Active noise control in the concept of IoT

Róbert Galambos
Department of Measurement and Information Systems
Budapest University of Technology and Economics
Budapest, Hungary
Email: galambos@mit.bme.hu

László Sujbert
Department of Measurement and Information Systems
Budapest University of Technology and Economics
Budapest, Hungary
Email: sujbert@mit.bme.hu

***Abstract*—This paper introduces a ”proof of concept” system for Active Noise Control (ANC) where the audio data from the reference microphones are travelling through the already existing Ethernet network. The system realizes the active noise control with the help of a multi channel DSP card. The noise control algorithm is the Filtered Error Least Mean Square (ELMS) – an LMS variant –, which has already proven its efficiency previously in different environments. The audio data from the reference microphone to the DSP card are digitalised with the help of an ADSP-BF537 Ez-Lite Kit, and transmitted over Ethernet to a counterpart ADSP-BF537 Ez-Lite Kit card, which converts it back to an analogue signal to be able to be fed into the multi channel DSP card. The article highlights the usage of Ethernet network for audio transmission, and shows the problems that are caused by the properties of such medium, as well as the potential that lies in it. The working of the system and its efficiency is illustrated by measurement results.**

I. Introduction

Active noise control (ANC) is a well known technology in the engineering, and is used to replace passive isolation where the usage of that is not sufficient, not possible, or not worth. ANC is used for low-frequency acoustic noise suppression, and it is based on the phenomenon of destructive interference [1]. The interference is created by a controlled noise source that suppresses – cancels out – the primary (e.g.: original) noise source.

Although the basic concept of ANC is known since almost a century, it became only in the last few decades more and more the center of interest. This was due to the increase of computation power, and the wide spreading of digital signal processors (DSPs), and so the new possibilities that were available to use. The experience acquired in these decades clarified the possible application areas of the principle [1] [2] [3].

The ANC as problem can be seen and separated into two branches: acoustics and control. The acoustic part of the problem is placing microphones and loudspeakers in an arrangement best suited for the given situation. The control part on the other hand is how the loudspeakers have to be controlled by taking into account the information from the microphones. In the last years many algorithms have been developed and implemented. These algorithms can also be grouped into two main categories: algorithms for stochastic or periodic noise cancellation. For periodic noise cancellation a feedback controller is sufficient, but for broadband stochastic noise cancellation feed-forward control is needed [1] [2]. This feed-forward control needs reference microphones and is more efficient when these microphones are located near to the noise source. Our ”proof of concept” system targets stochastic broadband noise cancellation, because it is the most sensitive for audio signal latency.

Even when the microphone arrangement can be determined, the realisation can be a problem. Noise sources can be far away from the cancellation area, and it takes a lot of effort to create a proper cable network suitable for audio transmission for the reference microphones to get information from near the noise sources, and so get better results. However in the information age, almost every building is already equipped with a sufficient network for local communication and for the internet. These networks have a bandwidth ranging from 100 Mbit/s to 1 Gbit/s, and even the internet connections are in the 10 Mbit/s range. Although the topology of these networks is mostly complex and hierarchical the response time is within the millisecond range. These two properties could make the Ethernet networks suitable for low-latency audio transmission.

The goal of our ”proof of concept” system is to show that there is a certain potential in using these networks. By replacing the analogue cables with the digital Ethernet transmission, we achieve a more flexible system that can be rearranged and modified much easier [2]. Even further this system allows to apply the concept of Internet of Things (IoT) in the field of ANC.

The structure of this paper is as follows. Section II gives a high level overview of the system, by defining each major component, and the arrangement of the microphones and loudspeakers. Section III introduces the algorithm used for the ANC, and gives a more detailed description about its properties. Section IV shows how the audio was transmitted over Ethernet, including the software architecture, the buffering scheme, and their properties. Section V contains the measurements results for the ”proof of concept” system, showing the efficiency of the ANC with the Audio over Ethernet and without.

II. System description

Fig. 1 shows the arrangement of the ”proof of concept” system. As seen the least complex arrangement was used for the system, with one reference microphone, one error microphone, and one noise source, but this noise source that needs to be cancelled radiates stochastic broadband noise, to be able to detect problems caused by the audio latency of the reference signal path. Besides that, two different kinds of DSP cards were also used. A multichannel DSP card, and two card for audio data transformation.

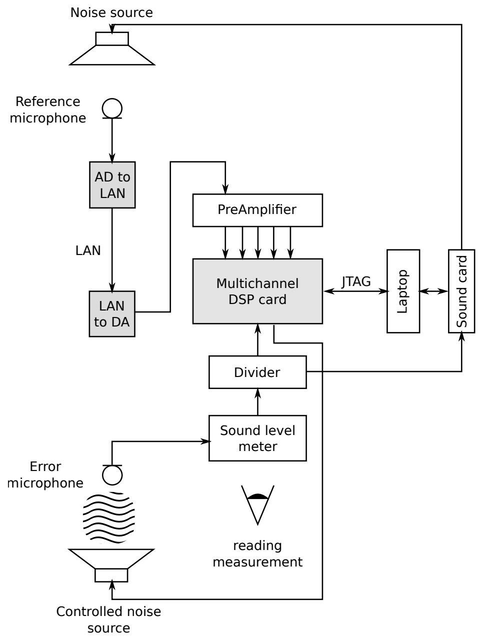

Fig. 1. The arrangement

The multichannel DSP card is a special DSP card consisting of a floating point ADSP-21262 Shark DSP processor [4] and is capable of sampling simultaneously several channels [5]. The noise control algorithm is running on this board, using the input channels to produce the cancellation noise on the output.

Because the multi-channel DSP card does not have any Ethernet interface there was a need for a solution to transmit audio over local area network (LAN). For this transmission two ADSP-BF537 Ez-Lite Kit boards were used [6]. These boards consist of an ADSP-BF537 (Blackfin) 16-bit fixed point DSP, and are equipped with 48 kHz/24 bit capable analogue to digital (ADC) and digital to analogue converters (DAC), and a 10/100 Mbit/s LAN interface.

III. Noise control

A. Noise control problem

The basic problem of the ANC can be summarized and shown in a very compact way, as seen in Fig. 2. In the generalised form of ANC all the signals are vectors, since usually the control system consists of several microphones and loudspeakers. The **x** input vector is the vector containing the samples of the reference inputs of the system, and the **e** vector is the error vector. These two vectors are fed into the **F** adaptive system, which is mostly some kind of an adaptive filter. The anti-noises generated by the system (**y**) are than radiated from the loudspeakers [7]. An important feature of the problem – from theoretical perspective – is that the generated anti-noise is not subtracted directly from the disturbance (**d**) signal, rather than filtered by the system **A**. This is the transfer function from the **F** adaptive system to the cancellation area. It includes the transfer functions of all analogue and digital parts and the imperfection of the sound radiation. Usually **A** is called the secondary path transfer function, since the acoustic path appeared only because of the ANC system, while the primary path is in-between the noise source and the **d** disturbance signal. The adaptive system **F** can utilize the reference vector **x** if it is a feed-forward controller, or do not use any reference input if it is a feedback-controller. A more detailed analysis of the problem can be found in [1] [8].

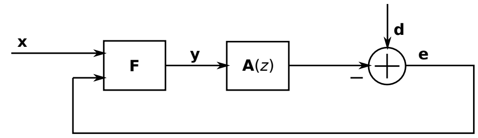

Fig. 2. The noise control problem

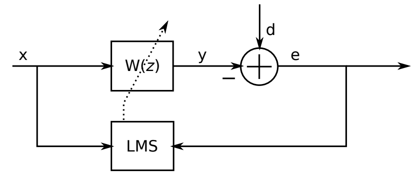

Fig. 3. The LMS algorithm

There are different approaches for ANC design, but all of them needs to take into account the identification of the secondary path **A**. An insufficient or inaccurate model of the secondary path **A** destabilises the system [3]. The system is stable if and only if the phase error of the model does not exceeds $\pi/2$. Considering that above the quarter of the sampling frequency a simple unit delay identification error can destabilize the system, this is a hard condition.

B. Noise controller

The mostly used adaptive algorithm for ANC is the least mean square (LMS) algorithm. The LMS adaptive filter can be seen in Fig. 3, where $W(z)$ denotes the adaptive transverse filter, x is the reference signal, y is the output of the filter, and e is the error signal of the system [9]. The signals are already one dimensional ones, as the ”proof of concept” system is a one-channel system.

However this fairly idealistic model does not take in account the disturbance caused by the secondary path (**A**), that can destabilize the controller. So an improved algorithm is needed. An advanced version of the LMS algorithm is the filtered-reference LMS (XLMS) algorithm [9], where the secondary path (**A**) is also considered in the form of filtering the reference signal accordingly. In order to increase the convergence rate, our ”proof of concept” system uses the filtered-error LMS (ELMS) algorithm [9]. The arrangement can be seen in Fig. 4, where $A(z)$ is the secondary path of the system, $A_D^{-1}(z)$ is the so called delayed inverse of the secondary path, and z^{-D} is the same delay that is used in the delayed inverse. The system can be described with the following equations:

$$y_n = \mathbf{w}_n^T \mathbf{x}_n \tag{1}$$
$$e_n = d_n - A(z) y_n \tag{2}$$
$$w_{n+1} = w_n + \mu A_D^{-1}(z) e_n \bar{\mathbf{r}}_n \tag{3}$$

where

$$r_n = z^{-D} x_n \tag{4}$$

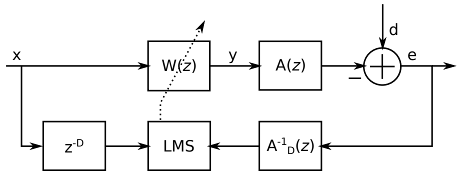

Fig. 4. The ELMS algorithm

The delayed inverse $A_D^{-1}(z)$ can be defined as follows:

$$A(z)A_D^{-1}(z) \approx z^{-D} \quad (5)$$

where $A_D^{-1}(z)$ is a finite impulse response (FIR) filter, and the overall transfer function of the secondary path and the error signal path approximates a simple delay.

Because the secondary path is unknown, the structure requires to identify $A(z)$. This can be done with a simple LMS algorithm in case of XLMS or ELMS algorithms, only the order of the approximated filters are different from the order of $W(z)$. For both algorithms the identification requires a white noise signal, and in the case of ELMS the z^{-D} delay has to be set experimentally.

Another problem of the structure is the required number of coefficients to realize the $A_D^{-1}(z)$ inverse filter. Taking the fair assumption that $A(z)$ is rational, the ideal inverse can easily be constructed by exchanging the poles and zeros of the $A(z)$. If $A(z)$ is a minimum phase filter, the exact inverse can be calculated without the need of any extra delay (z^{-D}), however if $A(z)$ is not minimum phase only the delayed inverse exist ($D \neq 0$). This is because if $A(z)$ is not minimum phase, the inverse would require a non-casual inverse filter. A workaround of the problem is to delay the inverse filter, and make it quasi causal. By constructing the delayed inverse the impulse response of $A^{-1}(z)$ has to be truncated at step $n = -D$, and it has to be shifted by D.

IV. Signal path

As mentioned earlier our "proof of concept" system uses Ethernet communication instead of analogue cable to transfer audio data from the reference microphone. The audio to LAN and the LAN to audio adapter is an ADSP-BF537 Ez-Lite Kit board, that uses the Visual DSP++ Kernel (VDK) as real-time environment (RTE). The board is equipped with LAN connectivity and the Visual DSP++ Framework is capable to generate a functional TCP/IP stack, and uses therefore the open-source lwIP implementation.

Unfortunately the VDK Framework provided possibilities for rapid prototyping (e.g.: generating TCP/IP stack) are sometimes drawbacks when fine tuning of the system is needed (e.g.: modification of the stack is a huge effort). It means that currently the lwIP stack is only capable of using Dynamic Host Configuration Protocol (DHCP) to obtain Internet Protocol address (IP). This makes it impossible to connect two cards directly via a simple Ethernet cable. Fortunately the realistic use case always requires a present and working network, and in such an environment it is safe to assume that DHCP can be used.

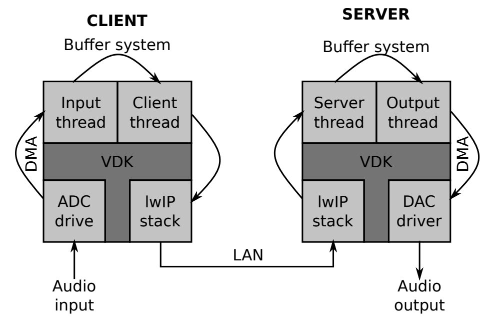

Fig. 5. The Software architecture

A. Software architecture

As seen in Fig. 5 the software consist of a client and a server. They are built from the same source, and only some configuration parameters need to be changed to compile for client or for server. The parameters that can be configured to fine tune the system are, the server IP address where the client tries to connect, the port on which the server accepts the clients' communication, the size of the buffers used for audio transmission, and the number of buffer for the buffering scheme.

On client side the audio data returned from the ADC are handled by the ADC driver which with the help of the DMA copies the data – without blocking the CPU – into the application buffers. The input thread waits in a blocking statement till the DMA finishes a block and signals through the VDK that it is ready. After this the input thread passes the buffer to the client thread's buffer, where the package is constructed. The client thread upon receiving the data from the input thread, starts a transmission over LAN. The packages go through the lwIP stack which attaches the required frame headers for the given transmission protocol. The final package, after processed by the lwIP stack, is put by the boards Ethernet peripheral onto the LAN as physical signals.

On server side, after the package has arrived into the boards LAN peripheral the lwIP stack removes the protocol headers from the frame, and passes the stripped buffer to the server thread. This thread extracts the audio information from the received buffer, and passes the buffer to the output thread. The output thread is in a blocking state till it gets a buffer from the server thread, and upon awakening it passes the audio data from the buffer received to the DMA. The DMA will put the data to the audio peripheral of the DSP and through the peripheral the DAC will produce the analogue sound.

B. Buffering scheme

On the client and on the server side, the buffering of data between the input thread and client thread, as well as the server thread and the output thread is done in a circular manner.

As seen in Fig. 6 there are several small buffers between the data source and the data destination, and if a data package is received by the data source from the input, the data will be extracted and put into one of the circular buffers, and then sent to the data destination. The data destination then sends the content of the buffer to the output.

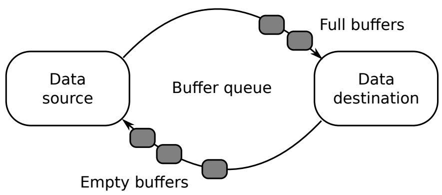

Fig. 6. The buffering queue

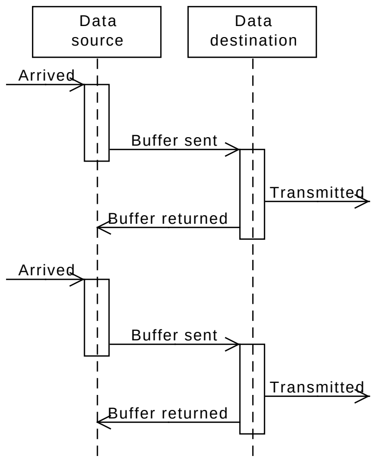

Fig. 7. The threads queue communication

Fig. 7 shows that the data source thread is blocked till new data arrived from the input, and after sending it to the data destination it will became blocked again. The data destination behaves similarly. It is in blocked state waiting for the buffer coming from the data source, and upon arrival, it processes the data and transmits it to the output. The already used buffer will be returned to the data source, so it will be able to use it again.

The implementation of the buffering is based on this scheme in both the client and the server side code. On client side the arrived data from the "input" are coming from the ADC and the DMA will signal if the data reception for a buffer is read. The "data source" is the input thread, and the "data destination" is the client thread, and therefore when the data are sent to the "output", it means it is sent through the lwIP TCP/IP stack and put on the Ethernet network. On server side the arrived data from the "input" are actually coming from the Ethernet network and the lwIP will signal if there are data which can be processed. The "data source" is the server thread, and the "data destination" is the output thread, that will send data to the "output" and therefore through the DMA out of the DAC.

C. Protocol consideration

As for communication over the Ethernet network first TCP/IP was considered because of it provides reliable, ordered and of course error-checked delivery of data between client and server [10]. Unfortunately even on a LAN the overhead of such a complex protocol is too high. In our tests we saw that even with the minimum number of circular buffers, and with the smallest circular buffer size, it was only possible to reduce the latency to approximately 18.5 ms. By taking into account that the speed of sound is approximately 334 m/s, that means that the noise source must be $18.5 \cdot 10^{-3} \cdot 334$ approximately 6.2 m away to reach the noise cancellation area at the same moment as the TCP/IP packages arrived on the LAN. Therefore we switched to User Datagram Protocol (UDP), as it is much simpler, connectionless transmission model with the minimum amount of protocol mechanism (e.g.: no handshaking) [11]. Of course by using such a protocol there is a risk that we loose packages, but considering that in a real-time application there is not much time to handle lost or damaged packages it is not a big restriction. By using UDP packages the latency of signal path was reduced to approximately 3 ms, which means that the noise source must only be $3 \cdot 10^{-3} \cdot 334$ approximately 1 m away. This is an acceptable compromise, since for a noise source in range of 1-2 m of the cancellation area, it is anyway easier to use analogue cables, than Ethernet network connection.

V. Results

The setup for the measurement was based on the arrangement in Fig. 1, the reference microphone was connected to the ADSP-BF537 Ez-Lite Kit board via analogue cable. The data were digitalized by the board and sent over Ethernet to the counterpart board, that did the digital to analogue conversion and fed the signal to the multichannel DSP card's reference input through a pre-amplifier. The multichannel DSP card was also connected to an error microphone, and a loudspeaker. These connections were analogue because the multichannel DSP card was located near to the noise cancellation area. The signal of the error microphone was measured with a sound level meter, and divided in two, so making it possible to record the error signal with the laptop. The source of the noise was another loudspeaker, getting its signal from the laptop through an external sound card.

The microphones used for the measurement are ECM8000 Behringer measurement microphones. The loudspeakers are active monitor loudspeakers. The primary noise was a band limited noise with a range from 100 Hz to 1000 Hz. The resolution of the laptop's external sound card was 44100 Hz at 24 bit.

The experiments were carried out in a laboratory, which is a 7.5 m $\times$ 5 m $\times$ 3 m room with moderate reflexions. The primary noise source was located approximately 4 m far from the cancellation area, while the error microphone was set directly to the secondary loudspeaker. The reference microphone was 1 m apart from the primary source.

The first measurement was the "reference measurement" and therefore the reference microphone was directly connected with an analogue cable to the multichannel DSP card, replacing so the Ethernet link for audio transmission.

Fig. 8 shows the result of the reference measurement. The figure depicts the stochastic noise Sound Pressure Level (SPL) change during the adaptation of the filter in decibel scale. The raw SPL level was smoothed and every 100 samples of SPL was replaced with the corresponding average SPL level, therefore the unit of the x axis is $\mathrm{sample}/100$. As it can be

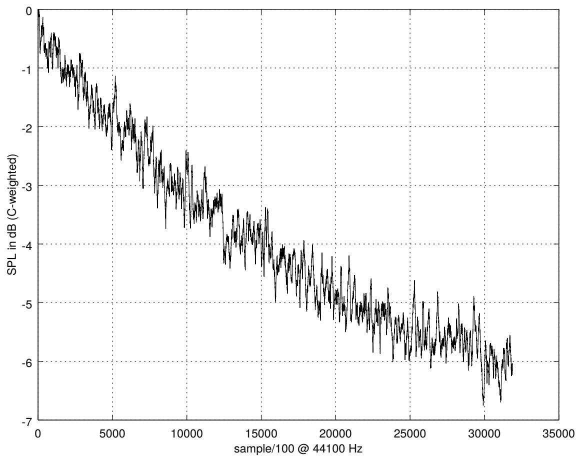

Fig. 8. The ANC using microphone cables

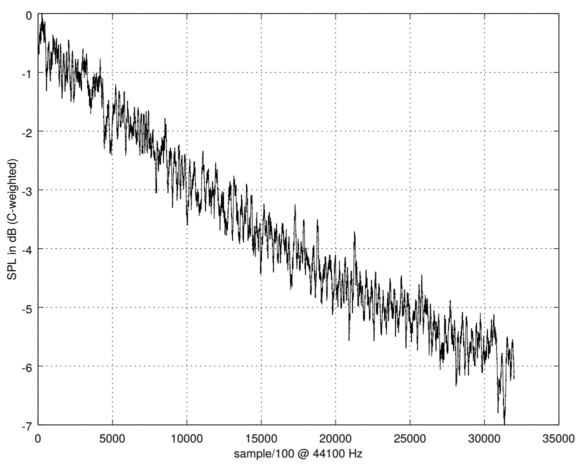

Fig. 9. The ANC using audio over Ethernet

seen, the attenuation of the stochastic noise was approximately 6 dB, and it took the system more than 70 sec to settle.

The second measurement was the ”audio over LAN measurement” and therefore the reference microphone was connected via Ethernet link to the multichannel DSP card, instead of an analogue cable.

Fig. 9 shows the stochastic noise Sound Pressure Level (SPL) change in decibel scale during the adaptation of the filter. This signal was also smoothed, and every 100 samples of SPL was replaced with the corresponding average SPL level, therefore the unit of the x axis is the same as before: $\mathrm{sample}/100$. As it can be seen, the attenuation of the stochastic noise was almost the same: approximately 6 dB, and it took the system more than 70 sec to settle. This means that the two responses of the system are not significantly different, and in the current setup the ELMS algorithm tolerates the network latency. Note that the physical arrangement as well as the algorithm has not been optimised for control purposes in this acoustic environment. However the results are characteristic for the development of a LAN-based ANC system.

VI. Conclusion

In this paper we have introduced a ”proof of concept” system to show that it is feasible to use Ethernet communication to transfer real-time audio data over the LAN in an ANC system, making it possible to transfer the reference microphones’ signal from a distant noise source without the need of expensive analogue wiring, by using the already present Ethernet network. Our system as shown in the measurement results did not significantly deviated from the analogue wired system, showing that if the sound propagation time from the noise source to the cancellation area is more than the latency on the network, then the ELMS algorithm is robust enough and the result will be almost the same as in the analogue wired case. The main goal of future work is to analyse the system’s behavior with realistic network load, improve the network package handling, and make the system more robust against network load related problems (e.g. package lost).

Acknowledgment

This work was partially supported by the ARTEMIS JU and the Hungarian Ministry of National Development (NFM) in frame of the R5-COP (Reconfigurable ROS-based Resilient Reasoning Robotic Cooperating Systems) project.

References

[1] S. M. Kuo and D. R. Morgan, “Active noise control: A tutorial review,” in *Proceedings of the IEEE*, vol. 87, no. 6, June 1999, pp. 943–973.

[2] K. Czyz and J. Figwer, “Suboptimal multichannel adaptive anc system.” in *IEEE Intelligent Systems and Computer Vision*, vol. 14, July. 2007.

[3] S. Elliott and P. Nelson, “Active noise control,” in *IEEE Signal Processing Magazine*, Oct. 1993, pp. 12–35.

[4] Analog Devices. (2015) Datasheets for the adsp-21262 sharc processor. [Online]. Available: http://www.analog.com/en/processors-dsp/sharc/adsp-21262/products/product.html

[5] Analog Devices. (2012) Datasheet for the ad773322 codec. [Online]. Available: http://www.analog.com/en/audiovideo-products/audio-codecs

[6] Analog Devices. (2015) Datasheets for the adsp-bf537 ez-lite kit. [Online]. Available: http://www.analog.com/en/evaluation/bf537-ezlite/eb.html

[7] S. J. Elliott, I. M. Stothers, and P. A. Nelson, “A multiple error lms algorithm and its application to the active control of sound and vibration.” in *IEEE Transactions on Acoustics, Speech, and Signal Processing*, no. 10, Oct. 1987, pp. 1423–1434.

[8] Y. Kajikawa, W.-S. Gan, and S. M. Kuo, “Recent advances on active noise control: open issues and innovative applications,” in *APSIPA Transactions on Signal and Information Processing*, vol. 1, Aug. 2012.

[9] E. Widrow, B. Walach, *Adaptive Inverse Control*. Prentice Hall, Inc, 1996.

[10] *RFC-793: Transmission Control Protocol*, DARPA Std., 1981.

[11] *RFC-7688: User Datagram Protocol*, Std., 1980.