A FPGA Implementation of Prediction Error Method for Adaptive Feedback Cancellation using Xilinx System Generator™

Marius Rotaru, Cristian Stanciu, Silviu Ciochină
Dept. of Telecommunications
University Politehnica of Bucharest
Bucharest, Romania
marius.rotaru@gmail.com, {cristian, silviu}@comm.pub.ro

Felix Albu, Henri Coandă
Electrical Engineering Department
Valahia University of Targoviste
Targoviste, Romania
{felix.albu, coanda}@valahia.ro

***Abstract*—This paper describes a real-time, field programmable gate array (FPGA) implementation of Feedback Cancellation (FC) system to improve the intra-cabin communication among the driver and passengers, which is typically degraded by the noisy environment and by the distance in between them. The feedback canceller, used to reduce the acoustic coupling the loudspeaker and the microphone, is based on the continuously adaptive filtering technique, implementing the prediction error method (PEM) for closed loop system identification. The adaptive algorithm implements the modified least mean square algorithm (MLMS) while for the linear prediction a fix-order linear predictor has been selected. The implementation was done using Xilinx System Generator™ (XSG)**

Keywords-Adaptive Algorithm; Adaptive Feedback Cancellation; Prediction Error; MLMS; FPGA; Xilinx System Generator

I. INTRODUCTION

The acoustic feedback is a major problem of audio processing field, occurring whenever the sound is captured and reproduced in the same environment. The communication in vans and limousines between the passengers in the front and the rear is degraded due to the presence of the noise as well as the long distance between them [1]. This can be improved by using a speech reinforcement system. The simplest, one channel speech reinforcement system, picks-up the speech using a microphone, amplifies it and then plays it back to a loudspeaker. Due to the electro-acoustic coupling between loudspeakers and microphone, a closed-loop system is created. To avoid the instability (howling) of the system, a feedback canceller has to be used.

Different methods attempting to minimize the effect of acoustic feedback have been proposed in literature. They are broadly classified as feedforward suppression and feedback cancelation techniques. For the case of feedforward suppression technique, the use of notch-filter based howling suppression (NHS) represents a traditional and robust solution [1-3]. The main disadvantage of this method is that it is reactive: in order to identify and eliminate the oscillation frequencies the howling must firstly occur. A more promising solution is to use the adaptive feedback cancellation (AFC) method, which is based on the estimation

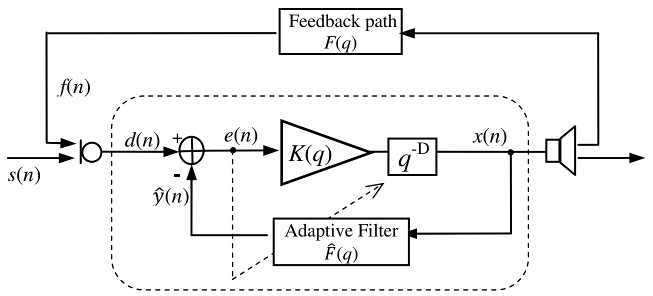

Figure 1. Adaptive Feedback cancellation structure

of acoustic feedback path, belongings to the class of room modeling methods. As illustrated in the Fig. 1, the feedback canceller $\hat{F}(q)$ produces an estimate $\hat{y}(n)$ of the feedback signal $f(n)$, obtained by filtering the loudspeaker signal $x(n)$ with $F(q)$, and subtracts it from the microphone signal $d(n)$ so that ideally the clean speech signal $s(n)$ amplified by a factor K and played back to the loudspeaker. Depending on the quality of the feedback path estimation, the feedback is almost eliminated. The most robust method to eliminate it is based on the closed loop identification theory [4] - [5]. The direct method for closed loop identification [5], named prediction-error-method (PEM) is a promising proactive solution for AFC. Prediction error for AFC (PEMAFC) algorithms has been deeply analyzed in context hearing aids applications [6-10]. Also, recently the PEMAFC method has been tested in the car scenario case [11-12]. In this paper we propose a FPGA implementation of the PEMAFC algorithm in Xilinx System Generator™, based on the configuration from [8-9] using a fix model of input signal.

The paper is organized as follows: Section II is dedicated to the description of the PEMAFC algorithm implemented on FPGA. The Section III covers the FPGA implementation aspects. In Section IV the experiments as well as the results are reported. Section V presents the conclusions of the work.

II. PREDICTION ERROR METHOD FOR AFC SYSTEM

The notation from [8] has been adopted: q^{-1} denotes the unitary delay, so that $q^{-1}\ u(n) = u(n-1)$. A discrete-time filter with filter length L is represented as a polynomial $F(q)$ in q, i.e.,

$$F(q) = f_0 + f_1 q^{-1} + \cdot\cdot\cdot + f_{L-1} q^{-L+1}, \tag{1}$$

or by its vector $\mathbf{f}=[f_0, f_1 \ldots f_{L-1}]^T$, so that the filtering operation consists of applying the polynomial to the input sequence:

$$F(q)x(n) = \mathbf{f}^T \mathbf{x}(n), \tag{2}$$

with $\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-L+1)]^T$.
As proposed in the [6], [7] and reiterated in [8], the direct method of the closed-loop identification of feedback path $F(q)$ and the desired signal model $H(q)$ is presented in the Fig.2. The main assumption of this method starts from the fact that the desired signal $d(n)$ can be modeled as a $H(q)w(n)$, where $w(n)$ is the white noise signal and the $H(q)$ is the desired signal model, which is inversely stable ($A(q)=H^{-1}(q)$). In such case the bias in the feedback path estimation can be eliminated by decorrelating the signal of the adaptive algorithm by passing them through $A(q)$. In case of PEM, the feedback path $F(q)$ and the desired signal model $H(q)$ are estimated by minimizing the energy of so calling prediction error $e_p(n)$:

$$e_p(n) = \hat{H}^{-1}(q)(d(n) - \hat{F}(q)x(n)), \tag{3}$$

i.e., $$J(\hat{\mathbf{f}}(n)) = E\{|\hat{H}^{-1}(q)(d(n) - \hat{F}(q)x(n))|^2\}$$
$$= E\{|(d_p(n) - \hat{\mathbf{f}}^T(n)\mathbf{x}_p(n))|^2\}. \tag{4}$$

Minimizing (4) generates:

$$\hat{\mathbf{f}}(n) = E\{\mathbf{x}_p(n)\mathbf{x}_p^T(n)\}^{-1} E\{\mathbf{x}_p(n) d_p(n)\}. \tag{5}$$

Writing $d_p(n)$ as:

$$d_p(n) = \hat{H}^{-1}(q)s(n) + F(q)x(n). \tag{6}$$

When $H(q) = \hat{H}(q)$,

$$d_p(n) = w(n) + F(q)x(n), \tag{7}$$

and the speech signal $s(n)$ is converted to a white noise signal $w(n)$, resulting an unbiased feedback path estimate.
From (4) it can be observed that minimizing $J(\hat{\mathbf{f}}(n))$ is equivalent with performing an adaptive filtering on the decorrelated (pre-whitened) signals $\{d_p(n), x_p(n)\}$ or equivalently on $\{e_p(n), x_p(n)\}$.

Both fixed and adaptive estimates of the desired signal model $H^{-1}(q)$ have been considered in literature [9]. Since our attention is focused on the FPGA implementation, in this paper we choose the fixed model, representing the averaging of speech spectrum, defined by:

$$H(q) = \frac{1}{1 - \alpha q^{-1}}, \quad |\alpha| < 1 \tag{8}$$

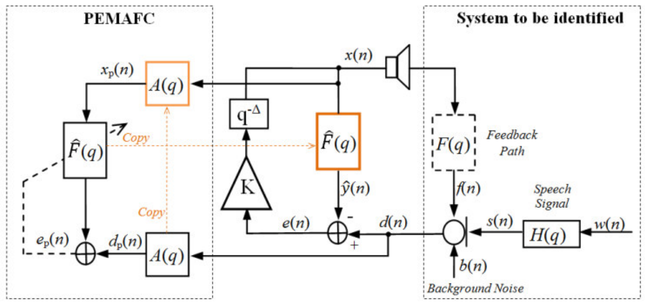

Figure 2. Feedback cancellation with prediction error method

The adaptive filtering algorithm selected to perform these minimization is the Modified LMS (MLMS) [13], which is particularly suited for adaptive systems whose performance suffers from the presence of strong target signals (such as speech) that exhibit large fluctuations in short-time power levels. The sum version of this algorithm has been selected, with the following formula for updating the filter coefficients:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + e(n)\mathbf{x}(n) f[x(n), e(n)], \tag{8}$$

$$f[x(n), e(n)] = \mu(n) = \frac{\bar{\mu}}{L(\hat{\sigma}_e^2(n) + \hat{\sigma}_x^2(n)) + \varepsilon}, \tag{9}$$

where L is the length of adaptive filter, $\bar{\mu}$ is an adaptation constant and $\hat{\sigma}_e^2$ and $\hat{\sigma}_x^2$ are the power estimates of the error respectively of the reference signal. They can be obtained as follow:

$$\hat{\sigma}_{e,x}^2(n) = \lambda \hat{\sigma}_{e,x}^2(n-1) + (1-\lambda)(e^2(n), x^2(n)), \tag{10}$$

where λ is a weighting factor chosen as $\lambda = 1-1/(KL)$, with $K>1$. A Summary of PEMAFC algorithm is described in the Table.1.

TABLE I. TIME-DOMAIN PEMAFC ALGORITHM FOR THE FIXED ORDER LP

```
Initialization:
    L, $\varepsilon$, $\bar{\mu}$, $\lambda$, $\hat{\mathbf{f}}(\mathbf{0})=\mathbf{0}$, $e(0)=0$,
Computation: for each input sample $x(n)$, $n$= 1, 2,. . .
    $e(n) = d(n) - \hat{\mathbf{f}}^T(\mathbf{n})\mathbf{x}(n)$;
    $x(n) = K \cdot e(n-D)$
    Pre-filter $e(n)$ and $x(n)$ with $A(q)$
        $e_p(n) = e(n) - \alpha e(n-1), \quad x_p(n) = x(n) - \alpha x(n-1),$
    Estimate the power of pre-filtered signals
        $\hat{\sigma}^2(n) = \lambda \hat{\sigma}^2(n-1) + (1-\lambda)(e_p^2(n) + x_p^2(n)),$
    Update the coefficients
        $\hat{\mathbf{f}}(n+1) = \hat{\mathbf{f}}(n) + \frac{\bar{\mu}}{L\hat{\sigma}^2 + \varepsilon} e_p(n)\mathbf{x}_p(n).$
End
```

III. FPGA IMPLEMENTATION

This section describes the adaptation of the PEMAFC algorithm on a realizable hardware platform suitable for automotive environment. Considering the PEMAFC based Feedback Cancellers a subcomponent of a speech reinforcement system including a noise cancellation component as well as a voice activity detector, the target platform must be cost effective with a relatively high performance DSP. The Xilinx Automotive (XA) Spartan®-6 families of FPGA was used [14].

A. Design Process

The PEMAFC algorithm was originally developed as MATLAB scripts using high precision floating-point arithmetic. The SoNoScout NVH Binaural recording and analysis system 653A and Sound Level Meter 2250L were also used. A one-to-one conversion to an equivalent FPGA implementation cannot be directly or easily implemented with reasonable resource utilization. Also, the precision of data in the FPGA implementation is limited to a fixed number of bits (fixed-point representation) which results in the addition of quantization errors to the system.

The first step of the implementation consists of identifying the parts of the algorithm, which became the main blocks of the hardware. A fixed point version of these blocks has been implemented using Fixed-Point Toolbox™ in Matlab® [15] and then compared to the floating point version.

Following the Matlab fixed point validation, a Xilinx System Generator^{TM} (XSG) [16] model was developed. Each block has been individually validated by passing the data to/from Matlab workspace as well as using the "scope" block in the model by connecting the signal of interest to it. Once the XSG model has been created it has been validated against Matlab implementation. Finally, the design has been synthesized using the Xilinx ISE 13.4 design suite and run on the FPGA target in the "hardware-in-the-loop co-simulation".

B. The Hardware implementation

The implementation of the PEMAFC algorithm is done on a Spartan6 FPGA, i.e., the XC6SLX45[15]. The system clock frequency is approximately 100 MHz and the sampling frequency is 16 kHz; consequently, there are approximately 6250 clock periods available between two successive samples.

The flow of the algorithm is described in Fig.3. In the first phase, the output of the adaptive MAC filter $y(n)$ is generated and the error signal $e(n)$ is computed. In the second phase a fixed order pre-whitening of error signal $e(n)$ and its delayed and amplified version $x(n)$ is realized based on (8). A power estimate (10) of both decorrelated error and reference signals is generated in the third step. In the fourth step the algorithm's step size is computed based on the previously power estimated values (9). The last phase is dedicated updating the adaptive filter's coefficients based on their past values, decorrelated reference and error signals $\mathbf{x}_p(n)$ respectively $e_p(n)$ and the previously computed step size. The FPGA implementation of the PEMAFC algorithm requires a few RAM memory blocks, as follows.

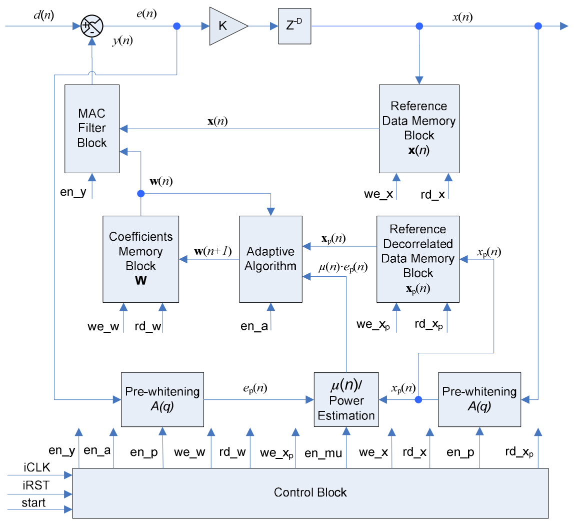

Figure 3. Implementation scheme of the PEMAFC

The first one is associated with the reference signal samples; this memory can be viewed as $L\times1$ matrix. The second one is associated with the decorrelated reference signal samples with the same depth as the previous one. The third memory block is used to keep the filter coefficients.

Only one division is associated with the PEMAFC algorithm (9). The implementation was based on the Xilinx Divider Generator 3.0 block (radix 2 non-restoring division version) instead of the CORDIC Divider block, which is much more resource consumer. For a better precision, the division has been done between $e_p(n)$ and the power estimation $\hat{\sigma}$ (both operands being reinterpreted as signed integers), the result being scaled by $\log_2(L)$.

By selecting $\bar{\mu}$ as power of two, only four multipliers cells are used to implement the PEMAFC algorithm. Two of them are used in a pipelined manner, i.e., series of L computations (one for updating the filter coefficients and the other for computing the output of the FIR filter in a multiply with accumulate mode). The other two are involved in the power estimation of $e_p(n)$ and $x_p(n)$ signals. The input and the output signals $d(n)$ and $x(n)$ are represented on 16 bits (Q15 representation) while the internal representation at different stages varies based on the dynamic range of internal "signals", in order to avoid the overflows and to minimize the quantization errors.

IV. RESOURCE USAGE AND SIMULATION RESULTS

The functional simulations have been done using FPGA target in the hardware-in-the-loop co-simulation mode as in the Fig.4. The real signal was read from Matlab workspace and it was either a voice signal with an additive white Gaussian noise (SNR=30dB) or an auto-regressive noise generated by passing a white Gaussian noise through a 10 order AR system.

The sampling frequency was 16kHz and the adaptive algorithm parameters used are: L = 256 (the same length as feedback path); fix step size $\bar{\mu} = 2^{-6}$; regularization factor

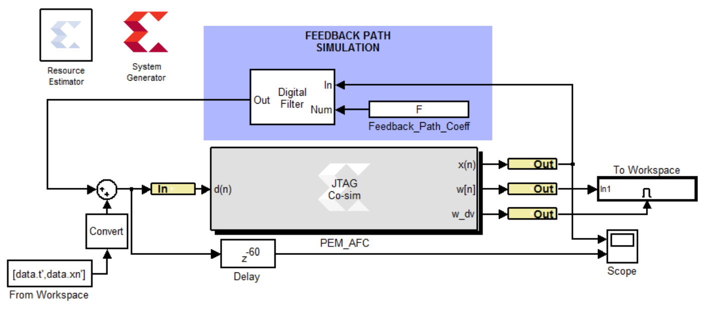

Figure 4. Hardware-in-the-loop co-simulation of the PEMAFC

ε=10^{-3}; weighting factor for power estimation $\lambda = 0.9961$; the parameter of fixed model $\alpha = 0.91$.

The gain and the delay of the forward path are K=10dB and D=60 samples (3.75ms) respectively. The feedback path was simulated using Simulink FIR Filter block.

The performance measure was the normalized misalignment (in dB), defined as $20\log_{10}\|\mathbf{f} - \hat{\mathbf{f}}(n)\|_2 / \|\mathbf{f}\|_2$, where **f** is the true impulse response of the feedback path and $\|\bullet\|_2$ denotes the l2 norm. The effect of the quantization error on the AFC performance is shown in Fig. 5.

Table II shows the resource requirement of the FPGA implementation as reported by the Xilinx ISE Foundation.

V. CONCLUSIONS

In this paper, a sequential time based, PEMAFC algorithm is implemented on FPGA using Xilinx System Generator. Comparable results with infinite precision version have been obtained. Resource analysis shows the design uses only 15% of the total available general logic resources making possible an integration with other in-car sub-systems on a single FPGA. The implementation of adaptive estimate of signal model will be considered in the future work.

ACKNOWLEDGMENT

This work was supported under the Grant POSDRU/107/1.5/S/76813 and Grant CNCS-UEFISCDI PN-II-ID-PCE-2011-3-0097.

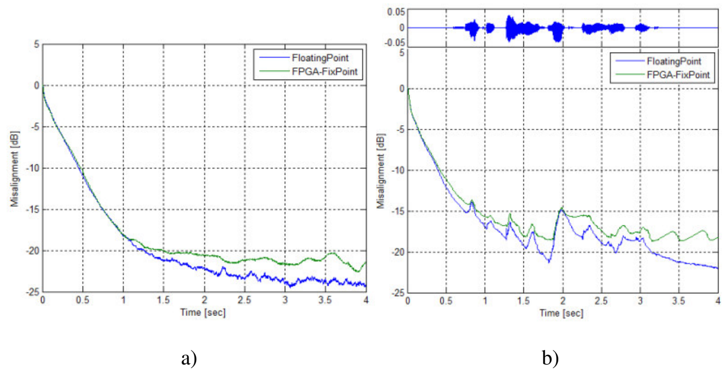

Figure 5. Misalignment of the PEMAFC (finite and infinite precision). a) auto-regressive noise; b) voice signal.

TABLE II. RESOURSE UTILIZATION FOR PEMAFC ALGORITHM WITH FIXED LP ORDER.

Available Resources (total)	Used Resources
Slices (6822)	896 (15%)
FFs (54.576)	3100 (5%)
4-LUTs (27288)	2895 (10%)
RAMB8B (323)	3 (1%)
DSP48A1s (58)	4 (6%)

REFERENCES

[1] E. Hänsler and G. Schmidt, "Acoustic Echo and Noise Control: A Practical Approach," John Wiley & Sons, New York, NY, USA, 2004.

[2] J. Chang and J.R. Glover, "The feedback adaptive line enhancer: a constrained IIR adaptive filter," IEEE Trans. Signal Process., vol. 41, Nov. 1993, pp. 3161–3166.

[3] P. Gil-Cacho, T. van Waterschoot, M. Moonen, and S. H. Jensen, "Regularized Adaptive Notch Filters for Acoustic Howling Suppression," Proceedings of 17th European Signal Process. Conf. (EUSIPCO '09), pp. 2574–2578.

[4] L. Ljung, "System Identification: Theory for the User," Prentice Hall PTR, 1998.

[5] U. Forssell, "Closed-loop Identication: Methods, Theory, and Applications", Dissertations No. 566, Linköping 1999.

[6] J. Hellgren and U. Forssell, "Bias of feedback cancellation algorithms in hearing aids based on direct closed loop identification," IEEE Trans on Speech and Audio Processing, vol. 9, Nov. 2001, pp. 906–913.

[7] J. Hellgren, "Analysis of feedback cancellation in hearing aids with filtered-X LMS and the direct method of closed loop identification," IEEETrans. Speech Audio Process., vol. 10, Feb. 2002, pp. 119–131.

[8] A.Spriet, I. Proudler, J. Wouters, and M. Moonen,"Adaptive feedback cancellation in hearing aids with linear prediction of the desired signal," IEEE Trans on Signal Processing, vol. 53, Oct. 2005, pp. 3749-3763.

[9] Ann Spriet, S. Doclo, Marc Moonen, and Jan Wouters, "Feedback Control in Hearing Aids," in Springer Handbook of Speech Processing, pp. 979–999. Springer Verlag, 2008.

[10] M. Rotaru, F. Albu, and H. Coanda, "A variable step size modified decorrelated NLMS algorithm for adaptive feedback cancellation in hearing aids," in Proc. ISETC, Timisoara, Romania, 2012, pp. 263–266.

[11] A. Ortega, E. Lleida, E. Masgrau, L.Buera, and A. Miguel "Acoustic Feedback Cancellation in Speech Reinforcement Systems for Vehicles" INTERSPEECH 2005 - Eurospeech, 9th European Conference on Speech Communication and Technology, ISCA, Sep. 2005, pp. 2061-2064.

[12] S. Cifani , L. C. Montesi, R. Rotili, E. Principi, S. Squartini, and F. Piazza, "A PEM-AFROW based algorithm for Acoustic Feedback Control in Automotive Speech Reinforcement Systems," Proceedings of 6th International Symposium on Image and Signal Processing and Analysis, Sep. 2009, pp. 656–661.

[13] J. E. Greenberg, "Modified LMS algorithms for speech processing with an adaptive noise canceller," IEEE Trans. Speech Audio Process., vol. 6, no. 4, Jul. 1998, pp. 338–350

[14] Xilinx Inc., "XA Spartan-6 Automotive FPGA Family Overview, " DS170 (v1.2), Dec. 2011.

[15] MathWorks, "Fixed-Point Toolbox," http://www.mathworks.com/products/fixed, [retrieved: March, 2013].

[16] Xilinx Inc., "Xilinx System Generator for DSP," http://www.xilinx.com/tools/sysgen.htm, [retrieved: March, 2013].