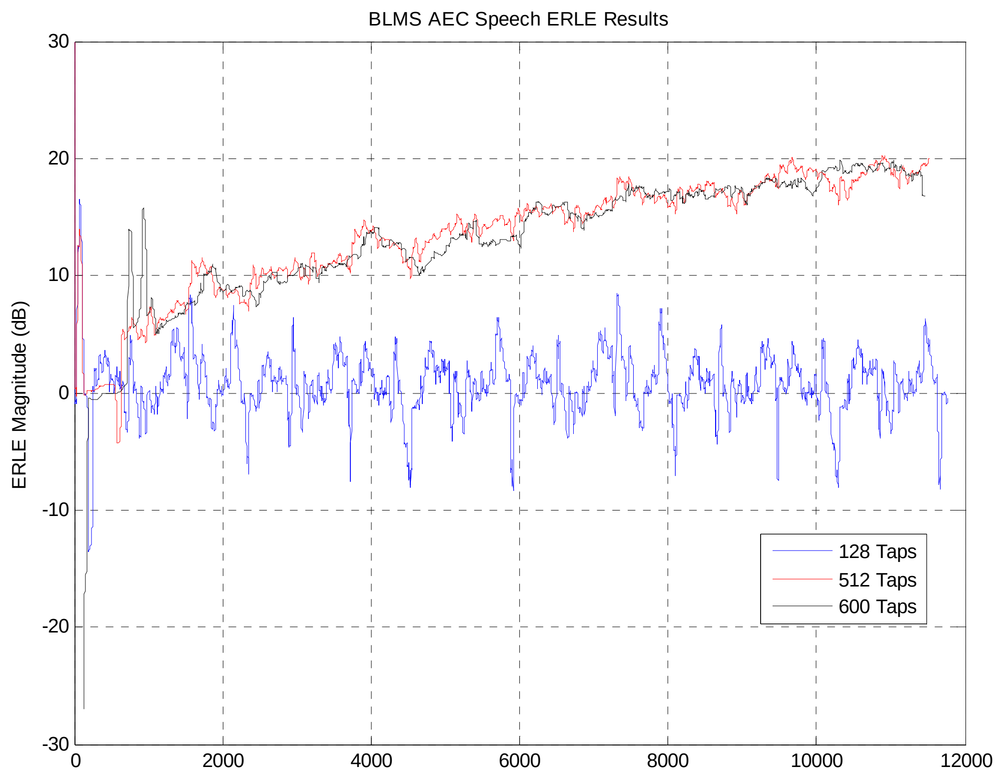

Figure 9 – BLMS AEC Speech ERLE Results

4. Conclusion

This report presented two real-time acoustic echo cancellation systems implemented on a C6713 DSK. The AEC systems were designed using Matlab and Simulink which allowed for rapid development and automatic code generation for the target device. The RTW code generated for the NLMS algorithm proved to be very inefficient and resulted in overrun conditions on the TMS320C6713 PDSP when a 129 tap filter was used. However, a ten fold improvement in execution time of the NLMS algorithm was realized by making a few small changes to the C code generated by RTW which allowed for much longer and useful NLMS AECs to be designed. The NLMS and BLMS based AECs achieved very good real-world echo cancellation results when adaptive filters employing 512 and 600 taps were used. The NLMS and BLMS AECs reached the same ERLE performance when WGN and speech were used as input signals for all filter lengths, as seen from Figures 6 through 9. Also, both systems were much slower to converge when speech was used as the input as shown in Figures 7 and 9 respectively. The BLMS based AECs operated approximately five times faster than the equivalent length NLMS based AEC as shown by the real-time profiling date in Table 1. In addition, increasing the adaptive filter length beyond 512 taps for both the NLMS and BLMS AECs resulted in only a small increase in ERLE performance. Also, the real-time performance of the NLMS and BLMS AECs were in agreement with the performance of other real-time PDSP AEC implementations reported in the literature.

References

[1] D. G. Messerschmitt, “Echo cancellation in speech and data transmission,” *IEEE J. Select. Areas Commun.*, vol. SAC-2, pp. 283-297, Mar. 1984.

[2] K. Murano, S. Unagami, and F. Amano, “Echo cancellation and applications,” *IEEE Commun. Mag.*, vol. 28, pp. 49-55, Jan. 1990.

[3] M. M. Sondhi and D. A. Berkeley, “Silencing echoes in the telephone network,” *Proc. IEEE*, vol. 68, pp. 948-963, Aug. 1980.

[4] M. M. Sondhi and W. Kellermann, “Adaptive echo cancellation for speech signal,” *Advances in Speech Signal Processing*, S. Furui and M. Sondhi, Ed. New York: Dekker, ch. 11, 1992.

[5] Wee Chong Chew, and B.F. Boroujeny, B.F, “Software simulation and real-time implementation of acoustic echo cancelling,” in *Proc. IEEE ICICS*, vol. 3, pp. 1270-1274, Sept. 1997.

[6] J.F. Doherty, and R. Porayath, “A robust echo canceler for acoustic environments,” *IEEE Trans. Circuits Syst. II*, vol. 44, no. 5, pp. 389-396, May 1997.

[7] S. Oh, D. Linebarger, B. Priest, and B. Raghothaman, “A fast affine projection algorithm for an acoustic echo canceller using a fixed-point DSP processor,” in *Proc. IEEE ICASSP*, vol. 5, pp. 4121-4124, April 1997.

[8] S. Haykin, *Adaptive Filter Theory*, 3rd ed., Upper Saddle River, NJ: Prentice-Hall, 1996.

[9] G.A. Clark, S.K. Mitra. S.R. Parker, "Block implementation of adaptive digital filters." *IEEE Trans. Acoust., Speech, Signal Processing*, vol 29, no. 3, pp. 744-752, June 1981.

[10] H. Ding, “A stable fast affine projection adaptation algorithm suitable for low-cost processors,” in *Proc. IEEE ICASSP*, vol. 1, pp. 360-363, June 2000.