

A FPGA-BASED ADAPTIVE NOISE CANCELLING SYSTEM

Wolfgang Fohl,

Jörn Matthies,

fohl@informatik.haw-hamburg.de joern.matthies@web.de

Bernd Schwarz,

schwarz.be@arcor.de

Dept. of Computer Science
University of Applied Science, Faculty TI
Hamburg, Germany

ABSTRACT

A FPGA-based system suitable for augmented reality audio applications is presented. The sample application described here is adaptive noise cancellation (ANC). The system consists of a Spartan-3 FPGA XC3S400 board connected to a Philips Stereo-Audio-Codec UCB 1400. The algorithms for the FIR filtering and for the adaption of the filter coefficients according to the Widrow-Hoff LMS algorithm are implemented on the FPGA board.

Measurement results obtained with a dummy head measuring system are reported, and a detailed analysis of system performance and possible system improvements is given.

1. INTRODUCTION

For long years the field of audio signal processing was dominated by dedicated programmable digital signal processors (PDSP) based on a RISC chip that contains one or more fast MAC (multiply-accumulate) units. Signal processing is performed in a multistage processing pipeline. In the recent years FPGA systems are continuously displacing PDSP systems due to their greater flexibility and their higher bandwidth, resulting from their parallel architecture [1].

This paper investigates the applicability of a FPGA system as a hardware base for realtime audio processing in a headphone-based augmented reality audio system. Such systems superpose artificial audio signals with the real ambient sound. One of the functions of such a system is to extinguish unwanted sound sources of the real environment, a task known as adaptive noise cancellation (ANC). ANC systems are distinguished by their different goals that lead to different architectures. If *all* ambient sound shall be reduced, a *feedback* system with its simpler architecture may be used. If, as in our case, *single sources* of unwanted sound shall be compensated, a *feedforward* system is required. A feedforward system is characterised by two audio inputs per channel: one *reference signal* input for the sound to be removed, and one *error input* for the sound after the compensation (see figure 1).

The remainder of this section gives an overview of related work and reviews the basic theoretical concepts of ANC systems. System setup and system design is described in the second section. Then the measurement results of the system are presented, followed by a review of the results and an outlook at the further development.

1.1. Related Work

Audio usually plays a subordinate role in augmented reality scenarios as the review articles of Benford et al. [2] or Magerkurth et al. [3] show. Works specifically dedicated to augmented audio systems are published by Härra et al. [4], Karjalainen et al. [5], and Tikander et al. [6]. The last mentioned work describes an audio earphone-microphone system with a functionality similar to the system to be described in this paper.

Kim et al. describe a FPGA system for blind signal separation and adaptive noise cancelling [7]. Tests with real audio signals are reported. Di Stefano et al. [8], Elhossini et al. [9], and Lan et al. [10] have published design studies of the implementation of an ANC system on an FPGA board. In these articles the reported test results are obtained by filtering synthetic signal data stored in the system memory.

Trausmuth et al. developed a FPGA based general purpose audio signal processor and wavetable synthesizer [11, 12].

1.2. Adaptive Noise Cancellation: Basic Concepts

The goal of the system is the *selective* cancellation of disturbing noise without affecting other sounds. For this purpose an adaptive feedforward system has to be designed. The basic system structure is given in figure 1: The primary signal $d(n)$ consists of the superposition of noise signal $s(n)$ and the wanted signal $n(n)$. The reference signal $x(n)$ is noise signal measured at the noise source. The system output signal $y(n)$ is an estimate of the noise signal with

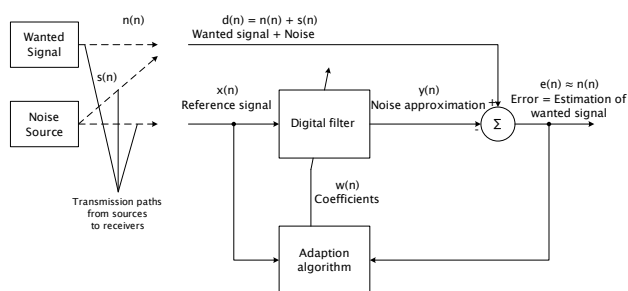


Figure 1: Adaptive feedforward filter structure

inverted sign. In the headphones this signal and the primary signal are superposed, so that the noise signal is cancelled. The error signal $e(n)$ is the result of this superposition. If the adaptive filter does properly model the transmission path from the noise source to the error microphone, the contribution of the noise signal to the error signal is minimised. This goal is achieved by minimising the mean square of the error signal (*MSE adaption*) [13, 14].

Adaptive filters are usually designed as FIR filters due to the fact that these filters are always stable and robust against parameter variations.

1.2.1. The Condition for Optimal FIR Parameters (Wiener-Hopf Equation)

In order to determine an optimal set of filter parameters \mathbf{w} for the minimisation of the error signal, we consider the filter output $y(n)$ of a filter of order $N - 1$ for the sample index n

$$y(n) = \mathbf{w}^T(n)\mathbf{x}(n) \quad (1)$$

where $\mathbf{x}(n)$ is the vector of the N most recent input samples at sampling point n , and $\mathbf{w}(n)$ is the vector of filter coefficients.

The error signal is given by

$$e(n) = d(n) - y(n) \quad (2)$$

Substituting $y(n)$ with the right-hand side of eq. (1) yields

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

and for the squared error we get

$$e^2(n) = d^2(n) - 2d(n)\mathbf{x}^T(n)\mathbf{w} + \mathbf{w}^T\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w} \quad (4)$$

Minimising the expectation value of e^2 under the assumption of a stationary and zero-mean reference signal $x(n)$ finally leads to the *Wiener-Hopf equation*

$$\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p} \quad (5)$$

where \mathbf{R} is the autocorrelation matrix of $y(n)$, and \mathbf{p} is the cross-correlation matrix of $d(n)$ and $x(n)$.

1.2.2. The LMS Adaption Algorithm

The Wiener-Hopf equation (5) is not directly suitable for real-time embedded applications, because \mathbf{R} and \mathbf{p} are neither known in advance, nor are they time-invariant. In addition, the inversion of \mathbf{R} is time-consuming for higher filter orders. The Widrow-Hoff least-mean-squares (LMS) algorithm [15] provides a means of calculating \mathbf{w}_{opt} without the need of knowing \mathbf{R} and \mathbf{p} , and without performing a matrix inversion. In this algorithm the target function for the minimisation is the running average of the squared error signal ($\overline{e^2}$) instead of the expectation value $E\{e^2\}$. In an iterative way the next FIR coefficient set $\mathbf{w}(n+1)$ is computed from the values at step n . A factor μ is introduced to control the step width of the iteration and thus the speed of convergence of the algorithm.

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \nabla e(n)^2 \quad (6)$$

with equation (4) we get

$$\mathbf{w}(n+1) = \mathbf{w}(n) + 2\mu e(n)\mathbf{x}(n) \quad (7)$$

An upper limit for μ is given by

$$0 < \mu \ll \frac{1}{N\bar{P}} \quad (8)$$

where \bar{P} is the mean normalised power of the reference signal $x(n)$.

2. SYSTEM DESIGN

2.1. System Components

Figure 2 shows the block diagram of the ANC system. The system components are a *FPGA board* Spartan 3 FPGA XC3S400, connected to a *stereo audio codec board* Philips UCB 1400, a condenser microphone AKG C3000B as *reference microphone*, an in-ear microphone Soundman OKM-II professional as *error microphone*, an *Audio interface* Motu 828 MKII, Sennheiser HD 600 *headphones*. The audio measurements were performed with a *dummy head measurement system* HEAD HMS II.4.

The in-ear microphone and the headphone were applied to the dummy head as shown in figure 7. Measurements were made by recording the audio signal of the dummy head microphones, which are located at the end of the ear canal.

The noise signal x is recorded with the *reference microphone*, which is connected via the Motu-interface to the left stereo input of the stereo audio codec board UCB 1400. Here it is digitised with $f_s = 48kHz$ sampling rate and routed with a serial protocol to the Spartan FPGA board.

The right stereo channel of the audio codec board is connected via the Motu-interface to the *error microphone*, an in-ear microphone, that records the result of the cancellation, i.e. the error signal e . This signal is utilised for the adaption of the FIR filter coefficients. The output of the adaptive filter y is fed to the headphone via the audio codec board and the Motu interface.

Our FPGA board offers only one interface to an audio codec board, so only two audio signals could be processed simultaneously, hence the noise cancellation could only be performed for one ear.

2.2. Tools

The system model was developed with Simulink, the evaluation of the measurement data was performed with Matlab.

ModelSim SE 6.2c was used for the hardware simulation of the VHDL design of the adaptive FIR filter and the interface to the audio codec. VHDL synthesis and FPGA configuration was done with the Xilinx ISE 9.1i development environment.

2.3. System Model (Simulink)

Preliminary measurements were carried out to determine the group delays in the signal path. The results are indicated in figure 2.

Label	Value	Path
t_1	296 μs	Signal generator output – audio codec input via reference microphone
t_2	1230 μs	Audio codec input – audio codec output (internal)
t_3	1330 μs	Audio codec output – audio codec input via error microphone
t_4	3200 μs	Signal generator output – audio codec input via error microphone

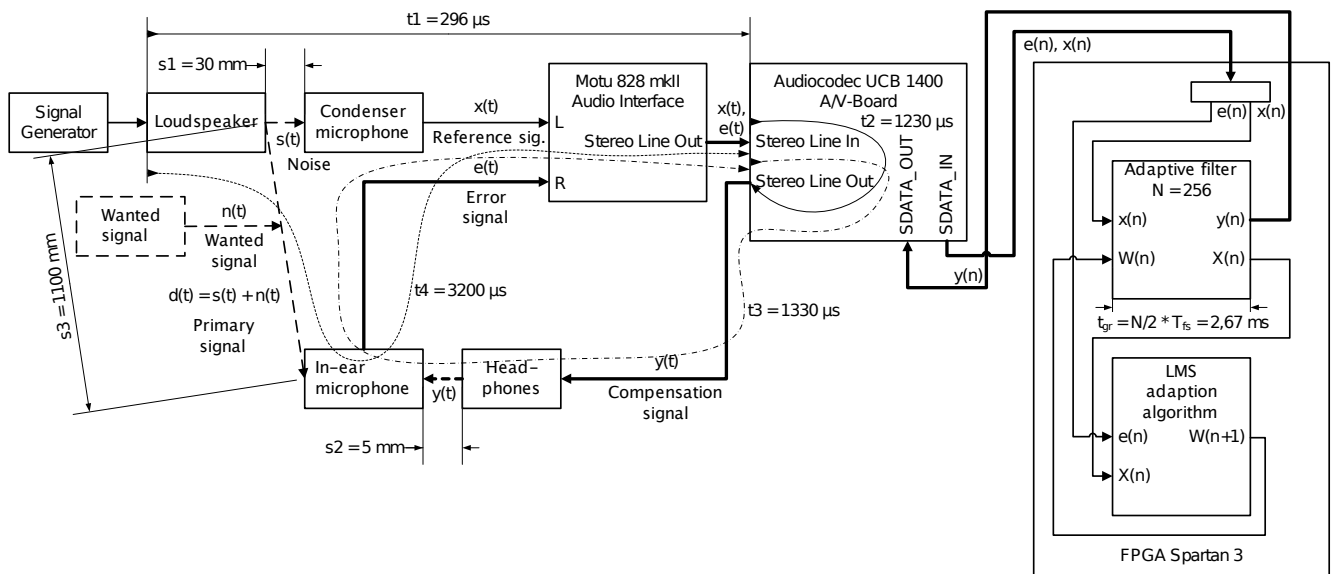


Figure 2: Block diagram of the ANC system with measured group delays in the signal path. The secondary path is marked with thick lines

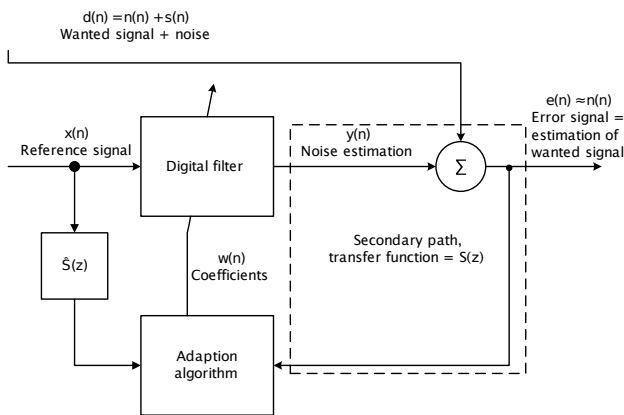


Figure 3: Adaption algorithm with secondary path compensation

At the present stage of development only the *group delays* in the signals are modeled. No attempts were made to model the frequency response of the transmission paths.

Especially important for the development and testing of the adaption algorithm is the proper modelling of the *secondary path* (see figures 2 and 3), i.e. the signal chain: FIR filter output – audio codec internal interface – audio codec output – headphones – error microphone – audio interface – audio codec input – audio codec internal interface. Measurements showed, that the amplitude and phase responses of the secondary path can be considered linear for frequencies up to approx. 2.5 kHz, so that in this frequency region the secondary path may be modeled by a simple delay line. The influence of this delay is compensated by inserting the block $\hat{S}(z)$ between reference signal input and adaption block (see figure 3). This modification of the LMS algorithm is called *filtered-x algorithm* or *FxLMS algorithm* [16, 17, 18].

2.4. Design Principles

For efficient VHDL synthesis it is important to consider the technology offered by the specific FPGA architecture and the parametrization of the synthesis tool. The VHDL design must lead to optimal usage of embedded cells like counters, MACs and FSMs. A set of rules and best practises is given in [19]. These rules for the register transfer modeling style (RT-model) define an approach of synthesis-oriented design and lead to an optimal utilisation of hardware resources and are thus yielding maximum system performance.

The VHDL model is structured as a processor element with a data path and a control path. The data path represents a processing pipeline, and the control path handles the data transformations between the single stages of the pipeline.

2.5. Implementation of the Adaptive FIR Filter

Figure 4 shows the components of the adaptive filter. The constituent parts of the data path are the two serial MAC units for the adaption algorithm (*MAC_LMS*) and for the FIR algorithm (*MAC_FIR*), the RAM unit for storage of the current filter coefficients, and the audio samples. The control path functionality is implemented in the state machine *FSM_LMS_FIR* and its connecting signals. Arithmetic is modeled with Q format number representation which provides for each pipeline stage an appropriate number of guard bits for representing the integer part and avoiding overflow effects.

The next sections describe the design of the FIR filter, the adaption process and the finite state machine of the control path.

2.5.1. The FIR filter

The FIR filter design is based on the transposed direct form in order to keep the maximum data path length short, since this is the limiting factor for the system clock frequency. Figure 5 shows, that for the direct form the maximum data path contains 1 multiplier and N adders, whereas the maximum data path for the transposed

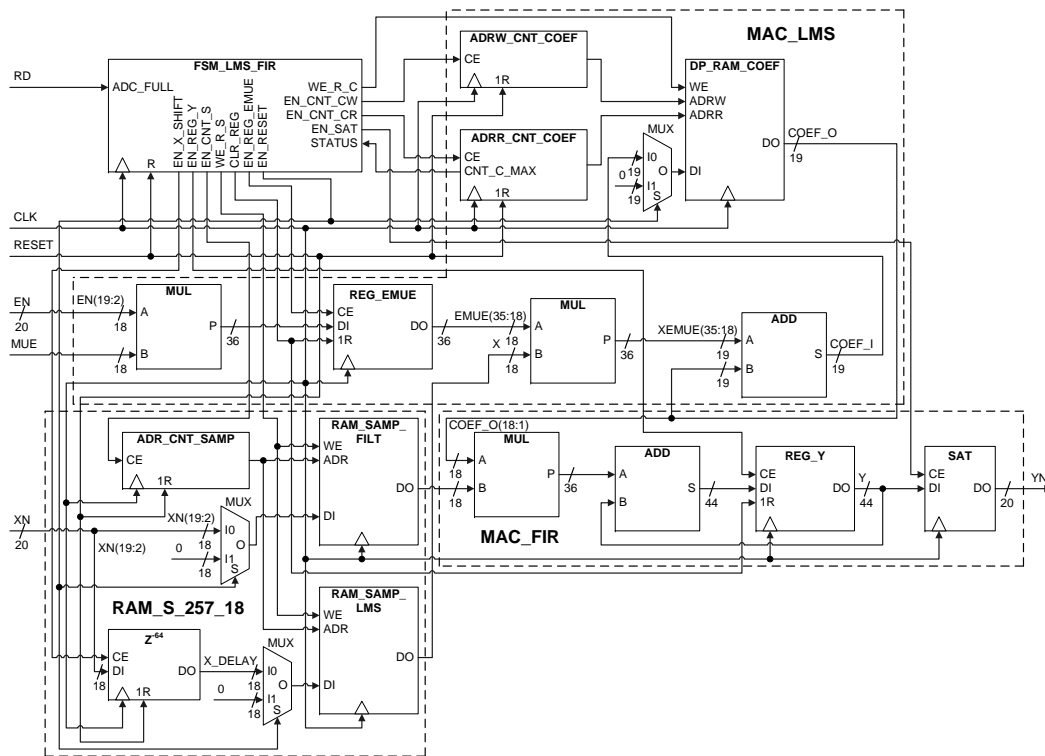


Figure 4: Functional units of the adaptive filter

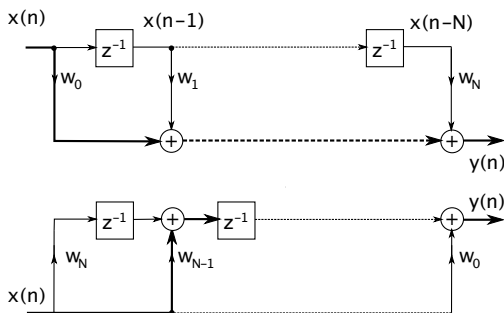


Figure 5: FIR filter structures. The maximum data path length is indicated by the thick lines. Top: direct form. Bottom: direct form transposed

direct form contains only one multiplier and one adder regardless of the filter order.

Finally the filter is implemented as a sequential MAC unit which performs $N + 1$ accumulations of products during every sample period so that a resource sharing can be utilised: since the audio sample period f_s provides a large amount of available clock cycles per audio sample, no parallel structure with $N + 1$ multipliers and N adders is necessary.

The updated input samples read from the filter RAM block (RAM_SAMP_FILT) are multiplied with their corresponding filter coefficient taken from the dual-ported RAM block DP_RAM_COEFF and stored in the accumulator. The filter output signal is fed to the

saturation block SAT , which prevents the filter output from overflow and inverts the sign of the output signal to provide the phase shift for the compensation step.

2.5.2. LMS Algorithm for FIR Coefficient Adaption

The MAC_LMS entity stores the FIR coefficients to the dual port RAM DP_RAM_COEFF . The coefficient adjustment $XEMUE$ is calculated by a product of the delayed input sample $SAMP_LMS$ and the weighted error signal $EN * MUE$. A register is inserted in this path that splits the arithmetic chain for achieving a shorter signal delay so that a clock frequency of $f_{CLK} = 50MHz$ can be met. The dual port RAM is chosen to support a parallel processing of the coefficient update and the MAC unit of the FIR filter. With two address inputs the reading address of the coefficients and the address for writing back the updated coefficients can be incremented within two interleaved clock periods.

2.5.3. The FSM of the Control Path

The finite state machine FSM_LMS_FIR in the upper left part of figure 4 controls the processing of the two parallel pipelined data paths. The state diagram of the FSM shown in figure 6 describes a sequence which is started with each new input sample pair EN (error signal) and XN (reference signal). In particular, the main controlled steps are the storage of a new sample of the reference signal XN , the calculation of the product of error signal EN and step size factor MUE , the alternating sequence of reading and writing the coefficients with parallel enabling the accumulator register REG_Y of the entity MAC_FIR . The last two states $STOP$ and

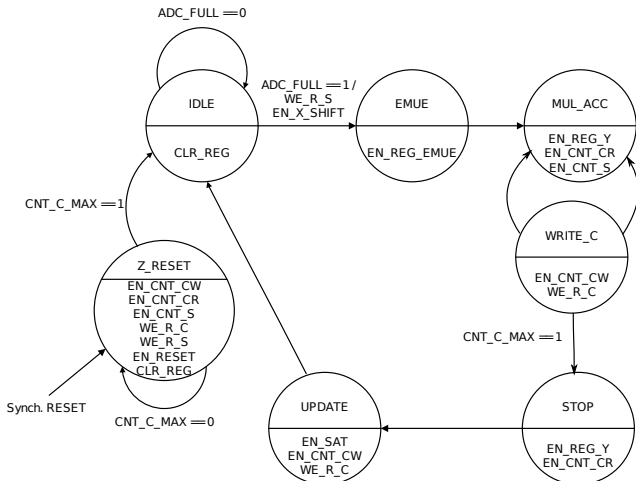


Figure 6: State diagram of the FSM of the adaptive filter

UPDATE provide the transfer of the accumulation result Y to the saturation module, which holds this value for one sample period and performs the adjustment of the RAM address counters for the next sequence.

3. RESULTS AND DISCUSSION

3.1. Hardware Resources

The system design has been transferred to VHDL code which was checked for functional and timing correctness with the ModelSim simulator.

The FPGA utilisation reported by the synthesis tool is below 25 % for all resource categories. Only 3 of the 16 available embedded multipliers were used. The longest data path in the system is given by the $MUL-ADD$ operations of the coefficient adaption and limits the clock frequency got $f_{CLK,max} \leq 80MHz$.

The maximum realisable filter order N_{max} depends on the available clock cycles N_{filt} within a sample period and the number of clock cycles M , which are necessary to process one FSM cycle. These are the timing values of the clocks. BIT_CLK is the clock signal for the audio codec which drives the serial telegram with the 20 bit data samples. To transfer one audio sample to and from the codec, 40 clock cycles are required. CLK is the oscillator clock signal for the operations of the RT model, and f_S is the sampling frequency of the audio codec.

$$\begin{aligned} f_{BIT_CLK} &= 12.288MHz \\ f_{CLK} &= 50MHz \\ f_S &= 48kHz \end{aligned}$$

Since filtering and adaption are implemented as parallel processes, the number of cycles required by the FSM is:

$$M = 2N + 5 \quad (9)$$

Therefore the maximum filter order results from

$$M = N_{Filt} = 2N_{max} + 5 \Rightarrow N_{max} = 436 \quad (10)$$

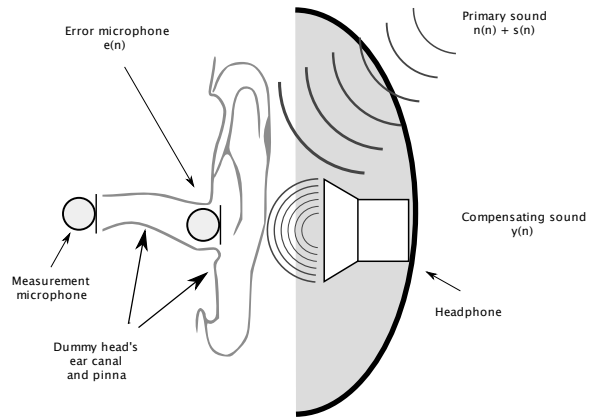


Figure 7: Measurement setup

3.2. Measurements

Figure 7 shows the measurement setup. The error signal $e(n)$ is measured with the error microphone at the entrance of the ear canal, the compensating signal $y(n)$ is fed to the headphones and mixes with the ambient primary signal. The resulting signal is recorded with the measurement microphone of the dummy head.

3.2.1. Cancellation of Sinusoidal Sounds

In the first measurements the cancellation of sinusoidal sounds was investigated. Figure 8 shows the compensation of an 1500 Hz sine signal. It can be seen from the upper part of the figure, that maximum compensation is achieved after approximately 1.5 seconds. The signal is reduced by 20 dB. The following table summarizes the measurement results for different frequencies.

Freq.	t	Attenuation	μ
200	1.3 s	21 dB	1 / 4096
400	1.0 s	23 dB	1 / 4096
1500	1.1 s	23 dB	1 / 1024
4000	0.7 s	11 dB	1 / 1024
5000	0.7 s	0 dB	1 / 1024

The compensation works properly in a frequency range between 200 Hz and 4000 Hz. The theoretical lower limit of the compensation frequency range is

$$f_{min} = \frac{f_S}{N} = \frac{48kHz}{256} \approx 188Hz \quad (11)$$

where f_S is the audio sampling frequency and N is the filter order. At frequencies above 4000 Hz the damping of the secondary path (see figures 2 and 3) results in a too small error signal e and thus a too low amplitude of the compensating signal y . Above 5 kHz there is no more compensation at all.

3.2.2. Cancellation of Fan Sounds

In the next experiment the compensation of broadband audio signal (the fan noise of an electronic device) was measured. The spectrum in figure 9 shows major contributions in the range from 200 Hz to 300 Hz. In this range a signal attenuation of approximately 10 dB is achieved. The adaption step size factor μ had a

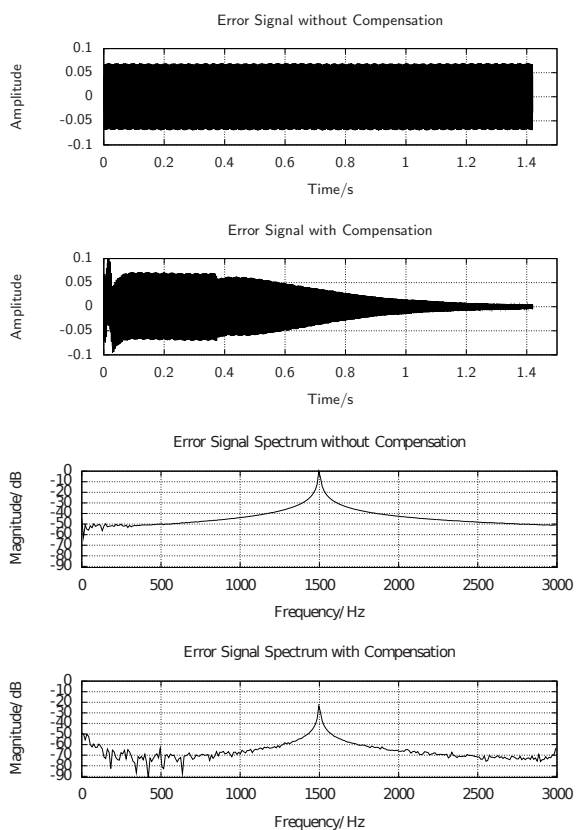


Figure 8: Compensation of a 1500 Hz sine signal. $\mu = 1/1024$

value of $1/128$. It can be seen, that the frequency components below 190 Hz are not attenuated at all, as was to be expected from the filter order of 256. It has to be noted, that the large LMS step size factor μ leads to an overshoot of the error signal for the first 5 seconds after activating the compensation.

3.2.3. Cancellation of Car Motor Sounds

This measurement was carried out with the recorded sound of a car. Again a step size factor μ of $1/128$ was chosen, which again lead to a significant overshoot for the first 1.5 seconds. According to equation (8), the overshoot time is shorter in this measurement, because the main spectral contributions to the signal are at higher frequencies than in the fan sound, resulting in a higher mean power of the signal, as can be seen from figure 10. The regions of major spectral power are located at the frequencies 210 Hz, 440 Hz and 475 Hz. In this spectral range an overall attenuation of 8 dB is measured.

3.2.4. Sounds with time dependent statistical properties

In a final measurement the system was tested with the sound of an accelerating car. Here the results were quite poor, the overall attenuation was below 3 dB. This poor performance arises from the fact, that the adaption step size was constant over the measurement time. The variations in level and frequency of the reference sound implied time-varying statistical properties of the sig-

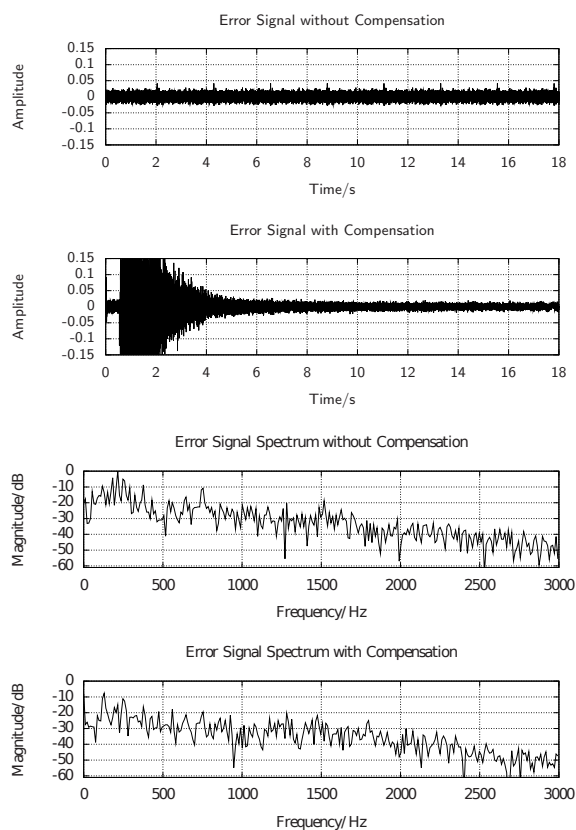


Figure 9: Compensation of a fan sound signal. $\mu = 1/128$

nal. From equation (8) it can be seen, that this leads to either poor convergence of the algorithm or to overshooting of the adaption algorithm, i.e. an *amplification* of the noise signal instead of an attenuation.

3.3. Discussion

The noise reduction works sufficiently for stationary noise in a frequency range between 188 Hz and 4000 Hz. As previously stated, the lower limit f_{min} is given by $\frac{f_s}{N}$, whereas the upper limit of the working range is given by the fact, that above 4000 Hz the amplitude and phase responses of the secondary path (see figure 3) can no longer be considered linear.

To overcome these limitations, three improvements will have to be made. First the step width factor μ will have to be continuously computed according to a running estimation of the mean power of the reference signal $x(n)$. The next improvement will be the identification of the transfer function $S(z)$ of the secondary path and design a compensating filter. Last the filter order N must be increased to extend the working range to lower frequencies. This can be achieved by instantiation of larger block RAMs for the coefficients and the input samples. Additionally the clock frequency f_{CLK} can be increased to $80MHz$ which provides more available clock cycles N_{filt} according to equation (9). This is possible without changing the general design, because there are enough unused hardware resources on the board.

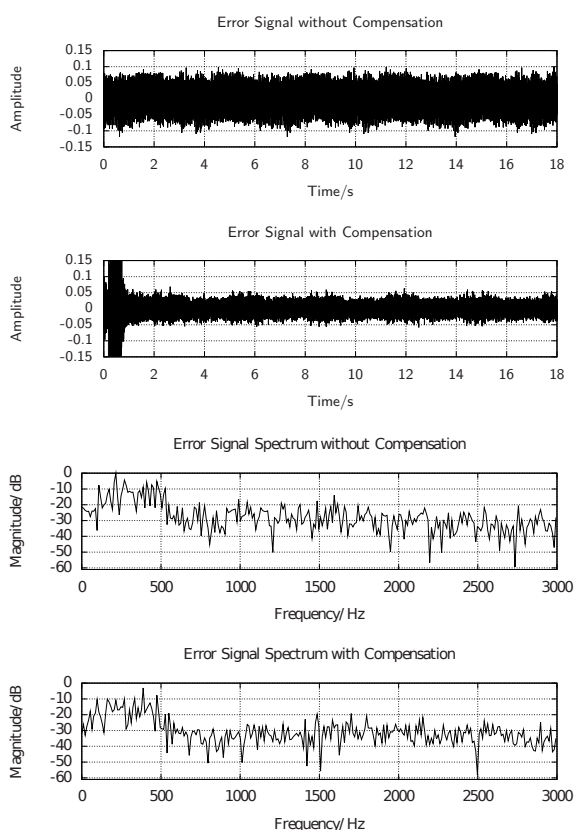


Figure 10: Compensation of a car motor sound signal. $\mu = 1/128$

4. CONCLUSION

The measurements of the previous section proved, that the FPGA platform is well suited for the complex real time audio processing tasks in augmented reality audio systems. An adaptive noise cancellation process has successfully been implemented.

Filter orders of 256 and above can be realized with the Spartan-3 FPGA XC3S400 board.

Measurements with real-life audio signals have been carried out to investigate the performance of the system. The limiting factors have been identified and will be overcome in the further development.

5. REFERENCES

- [1] Uwe Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Arrays*, Springer, 2004.
- [2] S. Benford, C. Magerkurth, and P. Ljungstrand, "Bridging the Physical and Digital in Pervasive Gaming," *Communications of the ACM*, 2005.
- [3] C. Magerkurth, A.D. Cheok, R.L. Mandryk, and T. Nilsen, "Pervasive Games: Bringing Computer Entertainment Back to the Real World," *Computers in Entertainment (CIE)*, vol. 3, no. 3, pp. 4-4, 2005.
- [4] A. Härmä, J. Jakka, M. Tikander, M. Karjalainen, T. Lokki, J. Hiipakka, and G. Lorho, "Augmented Reality Audio for

- Mobile and Wearable Appliances," *J. Audio Eng. Soc.*, vol. 52, no. 6, pp. 618-639, 2004.
- [5] M. Karjalainen, T. Lokki, H. Nironen, A. Harma, L. Savioja, and S. Vesa, "Application Scenarios of Wearable and Mobile Augmented Reality Audio," in *116th AES Convention, Berlin, Germany*, May 8-11 2004.
- [6] Miikka Tikander, Matti Karjalainen, and Ville Riikonen, "An Augmented Reality Audio Headset," in *Proc. of the 11th Int. Conference on Digital Audio Effects (DAFx08), Espoo, Finland*, 2008, pp. 181-184.
- [7] C.M. Kim, H.M. Park, T. Kim, Y.K. Choi, and S.Y. Lee, "FPGA Implementation of ICA Algorithm for Blind Signal Separation and Adaptive Noise Canceling," *IEEE Transactions on Neural Networks*, vol. 14, no. 5, pp. 1038-1046, 2003.
- [8] A. Di Stefano, A. Scaglione, and C. Giaconia, "Efficient FPGA Implementation of an Adaptive Noise Canceller," in *Computer Architecture for Machine Perception, 2005. CAMP 2005. Proceedings. Seventh International Workshop on*, July 2005, pp. 87-89.
- [9] A. Elhossini, S. Areibi, and R. Dony, "An FPGA Implementation of the LMS Adaptive Filter for Audio Processing," in *IEEE International Conference on Reconfigurable Computing and FPGA's, 2006. ReConFig 2006*, 2006, pp. 1-8.
- [10] Tian Lan and Jinlin Zhang, "FPGA Implementation of an Adaptive Noise Canceller," in *Information Processing (ISIP), 2008 International Symposium on*, May 2008, pp. 553-558.
- [11] R. Trausmuth and M. Kollegger, "ADAM-A 64 Channel General Purpose Realtime Audio Signal Processor," in *Proc. Int. Conf. on Digital Audio Effects (DAFx-04), Naples, Italy*, 2004.
- [12] R. Trausmuth and A. Huovilainen, "POWERWAVE-A High Performance Single Chip Interpolating Wavetable Synthesizer," in *Proc. Int. Conf. on Digital Audio Effects (DAFx-05), Madrid, Spain*, 2005.
- [13] Dimitris G. Manolakis, Vinay K. Ingle, and Stephen M. Kogon, *Statistical and Adaptive Signal Processing*, McGraw-Hill, 2000.
- [14] George S. Moschytz and Markus Hofbauer, *Adaptive Filter*, Springer, 2000.
- [15] B. Widrow and M.E. Hoff, *Adaptive Switching Circuits*, MIT Press Cambridge, MA, USA, 1988.
- [16] Y. Gong, Y. Song, and S. Liu, "Performance Analysis of the Unconstrained FxLMS Algorithm for Active Noise Control," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03)*, 2003, vol. 5.
- [17] X. Kong and SM Kuo, "Study of Causality Constraint on Feedforward Active Noise Controlsystems," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, no. 2, pp. 183-186, 1999.
- [18] Bernard Widrow and Samuel D. Stearns, *Adaptive Signal Processing*, Prentice-Hall, Inc., 1985.
- [19] Jürgen Reichardt and Bernd Schwarz, *VHDL-Synthese: Entwurf digitaler Schaltungen und Systeme*, Oldenbourg, 2007.