Conception and Real Time DSK C6713 of a Low Cost Adaptive Acoustic Noise Cancellation (ANC) Based Fast Fourier Transform (FFT) and Circular Convolution for Improving the Quality of Voice Communications

Azeddine Wahbi, Ahmed Roukhe, Laamari Hlou

Laboratory of Electrical Engineering and Energy System Faculty of Science, University Ibn Tofail, Kenitra, Morocco

Laboratory of Atomic, Mechanical, Photonics and Energy Faculty of Science, University Moulay Ismail, Meknes, Morocco

Corresponding author E-mail: wahbi_azeddine@yahoo.fr

Abstract—Acoustic noise canceller (ANC) is currently the technology emergent in the field of communications system. However quality of voice communications is one of the major aspects of communications system due to the concurrence in this field. In this paper we designed, developed and implemented on a fixed point DSP C6713 of a low cost adaptive acoustic noise canceller for improving the quality of the communication against the undesirable phenomena such as acoustic noise, and we focus on new methods of digital signal processing especially adaptive filter in frequency domain. The main scope of this paper is to implement the module, benefiting the advantage of DFT (Discrete Fourier transform) circular convolution properties and Fast Fourier Transform (FFT) high computation speed in frequency domain rather adaptive algorithms Normalized Least Mean Square (NLMS) and Recursive Least Square (RLS) in time domain with high complexity, also the simplicity of the implementation using simulink programming.

The needed DSP code is generated in code composer environment under Real Time Workshop. At the experimental level, implementation phase results verify that implemented module behavior is similar to Simulink model.

Keywords—Adaptive Algorithm; ANC; Real Time Implementation; Circular Convolution; Fast Fourier Transform (FFT); Digital Signal Processing (DSP).

I. INTRODUCTION

Frequently, acoustic noise is considered as the main problem for communication systems. Moreover, it is may further decrease the quality of voice communications, when it is dominated with speech signals by the voice gateways [1].

As multimedia transmission quality via communication systems is essential, the Acoustic Noise Cancellation (ANC) can be present a best solution, because it is widely used in real time applications in digital signal processing for its stability and simplicity.

In this work, the method used to achieve noise cancellation is known as adaptive filtering. This method is frequently used to enhance communication quality by removing line noise. This is why adaptative filters were developed and tested long before on analog bench platforms until a digital based technique breakthrough emerged, the DSP. This new technique allows better signal filtering design and found its benefits in High Fidelity audio systems or speech networks.

Different works involving noise cancellation adaptive algorithms developed across this paper are presented [2-3-4-5].

This paper will focus on software based FLMS adaptive algorithm to remove noise in voice communication systems. However, the ANC is modeled in Simulink environment by using digital filters, especially adaptive Fast Least Mean Square (FLMS) algorithm based FFT/IFFT operations and circular convolution frequency domain that require approximately NlogN real multiplications and reduce the computational complexity compared to RLS adaptive algorithms modeled by A.wahbi in [6] and NLMS adaptive algorithms presented in [7-8-9], which deal with time domain based on a linear convolution. Those algorithms require respectively 4N^2 and 3N+1 real multiplication. Finally this module is implemented and verified on a fixed point DSP C6713.
The paper is structured as follows: section II presents digital adaptive filters for noise cancelling, section III presents the DSK TMS320C6713 card, section IV presents simulation results, Section V presents module design and Section VI concludes this paper.

II. DIGITAL ADAPTIVE FILTERS FOR NOISE CANCELLING

Developing a filter that is able to comply with the statistics of the signal is the main scope of adaptive filtering. Adaptive algorithm efficiency depends on three criteria that size up:

- The complexity of computation and the amount of computation executed at each stage.
- The behavior of speed adjustment that permits an adaptive filter to reach Weiner solution.
- The estimated error generated by the dissimilarity between the actual Weiner solution and the adaptive algorithm resolution.

Adaptive cancellation of noise is the main pattern of adaptive filters.

A. Adaptive Filters

In this section we first go through an examination of the filter structure with an emphasis on Finite Impulse Responses (FIR) filters. This is followed by a review of the Wiener filter leading to the development of the Fast Least Mean Squares (FLMS) algorithm.

An acoustic noise canceller is a closed loop linear adaptive filter used for direct system modeling. There are many different combinations of filters and algorithms, depending on the particular application requirements. For noise cancellation, there is a classical standard adaptive filter formation. The filter part is made up of the most commonly used structure: a FIR filter which is also known as a tapped delay line, non-recursive or feed-forward transversal filter, as shown in Fig 1.

The FIR filter consists of a series of delays, multipliers and adders; has one input, x(n), and one output y(n). The output is expressed as a linear combination of the delayed input samples:

\[ y(n) = \sum_{j=0}^{L-1} h_j x(n-j) \]  

(1)

Where \( h_j \) are the filter coefficients and \( L \) is the filter length. Therefore \( y(n) \) the convolution (inner product) of the two vectors \( h \) and \( x(n) \). In this paper we will only consider FLMS filters for noise cancellation.

B. Fast Fourier Transform

For calculating the FFT (Fig. 2), \( M \) is to be chosen as a power of 2. In general, we choose \( N+L-1 \) if this value is suitable. Otherwise, we choose the nearest power of 2 that is greater than the latter value (in which case we have to complement the vectors by the number of zero coefficients necessary and discard the last vector components introduced during the reverse transformation). To make it simple, we assume here that we choose \( M=N+L-1 \) in order to give an implementation example with values of \( M \) greater than \( N+L-1 \).

Let \( W_F \) be the DFT matrix having \( M \) (\( M \times N \)) in size and the following coefficients:

\[ W_F(k,l) = \exp(-2j\pi kl/M) \]  

(2)

and \( W_F^{-1} \) the inverse transform matrix:

\[ W_F^{-1}(k,l) = \frac{1}{M} \exp(2j\pi kl/M) \]  

(3)

C. Fast adaptive algorithm in the frequency domain

One of the adaptive filter applications is the adaptive noise canceller. Fig. 3 describes its structure where the desired response is composed of an original signal plus the noised, which is uncorrelated with the signal.

The filter input is a sequence of a noised signal which is correlated with the noised signal in the desired signal.
By using the FLMS algorithm inside the adaptive filter, the error term \( e(n) \) produced by this system is then the original signal with the noise signal cancelled.

![Diagram](image)

**Fig. 3. Fast adaptive filter structure**

D. Fast Convolution in the Fourier transform domain

Noise signal values \( y_j \) are estimated, as shown above, by means of a linear convolution represented by the following equation:

\[
y_j = x_j \cdot h_j
\]

(4)

This calculation is very consuming regarding computation. The main idea emerging in terms of fast convolution algorithms is to fulfill convolution in the Fourier transform domain according to the principle of duality. Indeed, circular convolution in the time domain is equivalent to a term to term multiplication in the DFT frequency domain and hence quick calculation algorithms of the DFT are then used to perform this operation with reduced complexity.

The filter is implemented in the circular convolution frequency domain between vectors \( h_j \) and \( x_j \), both having length \( N+L-1 \), and respectively defined by:

\[
h_j = [h_j(0) h_j(1) \ldots h_j(L-1); 0 \ldots 0]^T
\]

(5)

Let \( y_j \) be the product of the circular convolution between \( x_j \) and \( h_j \):

\[
y_j = x_j \otimes h_j
\]

(6)

where \( \otimes \) represents the circular convolution product between two vectors.

The length of vector \( y_j \) is \( N+L-1 \), and its last \( N \) components correspond to the linear convolution of equation (4), that is to say to \( Y_j \) components. The first \( L-1 \) components result from circular convolution and should be excluded.

In a formal way, we exclude these first \( L-1 \) components by considering the truncation matrix \( T_N \) having \( N \times (N+L-1) \) in size and defined by:

\[
T_N = \begin{bmatrix}
O_{N \times (L-1)} & I_N
\end{bmatrix}
\]

(7)

Then we have:

\[
Y_j = T_N \cdot y_j
\]

(8)

The circular convolution completion is done in the frequency domain using the respective DFT of \( x_j \) and \( h_j \), noted therefore \( X_j \) and \( H_j \) plus having \( M \) in length:

\[
X_j = W_F \cdot x_j
\]

(9)

\[
H_j = W_F \cdot h_j
\]

(10)

The calculation of the linear convolution is then performed according to the following expression:

\[
Y_j = T_N \cdot W_F^{-1} (X_j \square H_j)
\]

(11)

Where \( \square \) represents the scalar Schur product, or the component to component product of the vectors. Returning to the equation (12) for the error signal.

\[
e_j = d_j - Y_j
\]

(12)

we then obtain:

\[
e_j = d_j - T_N \cdot W_F^{-1} (X_j \square H_j)
\]

Through the use of FFT, the calculation of the error generated by each block in this manner is less resource consuming than the block-based temporal LMS.
E. Adaptive filter update

As for the calculation of the error, the update of the $h_j$ filter can be performed in the frequency domain with a lower computational cost. The update equation (14) in the time domain can be expressed quite simply in the frequency domain.

For this purpose, we define the DFT of the error sequence, $E_j$, as follows:

$$E_j = W_F \cdot \begin{bmatrix} 0_{t-1} \\ \vdots \\ e_j \end{bmatrix}$$

(13)

at each iteration “j”, the update of the adaptive filter is then performed according to the equation:

$$H_j = H_{j+1} + \lambda_j \cdot \left( X_j \cdot E_j \right)$$

(14)

Where $\lambda_j$ is the adaptation step that controls the speed of convergence of the algorithm.

F. FLMS Algorithm

Block LMS (BLMS) algorithm can be made less computation time consuming if the temporal convolution is achieved in the frequency domain. Doing so, it is therefore possible to take advantage of DFT (Discrete Fourier transform) circular convolution properties and Fast Fourier Transform (FFT) high computation speed.

The FLMS Filter block shown in Fig 4 implements an adaptive Fast least Mean square (FLMS) filter, where the adaptation of filter weights occurs once for every block of samples.

The block estimates the filter weights, or coefficients, needed to convert the input signal into the desired signal.

Connect the signal you want to filter to the Input port. This input signal can be a sample-based scalar or a single-channel frame-based signal.

Connect the signal you want to model to the Desired port. The desired signal must have the same data type, frame status, complexity, and dimensions as the input signal.

The Output port outputs the filtered input signal, which can be sample or frame based. The Error port outputs the result of subtracting the output signal from the desired signal.

G. Signal to Noise Ratio (SNR)

Signal Noise to Ratio (SNR) is often expressed using the logarithmic decibel scale. In decibels, the SNR is defined as:

$$SNR_{dB} = 10 \log_{10} \left( \frac{\text{Power(signal)}}{\text{Power(noise)}} \right)$$

(15)

$$= \text{Power(signal)}_{dB} - \text{Power(noise)}_{dB}$$

However, the SNR plays an important role in digital signal processing; it is a measure used in that compares the level of a desired signal to the level of undesirable noise.
IV. SIMULATION RESULTS

A. Noise Canceller Modeling Under Simulink

In this work we modeled the system shown in fig. 7 under Simulink Blockset. In this case, the noise type is Gaussian with zero mean and one variance.

Also, we designed a reliable module based S-function for evaluating the performance of the ANC by calculating the SNR values.

The number of samples for each read from file (From Wave File) length is 256 samples per frame at 8000 Hz sampling rate. ANC implementation is setup with filter length=128 and block size of FFT=128. The step size is chosen as $\lambda = 0.0005$. 

\[
\begin{aligned}
X_j & \quad \text{Adaptive Algorithm in Frequency Domain} \\
& \quad - E_j \\
\end{aligned}
\]
B. Simulink Results

In the following graphics (figs. 9, 10, and 11), we observe the input signal (the original signal with noise) and how this noise is removed from the original signal after crossing by the “noise cancellation Fast adaptive Filter in frequency domain” module, knowing the noise signal has less amplitude than the original signal.

It is also showing how the signal is filtered, and the result is an output signal with less amplitude than the input signal and without noise.

Fig. 8. Noise cancelation under Simulink

Fig. 9. Result obtained using Simulink simulation (Original signal, Variable step size is \( \lambda = 0.0005 \))
The effect of modifying the step-size, the filter length, the delay value on the convergence rate and obtainable performance is tested.

It should be verified that a shorter filter length is required to obtain the desired cancellation while using the input signal, a wav file. Unofficial hearing tests should prove that the system is working properly: the periodic signal is almost cancelled whereas the speech maintains its natural quality.

<table>
<thead>
<tr>
<th>Time (S)</th>
<th>SNR of enhanced output (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20.96</td>
</tr>
<tr>
<td>2</td>
<td>24.2</td>
</tr>
<tr>
<td>3</td>
<td>25.9</td>
</tr>
<tr>
<td>4</td>
<td>30.67</td>
</tr>
<tr>
<td>5</td>
<td>20.39</td>
</tr>
<tr>
<td>6</td>
<td>27.2</td>
</tr>
<tr>
<td>7</td>
<td>30.11</td>
</tr>
<tr>
<td>8</td>
<td>28.13</td>
</tr>
<tr>
<td>9</td>
<td>19.31</td>
</tr>
<tr>
<td>10</td>
<td>26.94</td>
</tr>
</tbody>
</table>

Hence the performance of the ANC is verified noise and shows a higher level of improvement in Signal to Noise Ratio (SNR). The SNR for noise colored signal is 3.182 dB and enhanced output SNR is shown in Table I. The undesirable noise is almost completely cancelled.

V. MODULE DESIGN

A. Real time ANC modeling experimental level

Fig. 12 shows a design of ANC in real time using digital signal processing.
**B. Experimental setup**

The station is equipped with the following items: Personal computer with Matlab/Simulink and Code Composer Studio (CCS), C6713 DSP board, microphone, loudspeakers, headphones, oscilloscope and a signal generator. A typical station is shown in Fig. 13. The core of the station is the DSK C6713 evaluation board (Figs. 5 and 6).

**C. Real Time Implementation and Testing**

In the following paragraphs, the module implementation on C6713 DSP is discussed. In Layman’s terms, the module functionalities are exposed as independent blocks which are thereafter mixed into a single program that integrates C code inside the Code Composer Studio v3.3 (CCS) environment. The CCS compiles it, prepares necessary links, and then loads it into the target processor. Finally, the DSP processes the implemented algorithm and executes the code as shown in figs. 14, 15 and 16.
D. Experimental Results

In real world application, the module was tested and led to the following results. We notice that this module is a real-time process and that the graphs are similar to those generated using Simulink simulation. The following figures give an idea of what is produced by the module. Then, in order to verify proper switching of the module input, noised and output signals are probed on an analog oscilloscope as illustrated respectively in Figs 17, 18, 19.

The Result of Real-time implementation of the NLMS algorithm is carried out with the following specifications:

- Filter order N=32
- Variable step size $\lambda = 0.0005$
- Number of samples per frame in ADC source = 256.
VI. CONCLUSION

In this paper, we have tried to design and real time implements onboard an autonomous DSK C6713 of an adaptive filter module in frequency domain based FFT\!\!IFFT operations and circular convolution with low complexity rather than algorithms in time domain with low computation speed. This module, consisting of software blocks, was specifically designed to provide noise cancellation in voice communications to ensure enhanced quality of voice communications.

On the basis of the value of SNR of enhanced output, the proposed ANC represents a robust solution for enhancing the quality of voice communications in real time applications.

REFERENCES


