Oversampled Active Noise Control Headphones

André D. Barbosa
Electrical and Computer Engineering
Instituto Superior Técnico
Lisbon, Portugal
andre@trimagno.com

***Abstract*—A prototype Active Noise Control (ANC) headphone with oversampling was developed in this project. To develop the prototype, a ANC headset was used, with its controller removed and another developed based on a FPGA board. Additionally, high-performance converters were needed to be able to support oversampling. In order to use oversampled signals, signal processing techniques were used, namely, the use of Sigma-Delta converters. Thus, it was possible to implement the system with a high sampling frequency, yet utilizing fewer bits per sample, hence reducing the computational complexity of the system. FxLMS algorithms with feedfoward and feedback were implemented with the secondary path modeling performed in offline mode. To find out if the secondary path estimation was being performed correctly, several tests were performed. Lastly, the performance of the FxLMS system with feedback was examined for various types of noise. It was found that for sinusoidal noise the system was able to perform a full cancellation, leaving only ambient noise and high frequency noise that is not audible to the human ear. However, for white noise, the system can only reduce about 3% of the total signal power.**

***Index Terms*—Active Noise Control, Sigma-Delta Converters, FxLMS, ZedBoard, Feedfoward Controller, Feedback Controller**

I. Introduction

The technique for active noise cancellation consists of using a cancellation source, which generates a signal with phase opposition and amplitudes equivalent to that of the noise source [3], thus resulting in a cancellation (Figure 1).

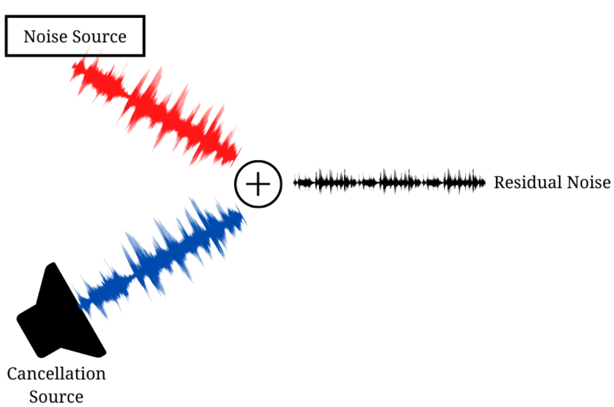

Fig. 1. Visual representation of Active Noise Reduction

For this project, one Active Noise Control (ANC) technique will be applied to a product commonly used in consumers daily life, namely in headphones. For the ANC feedfoward system a reference microphone will be used inside the headphones to measure the noise signal, which will be inverted and inserted into the cancellation source, which in this case will be the headphone speaker itself, in order to create an anti-noise wave [1]-[3]. To give the controller feedback that if the system is canceling correctly, a microphone was used in front of the speaker.

However, it is important to consider that there is a minimum delay margin between the reference signal and the error microphone (Primary Path), and between the sound propagated by the loudspeaker and the error microphone (Secondary Path), which requires that calculations concerning the anti-noise signal have to be performed in this short interval. Thus, to achieve significant noise reduction, it is necessary to implement a ANC with the shortest possible delay.

This delay cannot be greater than the difference between the PP delay (dp) and the SP delay (ds), and this delay can be small. Also, the delay in the ANC controller is equal to the sum of the delay of the reference sensor, converters and all the digital delays. To reduce this delay, high sampling frequencies can be used, but this increases the computational complexity and power consumption of the controller.

To reduce the computational complexity and power consumption of using a high sampling rate, signal processing techniques were used, namely, the use of Sigma-Delta converters, since with the use of these, the number of bits per sample are reduced.

II. Background

In this chapter theoretical concepts are introduced that are necessary to understand the methodology used in this project.

A. Adaptive filtering

Unlike ordinary (non-adaptive) filtering, adaptive filtering does not consist of a system defined in advance, but instead uses the information around it to estimate the value of the parameters, which control the linear filter transfer function, present in the adaptive filtering system [5].

The mode of operation of this type of filtering essentially involves two processes, namely the filtering process, in which a result is produced in response to a sequence of input data, and the adaptive process, which provides a control mechanism for the parameters used in the preceding process [6].

The structure of the adaptive filter (Figure 2) is composed of a digital linear filter, whose coefficients will vary with the adaptive algorithm, and input, output, reference, and error signals. The adaptive process is controlled by the error signal, which represents a measure of the adaptation of the filter parameters, thus indicating the level of conformity between the output and the reference signal, which is the desired response of the adaptive filter [5]. The main goal of adaptive filtering is then to optimize the error signal, that is, to minimize the difference between the desired signal and the output signal of the filter, through a constant adaptation of the parameters present in the filter [5].

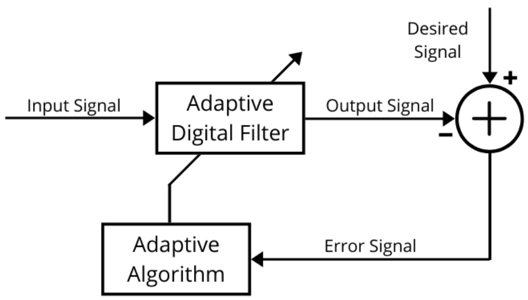

Fig. 2. Adaptive Filter Structure

1) Digital Linear Filters: The linear filter can be realized using finite impulse response (FIR) or infinite impulse response (IIR), however the most widely used adaptive filter structures are FIR, due to their unconditional stability and the relatively simple analysis of the properties of these filters [5], and due to these factors only FIR filters will be explored in this project.

Unlike IIR filters, FIR filters only depend on the previous inputs, and have the following equation

$$y(n) = \sum_{k=0}^{M-1} a_k * x(n-k) \tag{1}$$

Where M is the number of filter coefficients, a_k are the coefficients, $x(n)$ is the input signal and $y(n)$ is the output of the filter.

Since this type of filter only depends on the prior inputs, its transfer function only has zeros, which indicates that the filter is always stable. FIR filters also have a low sensitivity with respect to quantization errors of the coefficients [8].

Of all the existing FIR filter structures, the most intuitive would be to perform the multiplications first and add them to an accumulator (Direct Form: 3), however, there is another structure of FIR filters: the transposed.

In theory, both structures will give equivalent results, but when calculated with finite precision, there may be differences between the different implementations.

In this project it will be used the direct form because it has better numerical properties, namely, it is possible to use a buffer with a smaller number of bits. With a reduced FIR implementation it is possible to perform all multiplications in parallel and then perform the necessary sums.

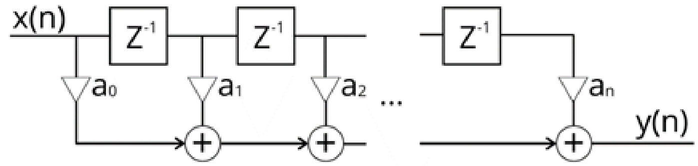

Fig. 3. FIR Filter Direct form

2) Adaptive Algorithms: The adaptive algorithm can take many forms, and is usually derived as a form of optimization, responsible for minimizing the error criterion, useful for the task at hand [9]. The goal of the adaptive algorithm is to set the adaptive filter coefficients so that the output minimizes an objective function, and thus the output tends toward the desired signal [10].

The least mean squares (LMS) procedure uses a Stochastic gradient method, in which the FIR filter coefficients, are updated based on the instantaneous error signal [5]. Where the error signal at discrete time n is represented by:

$$e(n) = d(n) - y(n) \tag{2}$$

$$e(n) = d(n) - \mathbf{x}^{\mathbf{T}}(n)\mathbf{w} \tag{3}$$

The parameter d(n) corresponds to the desired signal, y(n) is the output signal of the digital filter, $\mathbf{x}^{\mathbf{T}}(n)$ is the transposed vector containing the input signals, and W is the vector containing the filter coefficients. Thus, the LMS cost function is represented by:

$$C(n) = E[e^2(n)] \tag{4}$$

In other words, the cost function is equal to the expected value of the squared error, and this cost function is called the mean square error (MSE). The graphical representation of the cost function will have the shape of a ”bowl”, which will be called the performance surface. The minimum point of the performance surface will have the coefficients that minimize this function, and there are no local minimum, but only a global minimum.

Many useful adaptive processes that cause the coefficients of digital filters to trend toward performance surface minimum do so by gradient methods [11]. These take advantage of a mathematical property whereby when the gradient of a convex function is zero, it means that point is a global minimum. So, to find the coefficients that minimize the cost function, just calculate the gradient of that function and set it equal to zero, as represented in the following equations.

$$\mathbf{R} = E[\mathbf{X}(n)\mathbf{x}^{\mathbf{T}}(n)], \mathbf{P} = E[d(n)\mathbf{X}(n)] \tag{5}$$

$$\nabla C(n) = 2\mathbf{RW} - 2\mathbf{P} \tag{6}$$

$$\mathbf{W}^* = \mathbf{R}^{-1}\mathbf{P} \tag{7}$$

The variable $\mathbf{R}$ relates to the self-correlation matrix of the input signal, $\mathbf{P}$ to the cross-correlation vector of the input and reference signal, and W^* to the coefficients that minimize the performance surface. However, in many applications, the parameters of this quadratic performance surface are unknown, and an analytical description of it is not available.

A well-known method, which overcomes this issue, is steepest descent, it is an iterative gradient search method that also causes all components of the coefficient vector to change at each iteration [11]. The iterative gradient search procedure can be represented algebraically as

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \nabla C(n) \tag{8}$$

Where n is the iteration number, $\mathbf{w}(n)$ a vector that consists of the filter coefficients at "present" time, while $\mathbf{w}(n+1)$ is the new value of the coefficients, μ is the step size, and $\nabla C(n)$ is the gradient of the cost function with respect to $\mathbf{w}(n)$. In the stochastic gradient method, C(n) is approximated by $e^2(n)$, obtaining the following expression

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\mathbf{X}(n) \tag{9}$$

For this algorithm to be stable, that is to converge to the desired coefficients, the step size must be contained between 0 and $2/trace(\mathbf{R})$. The value $trace(\mathbf{R})$ represents the sum of the main diagonal elements (from top left to bottom right) of $\mathbf{R}$ [6]. The LMS algorithm is simple, and yet capable of achieving satisfactory performance under the right conditions. Its main limitation is a relatively slow convergence rate [6].

Variants of the LMS algorithm, such as the leaky LMS version, have been proposed to deal with some problems of the LMS algorithm, such as when there is no input signal or when a sinusoidal signal without noise is used as input, the LMS algorithm can result in divergence of the coefficients.

This is because the weight update stops, i.e., the weight increases are very small, and finite precision effects can cause the unconstrained weights to grow unboundedly, resulting in overflow during the weight update process. Introducing leakage into the LMS algorithm stabilizes the system, this algorithm is based on adding a parameter to the update equation, referred to as the leakage factor, which tends to bias each filter weight towards zero.

Leakage is known to result in distortions in the coefficients and, as such, must be kept at low values. Leaky LMS, changes the cost function, thus resulting in the following equations, where the leakage factor corresponds to λ.

$$C(n) = e^2(n) + \gamma \mathbf{w}^{\mathbf{T}}(n)\mathbf{w}(n) \tag{10}$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\mathbf{x}(n) - 2\mu\gamma\mathbf{w}(n) \tag{11}$$

$$\mathbf{w}(n+1) = (1-\lambda)\mathbf{w}(n) + \mu e(n)\mathbf{x}(n) \tag{12}$$

Where $\lambda = 2\mu\gamma$.

B. Active Noise Canceling

Active noise control systems provide sound attenuation by introducing a second electronically generated sound wave into the acoustic environment. If the amplitude of the second sound wave is equal to that of the first, but the phase reversed, then the two will be canceled [18]. One of the focuses of this project is a specific type of controller, which is the most common among researchers and developers, and called feedforward adaptive controller, whose structure is depicted in Figure 4. The adaptive feedback controller will also be a topic of study, where the main difference is that it does not contain a reference signal, but rather it is calculated from the error signal.

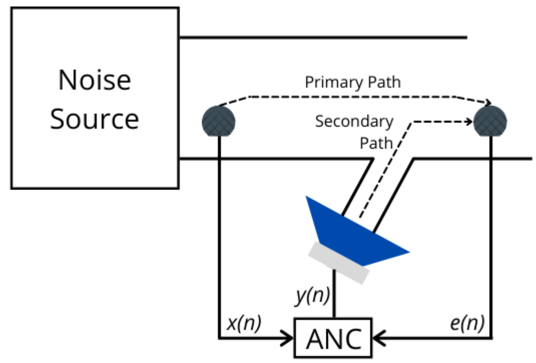

Fig. 4. ANC feedforward system

For feedforward systems the noise signal received at the reference microphone is represented by x(n), y(n) is the linear filter output signal, and e(n) is the error signal obtained at the error microphone. The presence of the secondary path has led to the existence of new adaptive algorithms such as the "Filtered-x LMS" algorithm (FxLMS), which is the adaptation for ANC LMS algorithm [19].

In this algorithm, the input signal samples are filtered by the secondary path transfer function estimate. This filtered set of signals is then multiplied by the error signal to produce the gradient estimate used to modify the current weight values so that the disturbance attenuation levels are improved [18].

To be able to filter the inputs with the transfer function of the secondary path, it is first necessary to estimate it ($\hat{S}$). This can be done in two ways, offline or online, and in offline mode the estimation occurs at system startup, since the secondary path practically does not vary with time [19].

To perform secondary path estimation in offline mode, the control system has to generate white noise and send to the headset [19]. Thus, the error signal will the white noise after passing through the secondary path. With this, it is possible to use the LMS method to estimate the secondary path.

In an ANC feedback system the reference microphone is not used, so no reference signal is available. An advantage of this system is that it avoids the acoustic feedback problem belonging to two-microphone feedforward systems, and also reduces the hardware required (ADCs, microphones and amplifiers). One disadvantage is that the reference signal has to be estimated, so this signal may contain noise. For the estimation of the reference signal we have:

$$u(n) = e(n) - \mathbf{y}^{\mathbf{T}}(n)\hat{\mathbf{s}}(n) \tag{13}$$

$$u(n) = d(n) + \mathbf{x}^{\mathbf{T}}(n)\mathbf{w}\mathbf{s} - \mathbf{x}^{\mathbf{T}}(n)\mathbf{w}\hat{\mathbf{s}} \tag{14}$$

If the estimation of the secondary path is done well then the terms cancel out, resulting:

$$u(n) = d(n) \tag{15}$$

The feedforward system performs better since it does not have to rely on the estimate of the reference signal, but rather gets the desired signal from the reference microphone with no estimate, which if miscalculated in the feedback system the system may not converge or even diverge.

C. Signal Processing Techniques with Oversampling

For ANC systems to be able to achieve a significant amount of noise reduction, it is necessary to implement the controller with a small delay. However, in most ANC applications, this delay is significant due to the typically low sampling rate, anti-aliasing (AA) and reconstruction filters (RC) of the AD and DA converters.

One technique to decrease this delay is to increase the sampling frequency, but with the consequence of a significant increase in computational complexity.

The use of sigma-delta converters, although these work at high sampling frequencies, use AA and RC filters, which introduce a large delay, and consequently makes their use in ANC limited. One way to overcome this problem would be to completely remove the AA and RC filters, thus working with the oversampled signals in the controller, and with the use of this type of converters it is possible to decrease the number of bits used per sample thus decreasing the computational complexity [7].

Normally the signal transfer function (STF) is unitary and the noise transfer function (NTF) can vary with design needs. However, for this project, we used a simple NTF that is given as $(1-z^{-1})^P$, where P is the order of the SDM. Resulting in the following equation:

$$Y(z) = U(z) + (1 - z^{-1})^P E(z) \tag{16}$$

Where U(z) represents the input of the SDM, Y(z) the output of the SDM, and E(z) the error that the quantizer introduces into the system, this type of SDMs has a relatively simple implementation.

Regarding the choice of quantizer, the simplest and historically oldest method is to use single-bit quantization. However, there are great advantages to employing a multi-bit quantizer, since stability is improved and the quantization error is reduced 6dB for each bit added in the quantizer resolution.

The common application of SDMs is to use frequencies higher than the sampling frequency of the system, to decrease quantization noise. However, for this project, since the sampling frequency is already high, it will not be necessary for SDMs to work at a higher frequency, since between samples their values do not vary much in the audible frequency band.

The NTF of the SDM represents a high-pass filter function, which is depicted by Figure 5. This suppresses the error at frequencies around 0, but the NTF function also increases the error at higher frequencies. As the order of the SDM increases,

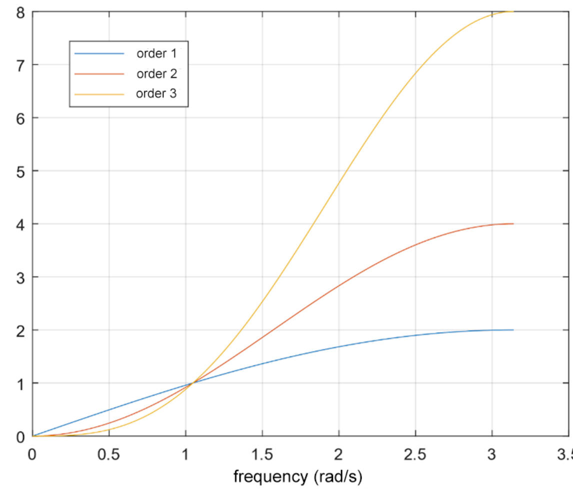

Fig. 5. Noise Modeling Curves and Noise Spectrum in First, Second and Third Order SDM

it can be seen that the error is suppressed more strongly at low frequencies, but is stressed even more at high frequencies.

This error caused at high frequencies by the SDM will spill over to the LMS algorithm, which is no problem as long as the sampling frequency is high enough so that the error is not present at frequencies audible to humans.

III. State of the Art

Modeling unknown systems using adaptive filtering has many applications for practical problems. One of the applications is in the area of communications, for example, in the transmission of information over long distances (antennas). In this type of communication, there are several paths (multipath) from the transmitter to the receiver (i.e. reflections from the ground), which can cause interference and echoes in the received signal. One way to overcome this problem would be to model the multipaths with an adaptive filter and thus at the receiver it is possible to remove the interference and echo in the received signal [11].

Short Word length (SWL) filters, in which signals and filter coefficients are stored using a reduced number of bits, have been explored since the early 1980s [22]. First, ternary FIR filters were proposed, which take advantage of a fixed coefficient that can only take the value -1, 0, or 1 [23]. Even though 1-bit signals have demonstrated applicability in real-time control, and a significant reduction in computational complexity [24], adaptive SWL filters (such as coefficients in 2-bit format), have been found to outperform the others (1-bit and ternary) [22]. Thus, it is essential that there is a trade-off between hardware efficiency and performance [25], which implies that a SWL implementation produces better performance in exchange for higher computational complexity [26].

The sigma-delta system, has proven not only to be more efficient at noise reduction than a standard modulator [27], but also capable of achieving higher resolution compared to analog circuits [28], and has therefore gained great popularity in audio signal processing as an effective method for building high-resolution ADCs and DACs [21][24].

The implementation of the aforementioned filters and other sigma-delta based techniques in FPGA were examined. It is recognized that this type of technology will provide more and more solutions to signal processing problems, as well as a mechanism to work with important variables such as the increasing demand for better performance, since they are able to maintain the flexibility of software-based solutions while providing high levels of performance [30].

Despite extensive research on the topic, and several headsets with different designs have already been implemented, such as in-ear noise-canceling headsets with the ASIC controller [34], there are still opportunities for improvement, namely in reducing costs, and investigating alternative noise modeling techniques to reduce system components [35], mainly because there is continuous pressure to achieve better performance with lower power consumption in a smaller area [36].

IV. Methodology

The ANC system of the prototype will be implemented on a ZedBoard FPGA, where the microphone and speaker signals will have to pass through high-performance converters.

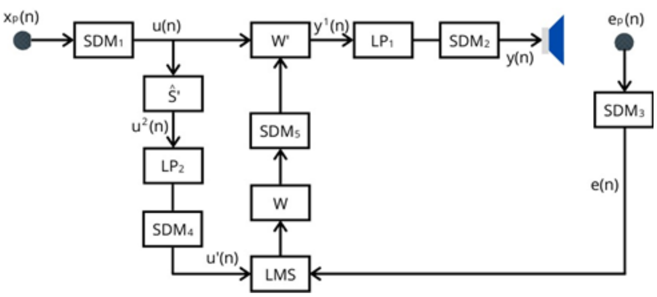

Fig. 6. ANC system with feedforward FxLMS algorithm and SDMs

SDM blocks will be used to reduce the number of bits in general from B bits (16 bits) to C bits (5 bits), thus lowering the computational complexity and be able to use a high sampling frequency.

The SDM_5 block converts W, which is the result of updating the FIR filter coefficients (B bits), into W', which corresponds to the coefficients used in the FIR filter, but with a smaller number of bits (C bits). To reduce computational complexity, the conversion is performed on only one coefficient in each sampling period.

The filter $\hat{S}$' represented using only C bits, is obtained by quantizing the estimate of $\hat{S}$. This estimation always happens at system startup using the offline method, as mentioned above. Since $\hat{S}$ has a limited number of coefficients, this number has to be well refined, since the estimation has to present a frequency and amplitude response very similar to the real secondary path, so if the number of coefficients is low, a good estimation of the secondary path may not be achieved, and if they are too many there is an unnecessary use of resources.

In the case of u^2(n) and y^1(n), since these signals have already been processed by SDMs, they are the input of the SDM_2 and SDM_4 blocks, in which case the high frequency quantization noise also dictates the maximum input level. Thus, the LP block (low pass filter) reduce the high frequency quantization noise before SDM_2 and SDM_4, thus allowing them to reduce their quantization steps.

Another system that will be covered in this project is the FxLMS with feedback, where the reference signal is calculated from the error signal, thus obtaining the following block diagram.

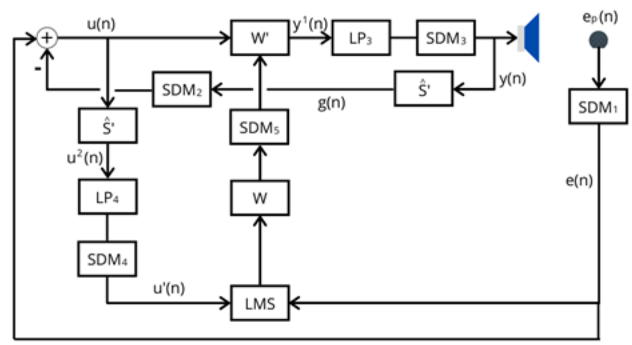

Fig. 7. ANC system with feedback FxLMS algorithm and SDMs

A disadvantage to this method is having to estimate the reference signal while the feedforward system uses hardware to obtain it, which implies that if the secondary path is only a D-sample delay, then the anti-noise signal must have a D-sample lead in order to compensate for the secondary path delay, so for cancellation to occur the following must be true: y(n) = d(n+D) ⇔ y(n)*S=d(n). This means that it is necessary for the anti-noise signal to be D samples ahead (a predictor is required), and this is only possible at the expense of an increase in high frequency noise, i.e., to reduce noise at low frequencies you have to increase it at high frequencies, this effect is known as the waterbed effect which results from Bode's integral formula [41].

In the feedforward system these problems do not occur, since the primary path will have a longer delay with respect to the secondary path (ignoring the controller delay), so the reference signal will be more than D samples behind compared to the error signal. Thus, it is only necessary for the primary noise signal to have a delay relative to the reference signal.

V. Hardware

The hardware of this ANC system consists of headphones, which contain a pair of microphones and speakers, an analog circuit to amplify their signals, a ZedBoard development board, and AD and DA converters to be able to interpret these signals.

The ZedBoard is a complete development kit for designers interested in high-performance solutions, as the board contains all the necessary interfaces and support functions to enable a wide range of applications. The choice of FPGA over other options was due to the fact that this project requires a high value for the sampling frequency and the need for many resources. Compared to DSPs (Digital signal processor), for high sampling rates, it can have difficulty capturing, processing and outputting the data without any loss. This is due to the many shared resources, buses and even the core within the processor. The FPGA, however, can dedicate resources to each function.

The ATH-ANC1 QuietPoint Headphones only contain within each capsule a loudspeaker and an error microphone, and therefore no reference microphone, which only allows the feedback system to be used. However, in order to be able to test the feedforward system, it can be used the error microphone of one of the capsules as the reference microphone and the other capsule for noise cancellation.

The amplification circuit for the microphone and speaker signals was previously developed. Using potentiometers it is possible to adjust the gain of the microphone and the speaker, thus controlling the gain of the secondary path.

To allow the ZedBoard to interpret the microphone signals, the ADS8363EVM analog-to-digital converter was used. It features two channels, which have the ability to sample simultaneously, and can sample signals at a maximum frequency of 1MHz, yielding 16-bit digital signals. The MAX5216PMB1 was used to convert the signal from digital to analog that would be sent to the speaker. This converter takes 16-bit words and converts them into an analog signal with a 2.5V reference.

VI. Software

The development of all the software for this project was carried out in three applications: Vivado, Vitis, where both are from Xilinx, and MatLab. These Xilinx development environments include all the necessary resources for synthesis and analysis of VHDL designs (FPGA programming language) and programming of ARMs and FPGAs.

A digital high-pass filter has been developed because the microphone signal readings are not absolutely zero mean, witch the offset can interfere with the performance of the system, so using a cut-off frequency below 5Hz it is possible to remove the DC component without interfering with the integrity of the microphone signals.

For this the ANC system, the first thing to do is estimate the secondary path, in offline mode and for this it was used the LMS algorithm. For this purpose, a value of 10M samples was set for the estimation, so for a sampling frequency of 1MHz this estimation takes 10 seconds. In this estimation, the coefficients must have a frequency response very similar to the real secondary path for the entire frequency band, so SDMs cannot be used. To obtain a good resolution, 16-bit words were used, and after the estimation, the coefficients were rounded to 5 bits, not causing much disparity in the frequency response.

The delays inserted in the system must be taken into account so that when updating the coefficients the signals are at the same time instant. In the feedback system it is also necessary to pay attention to the estimation of the reference signal, because at the slightest delay of one of the signals, the reference signal is no longer correct.

The low-pass filters (LP blocks), which are before the SDMs, are just moving average filters. This type of filter was chosen, since they require few resources and do not have much computational complexity when using a filter of dimension 4.

An improvement made in the calculation of the output of FIR filters was to use the parallelism allowed by the VHDL language. So, instead of constantly multiplying an input by a coefficient and subsequently adding it to an accumulator, a FIR filter with reduction was performed. That is, all multiplications are performed, placed in a vector, and after it is completely filled, sums are performed two by two in parallel, thus creating a tree of sums.

For the updating of the coefficients, the leaky LMS was used due to the advantages mentioned above. It should be noted that the filter sizes were varied throughout the project, since they depend on the sampling frequency and available resources. Regarding the step size and the leakage factor, both were determined experimentally to obtain better results.

VII. Results

This chapter presents the results of various simulations and tests performed on the hardware to understand the capabilities and limitations of the system. All tests were performed with the environment as similar to reality as possible, with ambient noise and sound reflections from various objects. For these tests, one earpiece is sufficient to evaluate the performance of the system, and for this case the left one was used arbitrarily.

To perform the secondary path estimation it was chosen to use the offline mode, as mentioned earlier. Thus, it is important to evaluate the performance of the system when doing this procedure. To do so, first, the algorithm was tested by performing simulations, with the following parameters: $\mu_{sp} = 7 * 2^{-11}; L_{sp} = 128; F_s = 1MHz$.

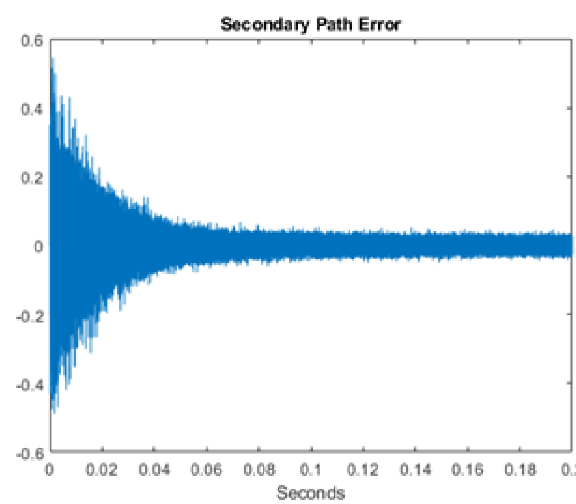

Fig. 8. Error of the estimation of the simulated secondary path

As you can see from Figure 8, the error converges to zero, which means that the coefficients are converging correctly.

However, since we are using a finite number of bits for each word (16 bits) and since the step size is relatively large, there will be quite a bit of residual noise.

The estimation of the secondary path does not have to exactly match the real one, but the phase error cannot be greater than 90°, otherwise the system is prone to instability [19].

Thus, to evaluate whether the estimation was well done, the difference of the frequency response of the real coefficients of the simulation and the estimated coefficients after being rounded was performed, and it was found that from approximately 0.15 F_s, the phase error is greater than 90º, which means that for the proper functioning of the FxLMS system one cannot exceed this frequency. Regarding the magnitude error, this only interferes in the rate of convergence, because the larger the error the longer the convergence time.

To get an idea of the secondary path that the algorithm is intended to estimate, a small program was first made to just send to the headset white noise and to read the values from the microphone. So the LMS algorithm was run in MatLab (64 bit words) to estimate the secondary path for different sampling frequencies, and also with the headphones placed outside the head and on the head. Regarding the placement of the headphones on the ears, there was no great difference in the change of the secondary path, however, the more the sampling frequency was increased, the greater the number of coefficients had to be to be able to estimate it correctly. Thus it was obtained an idea of what the structure of the coefficients would be (figure 9).

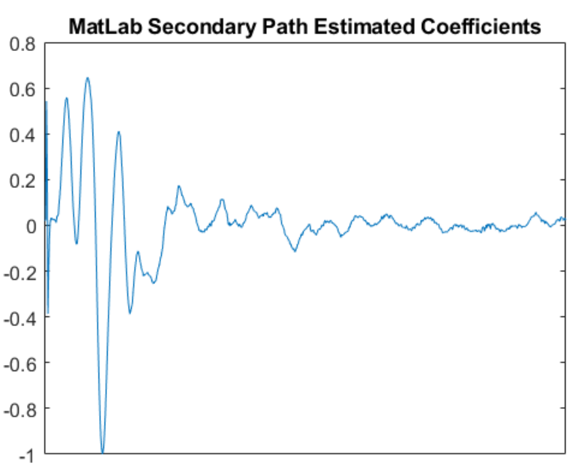

Fig. 9. Secondary path estimation in MatLab

After getting an idea of the values of the secondary path coefficients, the algorithm developed in VHDL for the estimation of the secondary path was tested.

Since the secondary path is controlled by the headphone and microphone gains, these had to be adjusted so that the estimation did not have too low coefficient values but also not too high, since they could saturate. Another reason for adjusting the secondary path was also so that its gain was as close to 0dB as possible, so that the signals in the FxLMS system do not saturate.

To evaluate the performance of the secondary path estimation in Hardware, 10 estimations were performed to visualize the variation in the frequency response of the estimates. For these tests, the parameters used were as follows: $\mu_{sp} = 2^{-11}; L_{sp} = 128; F_s = 100KHz$. A lower sampling frequency than intended was used, since for a higher frequency one would have to have a larger number of coefficients and the ZedBoard does not have enough resources to have more than 128 coefficients together with the FxLMS system.

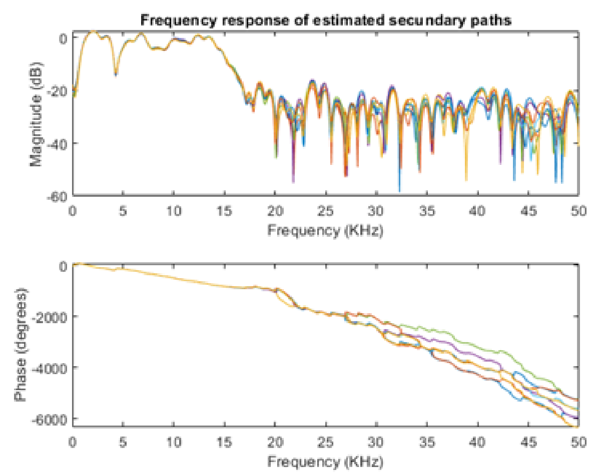

Fig. 10. Frequency response of the various estimates of the secondary path in Hardware

In Figure 10 it is possible to observe the frequency response of the various estimations, and it can be seen that the greatest variation is at high frequencies. With this difference it is possible to conclude that from 20KHz the system can have a phase error greater than 90º. As seen previously in the simulations, the error that the estimation causes is mostly from $0.15F_s$=15KHz, for F_s=100KHz, so it is possible to conclude with great certainty that from 20KHz the system can be unstable.

As a preliminary analysis of the performance of the FxLMS feedback system, we started by looking at the ambient noise that is picked up by the error microphone in order to examine possible anomalies.

It was found that the ambient noise is mostly in the low frequencies and presents two peaks, one at 300Hz with power of -13dB, corresponding to the computer fan, and another at 50Hz with power of 13 dB that comes from the building's mains voltage that works at 50Hz and causes electromagnetic interference in the system. This interference can cause problems in the system, since it is not possible to cancel it with sound signals. So, to reduce this interference, the cutoff frequency of the previously designed high-pass filter was changed to 88Hz, thus filtering out much of the interference.

The parameters used for the results presented below were:

TABLE I
SYSTEM PARAMETERS

Parameter	Symbol	Value
Control Filter Size	L	128
SP Filter Size	N	128
Moving Average LP Filter Size	T	4
Sampling Frequency	f_0	100KHz
SDM order	P	2
Coefficients and Signals Word Length (Most)	C	5
SP LMS Algorithm Step Size	μ_{sp}	$5 * 2^{-11}$
FxLMS Algorithm Step Size	μ_{pp}	$3 * 2^{-11}$
FxLMS Algorithm Leakage	λ	2^{-15}

With the aid of an auxiliary loudspeaker a sinusoidal signal with a frequency of 3KHz was generated and the error signal was observed to see if the system was converging correctly.

Thus, looking at Figure 11, it is possible to conclude that the system is performing noise cancellation. It took about 1 second for it to converge completely, leaving only high frequency noise, in which it is not audible.

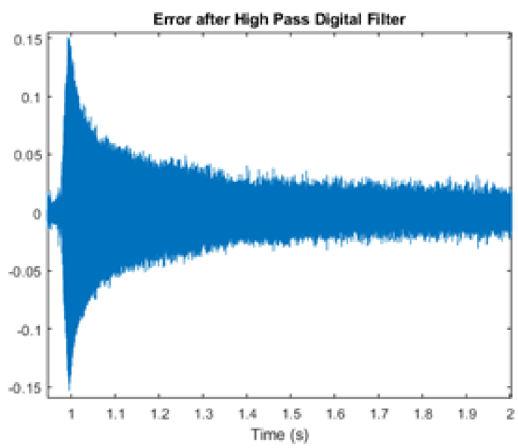

Fig. 11. FxLMS system error feedback signal for 3KHz sinusoidal noise

After verifying that the system was working properly, an evaluation of the amount of sinusoidal noise cancellation was made for several frequencies, for this, it is necessary to realize the difference of signal strength at the noise frequency when the system is deactivated and activated. The first step was to analyze the low frequencies.

Observing Figure 12, it is possible to conclude that the system works with greater efficiency for noises with frequencies above 300Hz, this may be due to the fact that a high-pass filter is being used to filter the signal coming from the microphone, with a cutoff frequency of 88Hz, the secondary path for low frequencies is poorly estimated due to the complexity of estimation of a high-pass filter and due to the plant (secondary path) cut off frequency. For frequencies above 300Hz it is possible to verify that the system can cancel more than 45dB, and looking at the spectrum of the error signal it is possible to verify that the noise was totally canceled at the sinusoid frequency, since there is no trace of the existence of the given

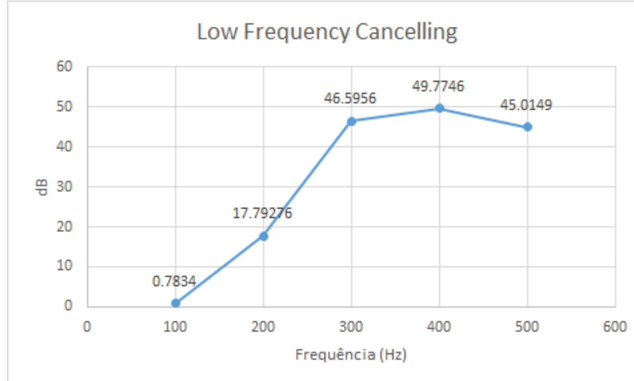

Fig. 12. System performance for low frequency noise

frequency. All the measurements of the cancellation values were only at the given frequencies and not of the total noise.

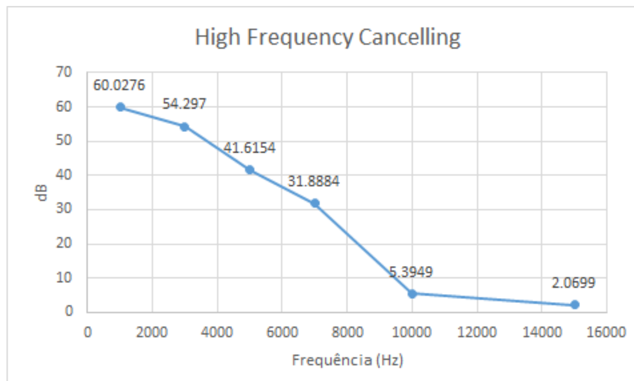

Fig. 13. System performance for high frequency noise

To check up to which frequencies the system had a good performance, Figure 13 was made, which indicates how much decibels the system can cancel at a given frequency. Thus, it is possible to conclude that the system has a good performance up to a frequency of 7KHz. This indicates that the system can cancel with good performance from 300Hz to 7KHz. One would expect the system to cancel up to 20KHz, however, since the sampling frequency is only 100KHz instead of 1.4MHz which was the originally intended sampling frequency, then the SDMs have a higher quantization noise, since the higher the working frequency of the SDM, the lower the quantization noise at low frequencies. Another hypothesis for the system not being able to cancel almost anything after 10KHz would be due to poor estimation of the secondary path, that from 0.15*F_s the phase error is greater than 90º, which may indicate instability, which in this case does not occur, but as there is already a large error it cannot cancel.

It was also tested by setting the speaker to produce soft noise and observing the behavior of the error signal with the system deactivated and activated, as can be seen in Figure 14.

As Figure 14, presents values on a logarithmic scale (dB) it is not possible to have much insight into whether the system within the audible band decreases the noise or increases it,

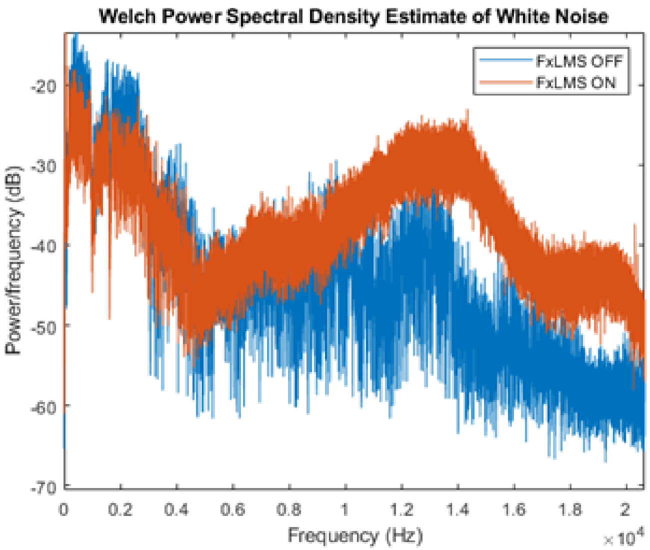

Fig. 14. System performance for high frequency noise

so performing the variance of both signals, it is obtained that the system decreases by approximately 3% the noise power, which is not much, but it was the expected for the FxLMS system with feedback.

VIII. Conclusion

The goal of this project was to implement an oversampled ANC headphone system with an FPGA development board, the ZedBoard housing an FPGA and a dual-core ARM. An analog circuit was also used to amplify the input/output signals to/from the AD and DA converters, which in turn generate signals to/from the ZedBoard. This system was then used to experiment with various system configurations to achieve good attenuation, stability and robustness.

FxLMS was the only algorithm to be developed. The active noise cancelling headphone system with feedback settings showed satisfactory attenuation. However, theoretically, feed-forward systems would perform better than feedback systems, but still both systems have better attenuation compared to some fixed-frame controllers.

The results obtained indicated that this ANC headphone system can be considered for commercialization. However, the stability performances need to be improved before reaching the end user. In this project, a prototype was created, which needs some future research and improvements in order to be sold in the market using its full potential at the lowest possible cost.

References

[1] S. J. Elliott, "Down with noise [active noise control]," IEEE spectrum, vol. 36, nº 6, pp. 54-61, 1999.

[2] C. Y. Chang, A. Siswanto, C. Y. Ho, T. K. Yeh, Y. R. Chen e S. M. Kuo, "Listening in a noisy environment," IEEE Consumer Eletronics Magazine, vol. 5, nº 4, pp. 34-43, 2016.

[3] S. M. Kuo e D. R. Morgan, "Active noise control: a tutorial review," Proceedings of the IEEE , vol. 87, nº 6, pp. 943-973, 1999.

[4] A. Zaknich, Principles of adaptive filters and self-learning systems, Springer Science & Business Media, 2005.

[5] B. Kovacevic, Z. Banjac e M. Milosavljevic, Adaptive Digital Filters, Springer-Verlag, 2013.

[6] S. Haykin, Adaptive Filter Theory, 3ª ed., Prentice-Hall, 1996.

[7] P. W. Wong e R. M. Gray, "FIR Filters with Sigma-Delta Modulation Encoding," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 38, nº 6, pp. 979-990, 1990.

[8] L. Litwin, "FIR and IIR digital filters," IEEE Potentials, vol. 19, nº 4, pp. 28-31, 2000.

[9] V. K. Madisetti e D. B. Williams, Digital Signal Processing Handbook, Atlanta, Georgia: CRC Press, 1999.

[10] P. S. R. Diniz, Adaptive Filtering Algorithms and Practical Implementation, Rio de Janeiro, Brazil: Springer, 2013.

[11] B. Widrow e S. D. Stearns, Adaptive Signal Processing, New Jersey: Prentice-Hall, 1985.

[12] S. S. Haykin, Adaptive Filter Theory, Prentice Hall, 2002.

[13] P. A. C. Lopes, G. Tavares e J. B. Gerald, "A NEW TYPE OF NORMALIZED LMS ALGORITHM BASED ON THE KALMAN FILTER," IEEE Xplore, vol. 3, p. 4, 2007.

[14] D. T. M. Slock, "On the Convergence Behavior of the LMS and the Normalized LMS Algorithms," IEEE TRANSACTIONS ON SIGNAL PROCESSING, vol. 41, nº 9, p. 15, 1993.

[15] K. Mayyas e T. Aboulnasr, "Leaky LMS algorithm: MSE analysis for Gaussian data," IEEE Transactions on Signal Processing, vol. 45, nº 4, pp. 927 - 934, 1997.

[16] H. C. Woo, "Variable Step Size LMS Algorithm using Squared Error and Autocorrelation of Error," Procedia ENgineering, vol. 41, pp. 47-52, 2012.

[17] R. H. Kwong e J. E. W., "A varible step size LMS algorithm," IEEE Transactions on Signal Processing, vol. 40, pp. 1633-1642, 1992.

[18] S. D. Snyder, Active Noise Control Primer, South Australia: Springer, 2000.

[19] S. M. Kuo e D. R. Morgan, Active Noise Control Systems: Algorithms and DSP Implementations, New York: Wiley-Interscience, 1996.

[20] P. A. C. Lopes e J. A. B. Gerald, "Low Delay Short Word Length Sigma Delta Active Noise Control"

[21] R. Schreier e G. C. Temes, Understanding delta-sigma data converters, IEEE Press, 2004.

[22] T. Memon, P. Beckett e A. Z. Sadik, "Sigma-Delta Modulation Based Digital Filter Design Techniques in FPGA," ISRN Eletronics, vol. 2012, pp. 1-10, 2012.

[23] T. Memon, P. Beckett e Z. M. Hussain, "Design and Implementation of a Ternary FIR Filter using Sigma Delta Modulation," International Symposium on Computing, Communication, and Control, vol. 1, 2009.

[24] X. Wu e R. Goodall, "One-bit processing for real-time control," European Control Conference, pp. 3347-3352, 2003.

[25] A. C. Thompson, P. O'Shea, Z. M. Hussain e B. R. Steele, "Efficient single-bit ternary digital filtering using sigma-delta modulator," IEEE Signal Processing Letters, vol. 11, nº 2, pp. 164-166, 2004.

[26] A. Z. Sadik e Z. M. Hussain, "Short word-length LMS filtering," International Symposium on Signal Processing and Its Applications, pp. 1-4, 2007.

[27] D. A. Johns e D. M. Lewis, "Design and analysis of delta-sigma based IIR filters," IEEE Transactions on Circuirs and Systems II: Analog and Digital Signal Processing, vol. 40, nº 4, pp. 233-240, 1993.

[28] S. S. Abeysekera e K. P. Padhi, "Design of multiplier free FIR filters using a LADF sigma-delta modulator," IEEE International Symposium on Circuits and Systems, vol. 2, pp. 65-68, 2000.

[29] C.-L. Chen e A. N. Wilson, "Higher order sigma-delta modulaiton encoding for the design of efficient multiplierless FIR filters with powers-of-two coefficients," IEEE International Symposium on Circuits and Systems, vol. 4, pp. 2361-2364, 1997.

[30] C. Dick e F. Harris, "FPGA signal processing using sigma-delta modulation," IEEE Signal Processing Magazine, vol. 17, nº 1, pp. 20-35, 2000.

[31] P. N. Samarasinghe, W. Zhang e T. D. Abhayapala, "Recent Advances in Active Noise Control Inside Automobile Cabins: Toward quieter cars," IEEE Signal Processing Magazine, vol. 33, nº 6, pp. 61 - 73, 2016.

[32] K. Mazur, S. Wrona e M. Pawelczyk, "Active noise control for a washing machine," Applied Acoustics, vol. 146, pp. 89-95, 2019.

[33] W. Niu, "Adaptive vibration suppression of time-varyung structures with enhanced FxLMS algorithm," Mechanical Systems and Signal Processing, vol. 118, pp. 93-107, 2019.

[34] H. S. Vu e K. H. Chen, “A low-power broad-bandwidth noise cancellation vlsi circuit design for in-ear heaphones,” IEEE Transactions on Very Large Scale Integration Systems, vol. 24, nº 6, pp. 2013-2025, 2016.

[35] P. N. Samarasinghe, W. Zhang e T. D. Abhayapala, “Recent Advances in Active Noise Control Inside Automobile Cabins: Toward quieter cars,” IEEE Signal Processing Magazine, vol. 33, nº 6, pp. 61-73, 2016.

[36] T. Memon, P. Beckett e A. Sadik, “Power-area-performance characteristics of FPGA-based sigma-delta FIR filters,” Journal of Signal Processing Systems, vol. 70, nº 3, 2013.

[37] J. Blauert, Spatial hearing: the psychophysics of human sound localization, MIT press, 1997.

[38] D. H. Crawford e R. W. Stewart, “Adaptive IIR filtered-v algorithms for active noise control,” Acoustical Society of America, vol. 101, nº 4, pp. 2097-2103, 1997.

[39] Texas Instruments, “TLC5540/TLC5510/TLC5510A Evaluation Module, User’s Guide,” 2002.

[40] M. Integrated, “MAX5214/MAX5216 14-/16-Bit, Low-Power, Buffered Output, Rail-to-Rail DACs with SPI Interface,” 2013.

[41] H. K. Sung and S. Hara, ”Properties of sensitivity and complementary sensitivity functions in single input and single output digital control systems”, Int. J. Control, vol. 48, no. 6, pp. 2429-2439, 1988.