Abstract

Conventional (passive) headsets used in propeller aircrafts are reasonably good at attenuating mid to high frequency noise, but fail to achieve good attenuation in the low frequency region (below approximately 300 Hz). Active Noise Reduction (ANR) improves the low frequency attenuation by introducing an anti-noise signal creating destructive interference thus decreasing the residual noise level. The aim of this thesis is to develop and implement a digital narrowband active noise reduction headset that works properly in aircrafts and not only in a laboratory environment. The implementation is based on a narrowband filtered-X least mean squares (FXLMS) algorithm where the tonal components in the noise spectrum are synthesized for use as reference signals to the algorithm. The controller is implemented in a parallel fashion where each tonal component is handled separately. The system is built into a headset and laboratory tests show that the algorithm can handle five simultaneous tonal components with an adaptation time of less than one second. Aircraft tests show peak attenuation of 17 dB in both single- and twin-engine aircrafts thus fulfilling the requirements. Simulations and true performance show some minor discrepancies which are explained and discussed.
# Contents

1 **Introduction** 4  
1.1 Aircraft noise 4  
1.2 Advantages with active noise reduction 5  
1.3 How ANR works 5  
1.4 Feedforward vs feedback 6  
1.5 Broadband vs narrowband 7  
1.6 Previous work 8  
1.7 Novelty 9  

2 **Problem definition** 10  
2.1 Problem definition 10  
2.2 Non-disclosure of proprietary information 10  

3 **Theory** 11  
3.1 Adaptive filtering 11  
3.1.1 Mean square error 12  
3.1.2 Steepest descent 12  
3.2 Least mean squares 13  
3.2.1 LMS stability and convergence rate 13  
3.2.2 Leaky LMS 13  
3.3 Filtered-X least mean squares 14  
3.3.1 Narrowband leaky FXLMS 15  
3.3.2 Step size constraints with secondary path 15  
3.3.3 Secondary path estimation precision 16  
3.3.4 FXLMS controller seen as notch filter 16  
3.4 Identifying secondary path 17  

4 **Measuring performance** 19  
4.1 Lab 19  
4.2 Aircraft 20  

5 **Simulation** 22  
5.1 Matlab 22
6 Hardware and software implementation
   6.1 Hardware .................................................. 25
   6.2 DSP software ........................................... 25
       6.2.1 Secondary path estimation ......................... 27
       6.2.2 Step size and leak factor .......................... 27
   6.3 Windows software ...................................... 28

7 Results
   7.1 Lab tests ................................................ 30
   7.2 Aircraft tests ......................................... 30
       7.2.1 Single engine aircraft ............................ 30
       7.2.2 Twin engine aircraft ............................... 33

8 Conclusions ................................................. 35

9 Discussion .................................................. 36
<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-weighting</td>
<td>Filter that approximates the human hearing frequency response [1]</td>
</tr>
<tr>
<td>ADC</td>
<td>Analog to Digital Converter</td>
</tr>
<tr>
<td>ANR</td>
<td>Active Noise Reduction</td>
</tr>
<tr>
<td>CODEC</td>
<td>COder/DECoder, a combination of an audio ADC and audio DAC into a single chip.</td>
</tr>
<tr>
<td>DAC</td>
<td>Digital to Analog Converter</td>
</tr>
<tr>
<td>DSP</td>
<td>Digital Signal Processor</td>
</tr>
<tr>
<td>FXLMS</td>
<td>Filtered X Least Mean Squares algorithm [2]</td>
</tr>
<tr>
<td>LMS</td>
<td>Least Mean Squares algorithm [2]</td>
</tr>
<tr>
<td>MEMS</td>
<td>Microelectromechanical systems</td>
</tr>
<tr>
<td>NRR</td>
<td>Noise Reduction Rating [3]</td>
</tr>
<tr>
<td>PCB</td>
<td>Printed Circuit Board</td>
</tr>
<tr>
<td>Primary source</td>
<td>The unwanted noise source</td>
</tr>
<tr>
<td>PSD</td>
<td>Power Spectral Density</td>
</tr>
<tr>
<td>Secondary source</td>
<td>The anti-noise source</td>
</tr>
<tr>
<td>SNR</td>
<td>Single Number Rating [4]</td>
</tr>
<tr>
<td>SPL</td>
<td>Sound Pressure Level, referred to $20\mu Pa$ [5]</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Conventional (passive) headsets used in propeller aircrafts are reasonably good at attenuating mid to high frequency noise, but fail to achieve good attenuation in the low frequency region (below approximately 300 Hz). To passively improve the low frequency attenuation, the inner volume of the headset cups and/or the weight of the cups must be increased. That would lead to obvious practical problems and uncomfortable fit. A better approach is to use Active Noise Reduction (ANR), introducing an "anti-noise" signal leading to destructive interference and thus lower noise level.

1.1 Aircraft noise

Propeller driven aircrafts have quite characteristic noise spectrums, showing similarities between different aircrafts. The main parts are broadband random noise (e.g. wind noise) and narrowband tonal noise (e.g. rotating parts as the engine, propeller, gearbox and alternator). This can clearly be seen in figure 1.1 where also the low frequency tilt of the spectrum is seen, making the noise very boring and tiresome.

Our measurements show that the A-weighted [1] noise level is 85–95 dBSPL(A) in most aircrafts of this type, thus requiring noise attenuating headsets to avoid hearing loss [6] and improve radio and intercom speech intelligibility. During typical cruise in a small, 4-cylinder airplane, engine speed is around 2300 rpm, corresponding to about 38 Hz. The tonal peaks are harmonics of this fundamental tone, for example a two-bladed propeller gives a peak at twice fundamental frequency along with additional harmonics. Except for take-off and landing, the propeller (and engine) rpm is normally held constant, with only slow variations.
Figure 1.1: PSD of free field cabin noise recorded in a Piper PA-32 aircraft.

1.2 Advantages with active noise reduction

There are several advantages with active noise reduction, both for noise comfort and for hearing protection. Comfort is enhanced by attenuating low frequency noise which is very tiresome and thus affecting safety. ANR helps by reducing the residual noise level and making the audio spectrum more flat. Another advantage is that the communication speech intelligibility is enhanced, since the voice audio does not get drowned in noise.

1.3 How ANR works

Active noise reduction systems are based on superposition, where the unwanted primary noise (from engine, propeller, wind noise etc) is combined with an "anti-noise" secondary signal to achieve destructive interference [2] within a spatially limited "quiet zone", leading to a lower overall noise level inside the headset cups, see figure 1.2.

As seen in figure 1.3, the ANR controller feeds an output signal $y(n)$ through the ANR speaker to generate the anti-noise secondary signal that gets combined with the unwanted noise $d(n)$ to form the residual noise $e(n)$ at the acoustic summing junction around a microphone situated in each headset cup. The dynamics of the primary (cockpit, headset cups) and secondary (power amplifier, speaker, acoustics inside cups) paths are modeled by the linear systems $P(z)$ and $S(z)$ respectively. The feedforward block
contains algorithms, tachometers and sensors to analyze the ambient noise and provide synthesized reference signals to the ANR controller.

In general, the size of the quiet zone is limited by the noise frequency to about 1/10 of the wavelength [7]. The headset cup inside dimensions are less than 100 mm in all directions, which means the highest noise frequency that still gives full attenuation inside the whole cup is 1 m, corresponding to a frequency of around 340 Hz. As seen in figure 1.1, all major tonal components are below 200 Hz, thus fulfilling the quiet zone requirements.

![Figure 1.2: Primary (unwanted) and secondary (generated) noise get combined through superposition giving a lower total level.](image)

### 1.4 Feedforward vs feedback

There are two basic ways of designing the ANR controller and thus generating the anti-noise signal: feedforward and feedback. The main external difference is whether any signal path is present from the primary noise source to the ANR controller. See figure 1.4 where $P(z)$ is the transfer function from the primary noise source $x(n)$ to the error microphone with output signal $e(n)$.

The feedforward case can be seen as a system identification problem where $P(z)$ is the unknown system that $W(z)$ should converge to [2]. The feedback case can be seen as a general prediction problem, where the controller converge to a solution that estimates the future values of $d(n)$ [2].

Feedforward ANR generally has a causality constraint, meaning that the part of the primary noise going through $W(z)$ must reach the error micro-
Figure 1.3: Basic ANR system inside headset cups (only one side shown). \( x(n) \) is the primary (ambient) noise which gets filtered through \( P(z) \) to form the unwanted noise \( d(n) \) inside the cup. \( y(n) \) is the controller output, \( e(n) \) is the combined primary and secondary noise picked up by the ANR microphone (acoustic summing junction). \( S(z) \) is the secondary path transfer function, including ANR speaker, power amplifier and acoustic enclosure.

phone no later than the noise going through the \( P(z) \) path [2], otherwise it is impossible to design a realizable controller giving optimal broadband performance.

Feedback ANR has no particular causality constraint, but larger control loop delay gives degraded performance (decreased attenuation and lower bandwidth) and eventually leads to stability problems.

In this thesis, \( x(n) \) is available to the DSP through external circuitry, so the feedforward approach may be used.

1.5 Broadband vs narrowband

Depending on application, the controller can be implemented to attenuate either broadband or narrowband (tonal) noise. Broadband controllers work on the whole frequency range of interest while narrowband controllers only
Figure 1.4: Feedback ANR (left) and feedforward ANR (right). $x(n)$ is the primary noise (e.g., in aircraft cabin). $P(z)$ is the transfer function from the primary noise to the inside of the headset cups, where $e(n)$ is the residual noise in the cups. $W(z)$ is the ANR controller creating the anti-noise signal $y(n)$. Secondary path dynamics are not included.

attenuate noise at given frequencies with a narrow bandwidth but on the other hand may give better noise suppression and robustness. As seen in figure 1.1, the present noise spectrum has strong narrowband peaks which makes the narrowband ANR controller suitable, along with an already implemented broadband analog ANR controller (not covered in this thesis). This thesis will therefore focus on a feedforward narrowband ANR system.

1.6 Previous work

Peter Rybing made a master thesis at KTH in 2003 about an Active Noise Control headset for use in home environment using feedforward ANR implemented on a floating point DSP in an open earphone type headset [8]. The aim was to attenuate broadband noise from a noise source (e.g., a vacuum cleaner) while not affecting wanted signals (e.g., speech). Performance was quite low due to the low coherence between the reference noise signal and the noise near the headset.

Sven Johansson made a doctoral dissertation in 2000 regarding active control of propeller-induced noise in aircrafts, including multiple actuator systems [9]. The last paper included in his dissertation is about a hybrid ANR headset, resembling the design presented in this thesis but the digital ANR is only implemented in Matlab and no real world performance was analyzed with digital ANR.

Texas Instruments has several application reports regarding active noise reduction in general, for example the application report [10]. These are often made in a broader sense and mostly considering the usual educational ANR setup with an acoustic duct instead of a specific application such as a headset.

1.7 Novelty

The novel part of this thesis is to develop a robust headset that works in the real world in real aircrafts and not only in a lab environment, as opposed to many previous papers. I have used more than 30 different airplanes and helicopters for field testing and validation to make sure the headset works as expected in all situations. Unfortunately, all design and implementation details can not be shown in this report to avoid disclosing any proprietary information.

The narrowband ANR is implemented in a parallel fashion where each frequency of interest has its own sets of taps and step size, thus giving optimal performance where each part can adapt at an optimal rate. A special algorithm, together with various sensors, keeps track on the current noise field in the aircraft cabin and determines which frequency component peaks to attenuate. Only peaks that are strong enough to be audible get selected.
Chapter 2

Problem definition

2.1 Problem definition

The project aim is to design and implement digital narrow band Active Noise Reduction (ANR) functionality into a personal hearing protection headset using a Texas Instruments fixed point TMS320C5509A [11] DSP and an AIC23a [12] CODEC. The intended use is in small aircrafts where the cabin sound pressure level is relatively constant and the periodic audio components from the engine, propeller etc are stationary or slowly varying. The headset should be able to attenuate up to five simultaneous tonal components between 75–400 Hz, narrowband attenuation should be at least 15 dB on the strongest tonal component and adaptation time should be less than three seconds.

2.2 Non-disclosure of proprietary information

Since the headset will be launched as a commercial product, this thesis text will only cover the basic functionality of the system. There is a great step between a headset that works in a lab environment compared to a robust headset that works in the real world. Great effort has been put into both hardware and software implementation to avoid artifacts like clicks, pops and unwanted noises and also make sure the headset works as expected during all phases of the flight (engine start, taxing, take-off, ascent, cruise, descent, landing etc). All these details has deliberately been left out to not disclose any proprietary information.
Chapter 3

Theory

As seen in figure 1.3, the ANR system is a closed loop control system, with the microphone as error sensor, the filter (either analog or digital) as the controller and the speaker as the actuator. This means that the system can be designed with standard control theory.

3.1 Adaptive filtering

In order to maintain a stable destructive interference, the digital filter needs to be constantly updated to accommodate for a non-perfect system model or that the system changes over time, e.g. changes in headset position on the head or different users wearing the headset. There are several ways to perform this update, but the most common way is to use some variant of the steepest descent algorithm [2].

In general, an adaptive filter consists of two parts: a digital filter to perform the filtering process and an adaptive algorithm to adjust the coefficients of the filter in order to achieve optimal performance. The filter is usually of finite impulse response (FIR) or infinite impulse response (IIR) type. This thesis will only focus on FIR filters and variants thereof. One advantage with FIR filters is that they are always stable, due to only zeros and no poles (but the complete system may however become unstable since it contains both poles and zeros).

Referring to figure 3.1, for an FIR filter of length $L$, the reference input vector $x(n)$ and coefficient vector $w(n)$ at time $n$ are constructed as

$$x(n) = [x(n) \ x(n-1) \ \ldots \ x(n-L+1)]^T$$

$$w(n) = [w_0(n) \ w_1(n) \ \ldots \ w_{L-1}(n)]^T$$

The output signal $y(n)$ at time $n$ will then be:

$$y(n) = w^T(n)x(n)$$
Figure 3.1: Block diagram of a general adaptive filter where $d(n)$ is the desired signal, $x(n)$ is the reference signal, $y(n)$ is the output from the filter $W(z)$ and $e(n)$ is the error signal.

The residual error $e(n)$ will then be defined as:

$$e(n) = d(n) - y(n) = d(n) - w^T(n)x(n) \quad (3.4)$$

### 3.1.1 Mean square error

To be able to update the filter coefficients, the current performance of the filter must be measured, otherwise it is not possible to adapt it towards a "better" set of coefficients. One such measure is the mean square error (MSE), defined as:

$$\varsigma(n) = E\{e^2(n)\} \quad (3.5)$$

where $E\{\cdot\}$ denotes expected value. In order to achieve maximum attenuation, which means lowest residual noise, the MSE should be as small as possible and the optimization should therefore strive towards as small $\varsigma(n)$ as possible.

### 3.1.2 Steepest descent

To reach the optimum set of coefficients which minimizes $\varsigma(n)$, the method of steepest descent is used, mainly since it is widely known and can be implemented on a sample-by-sample basis. The objective with this method is to adjust the filter coefficients $w(n)$ in small steps in the direction of the negative gradient of $\varsigma(n)$ to make the MSE as small as possible. The gradient is defined as the partial derive of $\varsigma(n)$:

$$\nabla \varsigma(n) = \nabla E\{e^2(n)\} = \nabla E\{e(n)e(n)\} = 2E\{\nabla(e(n))e(n)\} \quad (3.6)$$

The definition of $e(n)$ in equation 3.4 then gives:

$$\nabla e(n) = \nabla \{d(n) - w^T(n)x(n)\} = -x(n) \quad (3.7)$$

$$\nabla \varsigma(n) = -2E\{x(n)e(n)\} \quad (3.8)$$
This lets us update the filter weights in the negative gradient direction:

\[ w(n + 1) = w(n) - \frac{\mu}{2} \nabla \varsigma(n) = w(n) + \mu E \{ x(n)e(n) \} \quad (3.9) \]

where \( \frac{\mu}{2} \) is the step size for each coefficient vector update.

### 3.2 Least mean squares

To use equation 3.9, an estimate of \( E \{ x(n)e(n) \} \) must be found. A simple approach is to approximate the expected value as:

\[ E \{ x(n)e(n) \} \approx x(n)e(n) \quad (3.10) \]

which gives us the least mean squares (LMS) algorithm, widely used for adaptive filters due to its simplicity and low computational effort. Equation 3.10 then gives the following LMS update equation:

\[ w(n + 1) = w(n) + \mu x(n)e(n) \quad (3.11) \]

#### 3.2.1 LMS stability and convergence rate

In order to use equation 3.11 the step size \( \mu \) must be determined. It can be shown [2] that the step size with a no-delay secondary path is bounded by:

\[ 0 < \mu < \frac{2}{LP_x} \quad (3.12) \]

where \( L \) is the filter length and \( P_x \) is the instantaneous power of \( x(n) \). A small value of \( \mu \) will lead to slow convergence and a large value will either lead to excessive noise (filter weights will fluctuate around the true weights causing audible artifacts) or that the algorithm diverges. A compromise must therefore be made if using the standard LMS algorithm.

#### 3.2.2 Leaky LMS

Since the electro-acoustic open loop system, with microphone and speaker, inherently is AC-coupled (contains at least one null at S-plane origin), the filter weights \( w(n) \) may try to adapt to a solution containing a DC component, which can not be reproduced by the speaker. The DC component leads to practical problems such as reduced filter tap calculation headroom and numerical overflow. To make sure the filter taps does not get too large DC component, the LMS update algorithm equation 3.11 is modified as:

\[ w(n + 1) = \alpha w(n) + \mu x(n)e(n) \quad (3.13) \]

where \( \alpha \) is a constant close to, but less than, unity. This give two "forces" acting on the taps, one is the tap update \( \mu x(n)e(n) \) making the taps grow
until optimum solution is found and the other is the $\alpha$ constant which makes the taps strive towards zero. $\alpha$ must be chosen with care, since all values less than 1 (which is the standard LMS algorithm) give a higher MSE and thus higher residual noise. Another practical advantage with the leaky LMS algorithm is that the robustness is improved and excessively large tap sizes gets decreased [2], which reduces peak power delivered to the speakers, thus increasing battery life.

### 3.3 Filtered-X least mean squares

A feedforward adaptive filter considered as a system identification problem as in figure 1.4 does not really reflect the reality when it comes to active noise reduction, since the summing junction is acoustic (see figure 1.3), and therefore need to take other parts, such as loudspeakers, microphones and the acoustic environment, into account. Since electret microphones usually have very flat frequency response compared to speakers and the primary path $P(z)$ is not perfectly known, it is convenient to include also the microphone dynamics into the secondary path model $S(z)$ and the primary path $P(z)$, leading to the same result (but with simplified calculations) as if the microphone dynamics had been placed after the summing junction in the model. The secondary path filters the adaptive filter output $y(n)$ to $y'(n)$ as seen in figure 3.2. The addition of $S(z)$ makes the error signal $e(n)$ no longer look like in equation 3.4, but rather like:

$$e(n) = d(n) - y'(n) = d(n) - s(n) * y(n) = d(n) - s(n) * [w^T(n)x(n)] \quad (3.14)$$

which makes the LMS update equation 3.11 change to:

$$w(n + 1) = w(n) + \mu s(n) * x(n)e(n) \quad (3.15)$$

However, $S(z)$ is normally unknown and must therefore be estimated as $\hat{S}(z)$, leading to the filtered-X least mean squares (FXLMS) update equation and block diagram in figure 3.3:

$$w(n + 1) = w(n) + \mu \hat{s}(n) * x(n)e(n) \quad (3.16)$$
3.3.1 Narrowband leaky FXLMS

Since the digital ANR in this thesis is aimed towards tonal (narrowband) components, the standard broadband LMS/FXLMS implementation can be modified into a narrowband algorithm by only using the periodic parts of the reference signal \( x(n) \). By using external circuitry monitoring the noise source, a synthesized reference signal can be created with only narrowband components and without broadband noise. This solution has several advantages such as the LMS stability constraints in equation 3.12 getting simplified since the reference signal now has a constant amplitude and therefore constant signal power, thus letting the step size \( \mu \) be constant. Furthermore, the supervising control system may select which narrowband components to attenuate, based on e.g. frequency and perceived loudness.

For a single frequency \( \omega_0 \), the narrowband leaky FXLMS reference and filter weight vectors are defined as:

\[
\begin{align*}
  x(n) & = [\sin(\omega_0) \cos(\omega_0)] \\
  w(n) & = [w_0(n) \ w_1(n)]
\end{align*}
\]  

(3.17)

(3.18)

where \( \omega_0 \) is the digital frequency of interest, giving a tap update equation:

\[
  w(n+1) = \alpha w(n) + \mu \hat{s}(n) * x(n)e(n)
\]

(3.19)

which can be expanded to:

\[
\begin{align*}
  w_0(n+1) & = \alpha w_0(n) + \mu \hat{s}(n) * \sin(\omega_0)e(n) \\
  w_1(n+1) & = \alpha w_1(n) + \mu \hat{s}(n) * \cos(\omega_0)e(n)
\end{align*}
\]

(3.20)

(3.21)

Multiple single-frequency ANR blocks (with individual \( \omega_0 \) frequencies) may be paralleled to be able to attenuate a composite signal consisting of more than one frequency component.

3.3.2 Step size constraints with secondary path

When the secondary path \( S(z) \) is present, the step size constraints in equation 3.12 will be modified and can be approximated by [13]:

\[
\text{Figure 3.3: Adaptive FXLMS filter including secondary path estimate } \hat{S}(z).
\]
where $L$ is the filter length (2 in the case of narrowband FXLMS), $P_{x'}$ is the instantaneous power of $x'(n)$ and $\Delta$ is the delay of the secondary path expressed as number of samples.

### 3.3.3 Secondary path estimation precision

As shown in [2], for small step sizes, the closed-loop transfer function from the primary noise $d(n)$ to the residual noise $e(n)$ can be approximated as:

$$
\frac{E(z)}{D(z)} \approx \frac{z^2 - 2z \cos \omega_0 + 1}{z^2 - [2 \cos \omega_0 - \mu A_s \cos(\omega_0 - \phi_\Delta)]z + 1 - \mu A_s \cos \phi_\Delta} \quad (3.23)
$$

where $A_s$ is the gain of the secondary path at $\omega_0$ and $\phi_\Delta$ is the phase difference between $S(z)$ and $\hat{S}(z)$ at frequency $\omega_0$. For small step sizes, the pole radius of equation 3.23 is:

$$
r_p \approx \sqrt{1 - \mu A_s \cos \phi_\Delta} \quad (3.24)
$$

which can only be greater than 1 (instability) if $\cos \phi_\Delta$ is negative, thus the constraints on $\phi_\Delta$ are:

$$
-90^\circ < \phi_\Delta < 90^\circ \quad (3.25)
$$

This means that it is enough to determine the secondary path phase estimate within $\pm 90$ degree of the true phase to maintain stability, although it is shown in [2] that increasing phase estimation error affects adaptation time.

### 3.3.4 FXLMS controller seen as notch filter

As shown in equation 3.23, the closed loop transfer function takes the form of a second order system, having two complex poles and zeros respectively.
Both zeros are located on the unit circle and the poles are slightly inside the unit circle and, if \( \phi_\Delta \) is zero, at the same frequency as the zeros. The transfer function therefore takes the form of a notch filter, as seen in figure 3.5. Step size affects both bandwidth and depth of the notch, where larger step size gives a wider and deeper notch, which is not always feasible as will be shown later during simulations.

![Figure 3.5: Narrowband FXLMS transfer function from primary noise \( d(n) \) to residual noise \( e(n) \) for two different step sizes.](image)

### 3.4 Identifying secondary path

In order to use the FXLMS algorithm, the secondary path \( S(z) \) must be estimated. Since we are only interested in the secondary path transfer function at discrete frequencies, a frequency analysis method as in [14] is used to identify both amplitude and phase shift at a number of frequencies and then make a linear least squares fitting of the amplitude and phase responses respectively that the DSP algorithms will use during runtime. As shown in equation 3.25, the phase response does not need to be very precisely determined so linear interpolation is good enough. A sinusoid output signal \( u(n) \) is generated by the DSP and fed through the secondary path to the error microphone resulting in signal \( e_u(n) \).

\[
\begin{align*}
  u(n) &= A_{ident} \cos \omega_0 \\
  E_u(z) &= U(z)S(z)
\end{align*}
\]
where $A_{ident}$ is output amplitude of $u(n)$ and $\omega_0$ is the current frequency to be analyzed. A leaky LMS algorithm as in equation 3.13 is used to determine amplitude and phase of $e_u(n)$ (and thus $\hat{S}(z)$) at each discrete frequency.
Chapter 4
Measuring performance

The performance of passive personal hearing protection headsets are usually measured using Noise Reduction Rating (NRR) [3] in the US or Single Number Rating (SNR) [4] in Europe. Both methods use test persons to get a subjective performance figure condensed into a single number. The attenuation is A-weighted [1] before calculating the NRR/SNR value, which means that the low frequency ANR contribution to the total attenuation is almost ignored, thus making this measure inappropriate for the active part of ANR headsets. When developing an ANR headset, an objective, non-weighted and more detailed measurement method must be used to be able to make repetitive measurements and get detailed performance figures.

4.1 Lab

A soundproof room equipped with hi-fi speakers to reproduce an arbitrary excitation signal is used for lab tests, see figure 4.1. An artificial head as in figure 4.2 is placed in the middle of the room. The head is made of solid aluminum according to [15] containing electret mic capsules mounted flush to the "ear" surface to be able to measure the residual noise inside the cups.

An audio analyzer is used to generate the excitation signal and analyze the signals from the head microphones. The most common excitation signals are e.g. pink noise, white noise, one or more sinusoids or a swept sinusoid, depending on which kind of performance to measure. For the present application (aircraft noise) single or multiple sinusoid signals, optionally embedded in noise, resemble the real-world noise appropriately and are therefore used to analyze the performance of the headset. Attenuation (also called insertion loss) is measured as the level difference between ANR on and ANR off in each frequency bin of the resulting FFT or PSD spectrum.
4.2 Aircraft

Aircraft measurements were mainly carried out by using small Knowles MEMS microphones mounted on ear plugs inserted into the wearer’s as in [16], see figure 4.3. Together with a portable digital sound recorder, the residual noise inside the headset cups is recorded for further post-processing after flight where the attenuation is calculated in the same way as for the lab case.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{soundproof_room}
\caption{Schematic view of soundproof room used for lab tests.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{artificial_head}
\caption{Artificial head used for lab tests.}
\end{figure}
Figure 4.3: In-ear microphones used for aircraft tests.
Chapter 5

Simulation

5.1 Matlab

The narrowband FXLMS algorithm described in equations 3.20 and 3.21 was tested with a single sinusoid embedded in white noise giving a signal-to-noise ratio of 20 dB, since a pure sinusoid with no noise gives an unrealistically large attenuation during simulation (in the real world, there is always noise present).

Unity reference signal amplitude, step size of 0.03 and 0.12, secondary path delay of 11 samples, disturbance frequency of 100 Hz and a sample rate of 1 kHz gives attenuation results as shown in figure 5.1. The disturbance signal peaks at 0 dB and step size 0.03 gives an attenuation of 22 dB of the 100 Hz peak. When step size is increased to 0.12, the peak attenuation is increased to 41 dB, although at the cost of excessive out-of-band noise making the overall (broadband) level higher than with the small step size.

Residual error signal power and tap sizes as a function of time are shown in figure 5.2, where the larger step size gives higher error magnitude in the first graph as a consequence of the excessive tap adjustments seen in the third graph. It can also be seen that the adaptation time (time for each tap to reach $1 - e^{-1}$ of “true” value) with step size 0.03 is approximately 200 ms.

The effect of step size was further simulated by using the same parameters as above, but varying the step size in smaller amounts and measuring both the overall (broadband) and narrowband attenuations. Results are shown in figure 5.3. The narrowband attenuation increases with increased step size up to a certain point but the broadband residual level is approximately constant at step size 0.03–0.11, which means the out-of-band noise is increased. The algorithm finally diverges at a step size of approximately 0.14. This is a bit higher than the theoretical step size defined in equation 3.22 which yields an upper bound of 0.08. The reason is probably that the theoretical value is an approximation and made for wideband FXLMS systems, with up to hundreds of taps.
**Figure 5.1:** Matlab simulation PSD of disturbance and residual error signals. The error signals are the residual "noise" inside the headset cups.

**Figure 5.2:** Matlab simulation of LMS algorithm. Error magnitude squared and tap magnitudes at two different step sizes. Sample rate is 1 kHz. With step size 0.03, algorithm has converged after approximately 200 ms.
Figure 5.3: Matlab simulation of LMS algorithm. Broadband and narrowband residual levels as a function of step size. Algorithm starts to severely diverge at step size 0.14.
Chapter 6

Hardware and software implementation

6.1 Hardware

The system was first implemented using a custom designed PCB containing all electronic parts (DSP, CODEC, flash memory, ANR microphone and speaker amplifiers etc). After verification, the hardware was re-designed to fit inside the headset cups. Debugging was done through the JTAG port of the DSP in combination with an asynchronous RS232 port connected to a custom written software running on a PC.

The main parts of the hardware are:

- DSP (TI TMS320C5509A)
- CODEC (TI AIC23a)
- Flash memory
- Serial port
- Microphone amplifier
- Audio power amplifier

6.2 DSP software

The DSP software was developed using Texas Instrument Code Composer Studio, which is an Integrated Development Environment, including editor, compiler and debugger with JTAG support. All code is entirely written in C from scratch, except for the device drivers for the serial port and the codec, which are from TI:s libraries.
The codec and its built-in anti-alias and reconstruction filters are configured to run in 16 bit stereo mode at 8 kHz sample rate, see figure 6.2. The DSP is configured to read audio packets from the codec consisting of 8 stereo samples via DMA into RAM and then trigger an interrupt taking care of all audio processing. Since the highest frequency of interest is considered to be 400 Hz, the first processing step is to decimate the input audio. According to Shannon’s sampling theorem, the bandwidth of a sampled signal must be less than half the sampling frequency, so decimation by a factor 8 is suitable giving a processing rate of 1 kHz. Due to the narrowband implementation of the ANR functionality, it is more suitable to implement the decimation filters as band-pass filters centered around each frequency of interest instead of low-pass filters (which is most common). The algorithm creates biquadratic (two poles, two zeros) filters for each frequency component of interest and filter the incoming audio signals before decimation. The decimation filter coefficients are continuously updated to track the instantaneous frequency components in the noise inside the headset cups. This approach helps the ANR algorithms by filtering out signal components that are far from the center frequency.

ANR processing is performed at a rate of 1 kHz and the taps are updated on each iteration. Each frequency component has its own set of taps and step size which makes it possible to let each component adapt at an optimal rate.

The anti-noise generation is implemented through sine table lookup with a phase accumulator for each frequency component. By looking up 8 con-
secutive samples per iteration we get the upsampling action "for free" and no interpolation filters are needed (since we haven't actually made any up-sampling at all). Secondary path delay (phase angle) is looked up from the estimated and linearized response, as previously described, and added as an offset to the phase accumulator. This offset makes it possible to use a very fine resolution of the secondary path delay and also saves memory space, since no old reference values need to be stored.

6.2.1 Secondary path estimation

Secondary path transfer function, including decimation bandpass filters, was estimated as described in the Theory section. Figure 6.3 shows the phase response, which is also converted into corresponding group delay. Magnitude is not shown here, since (as shown in equation 3.25) it is only the phase response that affects the FXLMS algorithm stability. Since all frequencies of interest are below the resonance frequency of the speaker, the phase response can be adequately approximated by a linear function which is also shown in figure 6.3 by the group delay being almost constant at 11 samples regardless of frequency.

6.2.2 Step size and leak factor

To avoid excessive audible artifacts and ensure algorithm stability, the step size and leak factors described in the Theory part can not be constant values for all frequencies and circumstances. Both parameters are therefore functions of several variables such as the current noise spectrum shape, perceived sound level and the present type of aircraft (single engine, twin engine, helicopter) carefully designed during in-flight evaluations.
6.3 Windows software

To be able to control, configure and monitor the DSP, a custom Windows software was developed using Borland C++ Builder. The program uses the RS232 interface to communicate with the DSP using a custom protocol giving access to relevant DSP data structures and setting parameters in runtime, for example the step size and leak factors.
Figure 6.4: Windows software screen shot when connected to a headset during flight monitoring various algorithm signals. Top-left is noise inside cups, top-right is anti-noise output, middle-left is one set of taps.
Chapter 7

Results

7.1 Lab tests

An ambient sound consisting of five simultaneous frequency components (80, 120, 160, 200, 240 Hz) with a total level of 86 dBSPL(A) was used for test in the sound-proof room. The sound level is in the range of that of a normal aircraft. As can be seen in figure 7.1, the narrowband attenuation is 27 dB at 80 Hz dropping to 20 dB at 200 Hz and 11 dB at 240 Hz. The seemingly great number of harmonics over 200 Hz are due to the ambient loudspeakers and not the ANR processing. The 40 Hz peak is a sub-harmonic produced by the ambient loudspeakers.

The headset has no problems with five tones and the adaptation time is not affected by the number of tones, since each tone is processed in parallel by the DSP and adapt at its own optimal rate. The overall adaptation time was measured by turning the ambient sound on and off and measure the overall level. The off time is long enough to let the algorithm bleed to zero. Result is shown in figure 7.2 showing that the algorithm has converted 400 ms after the ambient sound reached final level. The 50 ms slope starting at time zero is due to the audio generator ramping up the generated sound to avoid damaging the ambient loudspeakers.

7.2 Aircraft tests

Aircraft tests have been carried out in over 30 different aircrafts, two of them will be presented here.

7.2.1 Single engine aircraft

The aircraft shown in figure 7.3 is a Piper PA-32 with a six cylinder engine and a two-bladed constant speed prop and is a typical aircraft where ANR headsets are suitable. Figure 7.4 shows the sound pressure level with and
Figure 7.1: Residual noise level inside headset cup with and without digital ANR. Measured on artificial head in sound-proof room with ambient sound consisting of five simultaneous frequency components with a total level of 86 dB SPL (A).

Figure 7.2: Sound pressure level as a function of time to show ANR adaptation time when exposed to ambient sound consisting of five simultaneous frequency components with a total level of 86 dB SPL (A). Sound is started at time zero, ramped up during 50 ms (to avoid damaging the ambient loudspeakers) and algorithm has converged fully after another 400 ms.
without ANR at the ear entrance when flying at normal cruise speed and altitude. Four major tonal components can be seen in the low-frequency region at approximately 40, 80, 120 and 160 Hz. The lowest frequency is at 40 Hz, where the human hearing is not very sensitive so this component is left out by the DSP algorithm. The 80 Hz component has largest amplitude and gets attenuated by approximately 17 dB. The residual spectrum shows almost no tonal components left below 400 Hz. The subjective impression is that the engine noise is almost completely removed and the noise sounds more as from a jet aircraft.

Figure 7.3: Piper PA-32, single engine airplane used for testing.

Figure 7.4: Sound pressure level at ear entrance in a Piper PA-32 airplane.
7.2.2 Twin engine aircraft

The Cessna 310 twin engine aircraft shown in figure 7.5 was also used for testing. The in-ear sound pressure level in figure 7.6 shows a peak attenuation of 17 dB on the strongest component and 13 dB on the smaller peak located at approximately 150 Hz.

Figure 7.5: Cessna 310, twin engine airplane used for testing.

Figure 7.6: Sound pressure level at ear entrance in a Cessna 310.
Figure 7.7: Headsets in use during approach to Bromma airport in Stockholm.
Chapter 8

Conclusions

The results show that the system works well in the lab and also in flight, fulfilling the performance requirements stated in the problem definition; Narrowband attenuation should be at least 15 dB, result shows 17 dB and five simultaneous frequency components are attenuated in the lab (no aircraft tests made with five major frequency components since no such aircraft has been found yet). Adaptation time is less than one second.
Chapter 9

Discussion

The theoretical maximum step size given in equation 3.22 yields an upper bound of 0.08, while the simulations showed that the algorithm was stable (although generates artifacts) with step size up to 0.14. The reason for this discrepancy is probably that the theoretical value is an approximation made for wideband FXLMS systems, with up to hundreds of taps, instead of only two taps as used in my algorithm.

The lab tests showed a peak attenuation of 27 dB while aircraft results showed a peak attenuation of 17 dB (which nevertheless is within the specifications). This is due to vibration components in the aircraft, which gets picked up by the ANR microphones and converted into corresponding electrical signals, affecting algorithm performance. Great effort has been put into mitigating these kind of issues in the final product. This is only one of the big difference between laboratory work and real-world implementations.

The adaptation time in the lab tests and simulations show a difference of a factor two. This is due to that the simulations were performed on the FXLMS algorithm with leak factory set to unity, while the lab tests were performed with a leak factor of less than unity.

When it comes to consumer products, hard facts are not always the selling point. Regarding aviation headsets, many pilots put greater value on, for example, a comfortable fit or just that it looks nice instead of buying it because it is the highest attenuating headset on the market. Also, some ANR headsets may leave audible artifacts, such as hiss, high frequency noise or pops, which may be irritating even though the attenuation itself is good. What this work has shown me is that the best way of designing or buying an ANR headset is to try it out, along with competitor headsets, in real aircrafts.
Bibliography


