Onwards to phantom source generation by means of adaptive filtering

Simons, R.E.A.

Award date:
1994
Onwards to phantom source generation by means of adaptive filtering

R.E.A. Simons

Eindhoven, august 1994
Summary

Adaptive filters using the LMS algorithm have been tested on their usefulness in creating a phantom source (a loudspeaker system which creates a virtual sound source, when actual sources and listener are at fixed positions). This problem is identical to that of active noise reduction, where an unwanted primary source is cancelled by secondary sources.

After a literature study, the behaviour of adaptive filters used for emulating and equalizing transfer functions from loudspeaker to the ear and to a microphone, is examined by simulations and measurements. The standard LMS algorithm gives good results for both tasks.

Due to the placing of the adaptive filters (straight after the noise source and before the "unknown" system), a so-called filtered-x LMS algorithm has to be used for equalizing the mentioned transfer functions and in the final phantom source creation system. Standard LMS has been used to make a comparison possible and to gain insight into the deconvolution process.

Two types of signal processing cards are available. For standard LMS, the XFIR card is used. However, for filtered-x the XMOT card has to be used, since this algorithm cannot be implemented using the available hardware filters from the XFIR card. A good solution is to implement a software adaptive filter on a fast DSP.

Future research has to be carried out on the single-point filtered-x algorithm. This can then be extended to a multiple-point algorithm. Finally, the active noise reduction problem, or phantom source creation with fixed sources and listener position can be tackled.

The ultimate goal is to build a system which allows movement for the listener, while still hearing phantom sources instead of the real sources. For this, also a positioning algorithm and a detector are necessary.
Contents

1 Introduction 4

2 Generating phantom sources 6
   2.1 Schematic representation of the situation .......... 7
   2.1.1 Transfer function of the filters ............... 8
   2.2 General remarks using existing knowledge ....... 10
      2.2.1 General remarks about the loudspeaker → ear transfer
           function .................................. 10
      2.2.2 General remarks about the filtering ........ 11

3 Inverse Modelling: a review of existing methods 13
   3.1 Inverse modelling without a delay ................. 14
   3.2 Inverse modelling with a delay .................... 15
   3.3 Adaptive inverse control system ................ 16
   3.4 Plant noise .................................. 17
   3.5 Filtered-x LMS ................................ 19

4 Practical methods for deconvolution of the plant 21
   4.1 Equalization at one point using filtered-x LMS ... 21
   4.2 Single-point equalization using standard LMS .... 24

5 Multiple-point deconvolution of the plant 25
   5.1 Multiple-point equalization method ............... 25
   5.2 Multi-channel control of random sound ............ 26
      5.2.1 Analysis in the z-domain .................. 27
   5.3 Multiple error LMS for multiple-channel system ... 30
      5.3.1 Least-squares filter design ............... 30

6 Modelling 32
   6.1 Emulating a loudspeaker → ear transfer function .. 32
   6.2 Forming an inverse loudspeaker → ear transfer function ... 34
Onwards to phantom source creation by means of adaptive filtering

7 Theoretical analysis
  7.1 AFIR in the direct model
      7.1.1 Optimum Wiener filter
      7.1.2 The LMS algorithm
      7.1.3 Determining the filter quality
  7.2 AFIR in the inverse model

8 Digital signal processing using the XFIR-card
  8.1 Introduction
  8.2 Motorola DSP 56001
  8.3 Motorola DSP 56200 CAFIR
  8.4 Maximum number of taps available
  8.5 Cascade interface
  8.6 DC-tap

9 Digital signal processing using the XMOT-card
  9.1 Introduction
  9.2 Special application: filtered-x

10 Simulations using the standard LMS algorithm
  10.1 Simulation method
      10.1.1 Generation of the input signals
      10.1.2 A simulation run
  10.2 Real-time processing
  10.3 Emulating a loudspeaker → microphone transfer function
      10.3.1 Measured system parameters
      10.3.2 Adaptation constant $\alpha$
      10.3.3 Adaptive filter length $I$
      10.3.4 Additive noise at the primary input of the adaptive filter
      10.3.5 Additive noise at the reference input of the adaptive filter
  10.4 Inverse filtering
  10.5 Equalizing a loudspeaker → microphone transfer function
      10.5.1 Adaptation constant $\alpha$
      10.5.2 Modeling delay $\Delta$
      10.5.3 Adaptive filter length $I$
      10.5.4 Additive noise at the reference input of the adaptive filter
  10.6 Conclusions from the simulations

11 Simulations using the pseudo filtered-$x$ algorithm
  11.1 Introduction
  11.2 Real-time processing
Chapter 1

Introduction

When a listener is placed in front of two loudspeakers serving as sound sources, then due to the physical nature of propagation of acoustic waves, the listener will perceive sound. This sound contains directional information, so the listener is capable of sensing the direction from which the sound signals are coming.

However, it is also possible to adjust the sound which is made by the two positioned loudspeakers, in such a way, that the sound appears to be coming from another direction. In this case, one speaks of “phantom sources”: sources which are not located where they are perceived to be.

The adjustment of the original signals happens by means of electronic filtering (nowadays using Digital Signal Processors (DSPs)).

Schroeder and Atal [17] developed a method for creating virtual sound images in which two loudspeakers were able to produce one imaginary source, as they were trying to compensate for unwanted crosstalk in their studies of concert-hall sound.

Damaske, who spoke of “TRADIS” (True Reproduction of All Directional Information by Stereophony), concentrated in his research on phantom sources and ways of generating them. He developed more practical methods for creating phantom sources, used for the reproduction of stereophonic sound transmission [7] and [8].

In the world of active noise reduction, unwanted sound (for example motor noise in cars and machines in factories) has to be damped. One very interesting principal for doing this is the creation of “anti-sound”, which is nowadays used extensively.
Until now, the main concern was controlling the sound field of rotating machines where the primary field is nearly periodic (one fundamental frequency and a number of acoustic modes depending on the enclosure). Reduction of noise has also been investigated, but primary sources (machines), secondary sources (additional loudspeakers) and error sensors (microphones placed at locations where damping is needed) were all at fixed positions. A person in a room, with his ears as error sensors, forms a new problem since that person generally moves his (head)position. Now, damping is wanted at the position of the ears.

If one carefully compares the methods for creating a phantom source and the method of cancelling unwanted noise, then it becomes clear that they are very much alike. Using secondary sources to compensate for unwanted primary sources is merely a 180° phase shift away from generating wanted phantom sources. (As an example: use an active noise control method to cancel a primary source. If the situation is as wanted, than freeze the system. Take away the primary source and apply a phase shift of 180° to the input of the control system. A phantom source is then realized.)

Research with fixed filters ([5], [7], [8] and [17]) showed that phantom sources can be created, but are very hard to realize due to head movement and modifications to measurement setups, when compared to a reference reproduction setup.

In an anechoic room, the problem reduces to generating silence at specified places, an engineering problem. In normal situations more factors play a role, for example the feeling that a sound source is placed at the location where a loudspeaker is situated (psycho-acoustic).

The object of this study is to solve the engineering problem, which means modelling in a practical way (obtaining the correct ear input signals and no concern about their interpretation). For simplicity, the acoustic features of the listening room (very long impulse responses with reflections and reverberations) are excluded in this report.
Chapter 2

Generating phantom sources

In order to create a phantom source, $LS_{RPh}$, the Right signal $V_R$ is processed according to figure 2.1. First it passes the adaptive filters $W_{L2}$ and $W_{R1}$, and then enters the real sources $LS_L$ and $LS_R$. Thereafter comes the path of “free space". Finally, the sound arrives at the listener.

For the sake of clarity, this picture contains only one phantom source. In the real situation there is an additional one on the left side of the listener, $LS_{LPh}$, so the set-up will be symmetrical. The transfer functions for the left phantom source are $H_{L1}^\alpha$ and $H_{Lr}^\alpha$. Its input signal is the Left signal, $V_L$. 
2.1 Schematic representation of the situation

The model in figure 2.2, shows the desired situation (upper part) and the available situation (lower part). Generating the desired transfer functions $H^\beta$ and cancelling for the available transfer functions $H^\alpha$ is the task of the adaptive filters. The transfer function of each adaptive filter will be calculated in the next section.
2.1.1 Transfer function of the filters

From figure 2.2, one can see the form of the adaptive filters' transfer functions. \( V \) denotes an electronic signal and \( P \) the sound pressure of the ear. Furthermore, \( \alpha \) and \( \beta \) give the directions of the real and desired sources respectively. Assume perfect adaptation, so the optimum solution \( W = W^* \) holds.

\[
\begin{align*}
P_l^\beta &= H_{L_1}^\beta V_L + H_{R_1}^\beta V_R \\
P_r^\beta &= H_{L_r}^\beta V_L + H_{R_r}^\beta V_R \\
V_{L_P} &= W_{L_1}^\ast V_L + W_{L_2}^\ast V_R \\
V_{R_P} &= W_{R_2}^\ast V_L + W_{R_1}^\ast V_R \\
P_l^\alpha &= H_{L_1}^\alpha V_{L_P} + H_{R_1}^\alpha V_{R_P} \\
P_r^\alpha &= H_{L_r}^\alpha V_{L_P} + H_{R_r}^\alpha V_{R_P} \\
P_l^\alpha &= H_{L_1}^\alpha W_{L_1}^* V_L + H_{R_1}^\alpha W_{R_2}^* V_R + H_{R_1}^\alpha W_{R_2}^* V_L + H_{R_1}^\alpha W_{R_1}^* V_R \\
P_r^\alpha &= H_{L_r}^\alpha W_{L_1}^* V_L + H_{R_r}^\alpha W_{L_2}^* V_R + H_{R_r}^\alpha W_{R_2}^* V_L + H_{R_r}^\alpha W_{R_1}^* V_R \\
P_l^\alpha &= (H_{L_1}^\alpha W_{L_1}^* + H_{R_1}^\alpha W_{R_2}^*) V_L + (H_{L_1}^\alpha W_{L_2}^* + H_{R_1}^\alpha W_{R_1}^*) V_R \\
P_r^\alpha &= (H_{L_r}^\alpha W_{L_1}^* + H_{R_r}^\alpha W_{R_2}^*) V_L + (H_{L_r}^\alpha W_{L_2}^* + H_{R_r}^\alpha W_{R_1}^*) V_R \\
\end{align*}
\]

Pressures in each ear should be equal, so \( P_l^\alpha = P_l^\beta \) and \( P_r^\alpha = P_r^\beta \), (let these ear pressures be denoted \( P_l \) and \( P_r \)) so for the transfer functions the next equations hold

\[
\begin{align*}
H_{L_1}^\beta &= H_{L_1}^\alpha W_{L_1}^* + H_{R_1}^\alpha W_{R_2}^* \\
H_{R_1}^\beta &= H_{L_r}^\alpha W_{L_2}^* + H_{R_r}^\alpha W_{R_1}^* \\
H_{L_1}^\alpha &= H_{L_1}^\alpha W_{L_1}^* + H_{R_1}^\alpha W_{R_1}^* \\
H_{L_r}^\alpha &= H_{L_r}^\alpha W_{L_1}^* + H_{R_r}^\alpha W_{R_2}^* \\
\end{align*}
\]

This gives

\[
\begin{align*}
W_{R_1}^\ast &= \frac{H_{R_1}^\beta H_{L_r}^\alpha - H_{R_r}^\beta H_{L_1}^\alpha}{H_{R_1}^\beta H_{L_r}^\alpha - H_{R_r}^\beta H_{L_1}^\alpha} \\
W_{R_2}^\ast &= \frac{H_{L_r}^\beta H_{L_1}^\alpha - H_{L_r}^\beta H_{R_1}^\alpha}{H_{R_1}^\beta H_{L_r}^\alpha - H_{R_r}^\beta H_{L_1}^\alpha} \\
W_{L_1}^\ast &= \frac{H_{L_r}^\beta H_{R_1}^\alpha - H_{L_1}^\beta H_{R_1}^\alpha}{H_{R_1}^\beta H_{L_r}^\alpha - H_{R_r}^\beta H_{L_1}^\alpha} \\
W_{L_2}^\ast &= \frac{H_{R_1}^\beta H_{R_1}^\alpha - H_{R_1}^\beta H_{R_1}^\alpha}{H_{R_1}^\beta H_{L_r}^\alpha - H_{R_r}^\beta H_{L_1}^\alpha} \\
\end{align*}
\]

for the converged, adaptive filters.
Onwards to phantom source creation by means of adaptive filtering

Suppose there is left-right symmetry (of the head and the position of the listener) then $H_{Li}^\beta = H_{Rr}^\beta$, $H_{Li}^\alpha = H_{Rr}^\alpha$, $H_{RL}^\beta = H_{Lr}^\beta$ and $H_{RL}^\alpha = H_{Lr}^\alpha$ hold, giving the following simplification

$$W_{L1}^* = W_{R1}^* = \frac{H_{RL}^\beta H_{Rr}^\beta - H_{Rr}^\beta H_{RL}^\beta}{(H_{RL}^\alpha)^2 - (H_{Rr}^\beta)^2}$$

$$W_{L2}^* = W_{R2}^* = \frac{H_{RL}^\alpha H_{Rr}^\beta - H_{Rr}^\beta H_{RL}^\alpha}{(H_{RL}^\alpha)^2 - (H_{Rr}^\beta)^2}$$

All of the transfer functions can be measured, since the positions of the phantom sources, the real sources and the position of the listener are known. (This means, all $H^\alpha$ and $H^\beta$ functions are available.) So by mathematical manipulation, such as squaring, subtraction, inversion and multiplication, the optimum transfer functions for the adaptive filters can be calculated.

What is meant by the “to ear” transfer function is that transfer function from a source to a certain reference point within the ear. This may be an arbitrary point in the auditory channel, as the properties of the outer ear play a role in all of the transfer functions.

It should be noted that every individual human has unique ears, and so unique transfer functions. Differences between human individual transfer functions can easily be heard! Nevertheless, initial calculations and measurements are carried out on an artificial head (with mean transfer functions) as this procedure creates less laborious and more reproducible situations. Subsequent experiments can be carried out with real subjects. This is sometimes unpleasant due to the conditions under which experiments take place: wires, microphones etc. attached to the head, with the person sitting very still. Trying to find volunteers is an additional problem!
Onwards to phantom source creation by means of adaptive filtering

2.2 General remarks using existing knowledge

2.2.1 General remarks about the loudspeaker → ear transfer function

The sound has to travel a significant distance through the air, from loudspeaker to ear necessitating a significant delay in the application of the transfer function, equal to the distance from loudspeaker to ear divided by the speed of sound.

\[ \Delta_{\text{plant}} = \frac{\text{distance loudspeaker} \rightarrow \text{ear}}{c} \]

In which \( c \) is the speed of sound at 343 m./sec.

As a rule of thumb, for loudspeaker \( \rightarrow \) ear distances in the range of 3 - 15 m. the air behaves as a non-dispersive channel for audio frequencies. Therefore, the time delay is frequency independent and can be seen as a pure delay [3]. For example:

\[
\begin{align*}
\text{distance loudspeaker} \rightarrow \text{ear} &= 5 \text{ m.} \\
c &\approx 343 \text{ m./sec.}
\end{align*}
\]

\[ \Delta_{\text{plant}} \approx 14.6 \text{ msec.} \]

It is important to take this time delay into account in the modelling stage. If an adaptive filter is being used, it tries to decorrelate input and error signal. Since two in-time shifted signals are less correlated, the adaptive filter could come up with an incorrect solution! (due to finite filter length: if very large filter lengths are available, no problem occurs, since the adaptive filter will form the actual delay by itself, using a certain number of filter taps)

Some important features of the loudspeaker → ear magnitude characteristics are [19]:

- The more flat the magnitude characteristic, the shorter the impulse response of the loudspeaker. This is important for the filter length (especially critical at low frequencies where a very flat characteristic is desirable).

- It follows from loudspeaker specifications, that the sharpest "peak" is created by the decrease in the lower frequency region (if one looks on a linear scale, the start of the transfer function is the steepest part).

For the calculation of a magnitude characteristic, different methods will be discussed. Because the transfer function is non-minimum-phase, direct inversion will not work as the inverse is then an unstable system.
Onwards to phantom source creation by means of adaptive filtering

For an average loudspeaker, the impulse response fits with sufficient accuracy in an FIR filter. If the impulse response is too long, than it must be shortened (due to the filter length), which means the smoothing of a peak in the frequency domain (creating a more global characteristic).

With the available hardware, the impulse response length can be slightly less than 6 ms. [19] giving a filter length of 200 to 300 taps for the adaptive and fixed filters, using a sample frequency appropriate for audio of $f_s = 44,100$ Hz.

If the impulse response is shortened immediately, in order to fit it into an FIR filter with fixed length, then an error will occur in the lower frequency region. Since this region contains a minimal amount of directional information (due to its high wavelength), this error need not be dramatic.

2.2.2 General remarks about the filtering

Two kinds of transfer functions are very important:

- loudspeaker → ear (or microphone)
- inverse loudspeaker → ear (or microphone)

As the inverse of the loudspeaker → ear (or microphone) transfer function has to be calculated, it is clear that inverse filtering must be used. The placing of the filters is essential, see figure 2.1. The path from loudspeaker to ear (or microphone) may be seen as a physical plant, and the filters are placed before that plant!

This is not the most common situation (since the filter is mostly placed after the plant), so investigations have to be made to look for an applicable inversion method.

Emulating an unknown transfer function by an adaptive filter is performed when using echo or noise cancellation. (Emulation means best possible approximated imitation.) Emulating the inverse of an unknown transfer function is done when equalizing is necessary, e.g. for data transmission. Extra attention should be paid to the latter when the unknown transfer function is non-minimum-phase as the inverse would then be unstable. A search for good inverse function approximating methods must be carried out.

\[ H(z) \]

Figure 2.3: The plant, the model and the real situation
Figure 2.3 depicts the acoustic path from loudspeaker to ear (or microphone), and the various ways it is used in schematic diagrams throughout this report. The symbolic notation for the plant will be $H(z)$, or $h(n)$, depending on the variable being used. Then a subscript (which source ($R$ or $L$) to which ear ($r$ or $l$)) and a superscript (the angle ($\alpha$ or $\beta$)) will be added.

In fact, the plant is the path from loudspeaker to human ear. However, this human ear can be modelled as a microphone, as is the case when using dummy heads.

The plant output can be used for quality evaluation of the system or as an input to the adaptive filter control system, as will be explained later in this report.

When a real person is involved, the plant output goes into the head, where psycho-acoustical evaluation takes place.
Chapter 3

Inverse Modelling: a review of existing methods

The inverse model of a system with an unknown transfer function is, in one sense, the best fit to the reciprocal of the unknown transfer function. Sometimes the inverse model response contains a delay which is deliberately incorporated to improve the quality of the fit. Inverse modelling has useful applications: speech processing, control and communication systems. (For example when having dispersive channels where adaptive filters are used as equalizers.)

The forming of an inverse function has several names, such as equalization and deconvolution which will be used throughout this report.

In the first two sections, the general concept of inverse modelling will be explained. These are given for better and easier intelligibility of the problem. These methods cannot be implemented in the phantom source generation system due to the place of the adaptive filters (see figure 2.1), but in the more useful methods for solving the deconvolution problem these concepts can be used.

In order to sustain compatibility with the phantom source generation system in figure 2.1, the methods will be explained using pictures which treat the path from right sound source, through the right loudspeaker, to the right ear.
3.1 Inverse modelling without a delay

First some general remarks have to be made about the schematic diagrams in this report:

- AFIR stands for adaptive finite impulse response
- the input of the system is \( v_R(n) \)
- the input of the plant is \( v_{Rp}(n) \)
- the output of the plant is \( p_{r}(n) \)
- \( w(n) \) is the impulse response of the AFIR
- \( W(z) \) is the transfer function of the AFIR
- \( w^* \) is the optimum solution of the AFIR in the time domain
- \( W^*(z) \) is the optimum solution of the AFIR in the frequency domain

Let it be noted, that \( v_R(n) \) is the input of an adaptive filter placed before the plant. If there is no adaptive filter before the plant, as will be the case for the first models, than the input of the system will be \( v_{Rp}(n) \). (this can be seen as saying \( v_R(n) = v_{Rp}(n) \)). This way of defining variables ensures compatibility throughout the report.
Onwards to phantom source creation by means of adaptive filtering

One way to perform inverse modelling is depicted in figure 3.1, the unknown transfer function is generally called plant. The ability to form an inverse will generally be limited by two factors:

- Prediction type
  The plant is generally a causal system, and generally the signal $v_{R_p}(n)$ will be delayed as it goes through the physical plant. Such conditions would require the inverse to be a predictor. If the plant is a pure delay then the task of prediction can only be performed approximately when using a causal adaptive filter.

- FIR
  If the adaptive filter has a transversal structure, than it is FIR. Such an impulse response can only approximate a desired impulse response when the latter one is IIR.

3.2 Inverse modelling with a delay

In many applications however, a delayed inverse is acceptable, alleviating the need for prediction. Furthermore, no prior knowledge is needed about possible non-minimum-phase behaviour.

$$e_r(n) = v_{R_p}(n - \Delta) - v_{R_p}(n) \ast h_{R_r}^\alpha(n) \ast w(n) \rightarrow W^*(z) = z^{-\Delta}(H_{R_r}^\alpha(z))^{-1}$$

Figure 3.2: Adaptive inverse model with delay

Figure 3.2 depicts the new situation. Inclusion of a delay of $\Delta$ samples generally permits much lower values of the Mean-Square Error (MSE) and causes the converged adaptive impulse response (after convolution with that of the plant) to approximate an impulse with delay $\Delta$. A delayed inverse is also advantageous whenever the plant is non-minimum-phase. A reciprocal transfer function would then have poles outside the unit circle. In order to be stable, the impulse response would need to be left-handed in
time (non-causal). However, a delayed non-causal impulse response can be approximated by a causal impulse response truncated in time ([6] and [20]).

The delay $\Delta$ consists out of two parts: the first part is a term due to the acoustical delay of the plant, $\Delta_{\text{plant}}$. The second term is present because of the method [20] and is in fact the chosen delay to compensate for the inverse adaptive filter.

The choice of the delay $\Delta$ is generally non-critical, but since there is an acoustical delay, $\Delta_{\text{plant}}$, it has to be larger than that one.

A good rule of thumb is [20]: $\Delta \approx \text{AFIR length/2} + \Delta_{\text{plant}}$

Let it be noted that $\Delta$ will be used for the overall delay present in the system under investigation, even if there is no delay in the plant ($\Delta_{\text{plant}} = 0$, used in simulations).

Since the adaptive filters are being placed before the loudspeakers, other methods must be used to solve the deconvolution problem.

### 3.3 Adaptive inverse control system

If the impulse response is non-minimum-phase, then the control signal (filter output signal) will be unstable, which means that the amplitude will increase, so some part of the system will get into saturation, causing loss of control.

$$e_r(n) = v_{Rp}(n - \Delta) - v_{Rp}(n) * h_{Rr}^o(n) * w(n) \rightarrow W^*(z) = z^{-\Delta}(H_{Rr}^o(z))^{-1}$$

Figure 3.3: Adaptive control system
Adaptive inverse control is based on inverse modelling techniques as discussed previously. It was developed primarily to accommodate situations where the plant might be non-minimum-phase.

When creating phantom sources, the plant is minimum-phase ([5], [10], [14] and [20]) and the filters have to be placed before it (see figure 2.1), so this method can be useful.

Figure 3.3 depicts the adaptive inverse control method. A delayed adaptive inverse (an adaptive transversal filter) forms a stable approximate inverse of the plant. The controller consists of the copy of the delayed adaptive inverse model. An unknown plant can be made to track an input command signal when this signal is applied to a controller whose transfer function approximates the inverse of the plant’s transfer function. The controller output becomes the driving signal for the plant.

In addition to the plant there is also a controller which provides an input signal to the plant. The controller receives information from the sound source and from the plant output.

### 3.4 Plant noise

![Diagram of Adaptive Control System with Plant Noise](image)

\[ e_r(n) = v_{R_p}(n - \Delta) - (v_{R_p}(n) * h_{R_r}(n) + u(n)) * w(n) \rightarrow W^*(z) \approx z^{-\Delta}(H_{R_r}^0(z))^{-1} \]

Figure 3.4: Adaptive control system with plant noise

In many practical cases, the plant to be modelled is noisy [20]. There is an extra non-white noise source in the system (see figure 3.4), which is not correlated with the desired input of the control system. The result of this will be that a “plant drift” is observed in feedback control systems that have at least one stage of integration within the control loop. The plant output contains a low frequency component superimposed on other plant outputs.
and it occurs spontaneously, not in response to plant inputs. Therefore, the final adaptive solution will be different.

From now on, this source is termed plant noise and denoted as \( u(n) \) in the figures. The output of the adaptive filter will be noisy, with a higher MSE. If \( u(n) \) has a fair amplitude then the inverse will be distorted. Generally, the optimum adaptive solution will be different to that of a delayed inverse. Furthermore, the choice for the delay, \( \Delta \) will be different [20].

The problem of plant noise has motivated the development of a new algorithm, the "filtered-x" LMS, which allows adaptation of the inverse filter placed forward of the plant in the cascade sequence (so that plant noise does not appear in the filter input). However, a new problem occurs, as can be seen in figure 3.5.

![Diagram](image)

Figure 3.5: Adaptive inverse before the plant with additive noise

If the adaptive filter is placed before the plant and the error signal \( e_r(n) \) is used directly with the LMS algorithm to adapt the inverse filter, the adaptive process is almost guaranteed to be unstable, or if not, to find an irrelevant solution, because normally the filter output is in the feedback loop and now \( e_r(n) \) is not the direct output of the adaptive filter.

If \( e_r(n) \) is to be used, then a fundamental change in the adaptive algorithm has to be made [9], [13] and [20]. This is because the feedback loop now also contains the plant and the plant noise! This new method is referred to as feed-forward control and is very popular when applied to active noise control.
3.5 Filtered-x LMS

Now two adaptive filters are being used (see figure 3.6): one (AFIR1) to generate a direct model of the plant, $\hat{H}_{Rr}(z)$, and the other one (AFIR2) is the filtered-x process that estimates the plant’s delayed inverse (with transfer function $W(z)$).

\[ e_p(n) = v_{Rp}(n) \ast h_{Rr}^{\alpha}(n) - v_{Rp}(n) \ast \hat{h}_{Rr}^{\alpha}(n) + u(n) \rightarrow \hat{H}_{Rr}^\alpha(z) \approx H_{Rr}^\alpha(z) \]

and then

\[ e_r(n) = v_R(n - \Delta) - w(n) \ast h_{Rr}^{\alpha}(n) \ast v_R(n) - u(n) \rightarrow W^*(z) \approx z^{-\Delta}(H_{Rr}^\alpha(z))^{-1} \]

Figure 3.6: Filtered-x algorithm

The unavailable plant, $H_{Rr}^\alpha(z)$, is replaced by an exact copy of the plant model, $\hat{H}_{Rr}^\alpha(z)$, in the filtered-x algorithm.

Experience has shown that $\hat{H}_{Rr}^\alpha(z)$ need not be very precise when incorporated into the algorithm [20]. The most important attribute of $\hat{H}_{Rr}^\alpha(z)$ is that its impulse response should have at least as great a transport delay as that of $H_{Rr}^\alpha(z)$ itself. (The transport delay is defined as the time from the initiating input pulse to the first nonzero output response.)
Onwards to phantom source creation by means of adaptive filtering

In figure 3.6, a delayed inverse model is used to provide the control signal $v_{Rp}(n)$ to the plant. The reference input drives the inverse model. When both adaptive processes converge individually, the impulse response of the path from $v_R(n)$ to $p^o(n)$ will be a delayed impulse. Although the two adaptive processes are not independent in a strict sense, they behave with slow adaptation as if they were independent [20].

In figure 3.6, AFIR2 stands for the adjustable filter just before the loudspeaker and $v_{Rn}(n)$ is the signal applied to the loudspeaker. After plant noise is superimposed on the plant output, the output signal arrives at the position of the human ear. So this is fed with sound pressure $p^o(n)$.

In figure 3.6, $r(n)$ is the filtered version of the input signal $v_R(n)$ and is used in AFIR2 to accommodate the fact that both inputs to the LMS algorithm have to be filtered in the same way.
Chapter 4

Practical methods for deconvolution of the plant

In this chapter, the information from the previous deconvolution methods is used in practice. This leads to useful suggestions for the phantom source generation system.

4.1 Equalization at one point using filtered-\(x\) LMS

A method for designing an equalization filter for a sound-reproduction system is considered [9]. Figure 4.1 shows a source, an adaptive filter, the electro-acoustic chain (plant) and the receiver (the microphone). In general, it is not possible to achieve a perfect inversion of the reproduction chain, since the acoustic path usually has delays and other non-minimum-phase behaviour (for example reflections). If a modelling delay (of \(\Delta\) samples) is used, a model, with filter before the plant, looks like that in figure 4.2.

![Diagram of equalization filter in a sound-reproduction system](image)

Figure 4.1: Usage of an equalization filter in a sound-reproduction system

The algorithm used, is the filtered-\(x\) algorithm, which also has the advantage that it is possible to generalize into multiple channels, necessary when using
Onwards to phantom source creation by means of adaptive filtering

Onwards to phantom souree creation by means of adaptive filtering

impulse response equalization filter

$W(z)$

impulse response reproduction chain

$H_R^a(z)$

$v_R(n)$

$v_{Rp}(n)$

modeled delay

$\sum$

$d(n)$

$e_r(n)$

Figure 4.2: Single-point equalization problem

multiple sources and receivers. Section 5.1 explains this further.

$w(n + 1) = w(n) + a r(n) e_r(n)$

is the coefficient update algorithm. For a graphical explanation see figure 3.6.

Suppose the adaptive equalization filter is FIR and has filter length $I$. Also assume that the response of the unequalized reproduction chain is a modelled FIR with filter length $J$.

The sampled source signal is called $v_R(n)$, the sampled signal fed to the loudspeaker $v_{Rp}(n)$, the sampled output from the microphone $p_r^a(n)$ and the plant impulse response $h_{Rr}^a(n)$, with coefficients $h$.

The desired situation is that $p_r^a(n) = d(n)$ holds (so $p_r^a(n)$ can be seen as an estimate $\hat{d}(n)$ for $d(n)$). Assume time invariant filter coefficients.

$$v_{Rp}(n) = \sum_{i=0}^{I-1} w_i v_R(n - i)$$

$$p_r^a(n) = \sum_{j=0}^{J-1} h_j v_{Rp}(n - j)$$

$$p_r^a(n) = \sum_{j=0}^{J-1} h_j \sum_{i=0}^{I-1} w_i v_R(n - i - j) = \sum_{i=0}^{I-1} w_i r(n - i)$$

which leads to (see figure 4.3)

$$r(n) = \sum_{j=0}^{J-1} h_j v_R(n - j)$$

the filtered reference signal
The calculation of $r(n)$ is made because transfer function reversal has to take place [10]. It is the output signal from the FIR filter onto which the plant estimate is copied in figure 3.6. This approach leads to the same result as the filtered-$x$ algorithm, which has been described earlier in a more general sense, so this can be seen as the mathematical verification of the technique showed in figure 3.6.

Written in vector form

$$p^\alpha_r(n) = d(n) = r^T(n)w$$

With

$$r^T(n) = [r(n) \, r(n-1) \, \cdots \, r(n-I+1)]$$

$$w^T = [w_0 \, w_1 \, \cdots \, w_{I-1}]$$

As an indication of how good $p^\alpha_r(n)$ approximates to $d(n)$, the $MSE$ is calculated

$$J(n) = E\{e^2_r(n)\}$$

$$= E\{(d(n) - p^\alpha_r(n))^2\}$$

$$= E\{d^2(n)\} - 2w^T E\{r(n)d(n)\} + w^T E\{r(n)r^T(n)\}w$$

With a minimum error for the optimum Wiener solution

$$w^* = (E\{r(n)r^T(n)\})^{-1}E\{r(n)d(n)\}$$

Adaptive algorithms can be used to adjust the coefficients of $w$ automatically to be a close approximation to $w^*$. 

Figure 4.3: Reversion of transfer functions
4.2 Single-point equalization using standard LMS

Another possibility is to assume that the transfer function from loudspeaker to microphone does not change (fast). In that case, after adaptation, the inverse transfer function could be copied into a fixed FIR filter, as depicted in figure 4.4.

An adaptive filter is placed after the microphone in cascade. During adaptation, switch S is turned to position a. After adaptation the weights of the converged adaptive filter (master) $W^*_m$ are transferred to an FIR filter (slave) $W_s$, just in front of the loudspeaker. Now the inverse model is complete. Next the switch S is turned to position b and adaptation is stopped. Evaluation of the inverse function can now take place.

If complexity and subsequent cost is not a consideration, the configuration of normal filtered-x LMS is preferable to the modified system: in the modified algorithm the microphone picks up signal from the loudspeaker as well as noise. This noise is an additive component to the adaptive filter and is uncorrelated with the desired response input of the system. The correlation matrix of the input signal is affected, therefore causing the solution of the adaptive process to be different to the ideal Wiener solution (when using the complete filtered-x LMS algorithm, noise does not appear in the primary adaptive filter input, see also figures 3.6 and 8.3).

![Diagram of adaptive loudspeaker system](image)

Set S in position $a$

$$e_r(n) = v_R(n - \Delta) - v_R(n) * h_R^n(n) * w_m(n) \rightarrow W^*_m(z) = z^{-\Delta}(H_R^n(z))^{-1}$$

and then

Set S in position $b$

Stop adapting, copy $W^*_m(z)$ to $W_s(z)$, the FIR filter.

Figure 4.4: Adaptive loudspeaker system
Chapter 5

Multiple-point deconvolution of the plant

5.1 Multiple-point equalization method

It is possible to obtain good equalization at a single microphone position, but the equalized response away from this point can be worse than the equalized response. A better method could be equalization at a number of points ($L$ points). The method explained in section 4.1 is used here and expanded into a multi-channel case, so that one loudspeaker and $L$ microphones are being used.

The sound pressures are denoted as $p^a(n)$ and will form estimates $\hat{d}(n)$ of the desired signal $d(n)$, as can be seen in figure 5.1.

Each microphone has its own specific delay ($\Delta_l$ samples for the $l$th microphone). The vector of output signals can now be represented as

$$e(n) = d(n) - R(n)w$$
Onwards to phantom source creation by means of adaptive filtering

where similar definitions hold

\[ e^T(n) = [e_1(n) e_2(n) \cdots e_L(n)] \]
\[ d^T(n) = [d_1(n) d_2(n) \cdots d_L(n)] \]
\[ (p^a(n))^T = [p^a_1(n) \cdots p^a_L(n)] \]
\[ R^T(n) = [r_1(n) r_2(n) \cdots r_L(n)] \quad \text{a matrix} \]

with \( r_i(n) \) and \( w \) as defined in section 4.1. By using

\[ J(n) = E\{e^T(n)e(n)\} \]
\[ = E\{d^T(n)d(n)\} - 2w^T E\{R^T(n)d(n)\} + w^T E\{R^T(n)R(n)\}w \]

as an error criterion, again there is a global minimum for

\[ w^* = (E\{R^T(n)R(n)\})^{-1} E\{R^T(n)d(n)\} \]

A generalization of the filtered-x LMS algorithm is used. The transformation of filtered-x LMS single-channel into multi-channel gives [10]

\[ w(n+1) = w(n) + \alpha R^T(n)e(n) \]

### 5.2 Multi-channel control of random sound

Creating anti-sound is very much the same as the generation of a phantom source. It can be concluded from figure 2.1 that it is possible to make the loudspeakers under angle \( \alpha \) sound similar to those under angle \( \beta \). When the adaptation is completed, one can invert the input signals to the speakers under angle \( \beta \) (180° phase shift). Now cancellation will take place due to the arriving signals being in anti-phase at the listener's position. This is the principal of anti-sound creation.

When the sound field to be controlled is not composed of a series of discrete frequencies, the additional complexity of using detection sensors to generate the reference signal has to be introduced [14]. Assuming there are \( K \) detection sensors, \( M \) secondary sources and \( L \) error sensors, the controller is a matrix of \( K \times M \) electronic filters.

In figure 5.2, a simple two-channel system for the control of random noise from two primary sources (so-called feed-forward) controller is depicted.
5.2.1 Analysis in the \( z \)-domain

**General case**

In figure 5.3 the block scheme of the feedforward controller is depicted. The signals from the primary sources are detected by \( L \) error sensors, whose outputs prior to control are represented by the vector \( d(n) \). The sound due to the primary sources is also detected by \( K \) detection sensors, whose output signals prior to control are represented by the vector \( a(n) \). These signals are passed through a matrix of electrical control filters, \( W(z) \), to produce the vector \( v_p(n) \). This represents the input signals to the \( M \) secondary sources. In general these signals from \( v_p(n) \) are fed back via the transfer function matrix \( F(z) \) and result in the corruption of the signals from the detection sensors. \( v(n) \) are the input signals for the matrix \( W(z) \) when control is applied. The transfer function matrix \( H(z) \) gives the paths from secondary sources to error sensors.
Applying multi-channel control on phantom sources

The general case dealt with in the previous section, can be simplified. When phantom sources are generated, input signals are known and can not be influenced: they are already electric signals originating from the sound source. Compare figures 5.2 and 5.4, the first one has unknown sources, whereas the second one has known sources. Furthermore, the path from primary and secondary sources to detection sensors is not present in the second picture. In other words: the feedback loop via the matrix \( F(z) \) in figure 5.3 can be deleted, when building a new block diagram for the situation depicted in figure 5.4. This also means that \( a(n) = v(n) \).

Furthermore it is known that there are two primary and secondary sources, left and right, and that there are two error sensors.

All this gives: \( K = 0 \) and \( L = M = 2 \).

When the loudspeakers are placed under angle \( \alpha \), the transfer function of the path is \( H^\circ(z) \).

The new schemes are shown in figures 5.4 and 5.5.
Suppose that the optimum filter is restricted to being causal as well as having an FIR structure. The analysis takes place in the discrete time domain. This is the most common practical case. The controller is implemented as a matrix of digital FIR filters.
5.3 Multiple error LMS for multiple-channel system

Figure 5.6: Block diagram of a feed-forward active sound control system

Figure 5.6 depicts the active sound control situation, in which two secondary sources are used to control one primary sound source. The analysis can be done by using figure 5.1 (in which the adaptive filter $W$ is replaced by two adaptive filters, and so the input to $H_i(z)$ will be a vector). $L = M = 2$ and $K = 0$ hold.

Let the output of the $l$th error sensor be $e_l(n)$. This signal consists of the primary signal $d_i(n)$ and a component created by filtering the input signal $v_R(n)$. Again the plant can be modeled as FIR with length $J$. The filter length of the adaptive filter is $I$. The signal $e_l(n)$ can be written as

$$e_l(n) = d_i(n) + \sum_{m=1}^{M} \sum_{j=0}^{J-1} h_{lmj} \sum_{i=0}^{I-1} w_{mi}(n - j)v_R(n - i - j)$$

Since $L \geq M$, the total error can be defined as

$$J(n) = E\{\sum_{i=1}^{L} e_l^2(n)\}$$

5.3.1 Least-squares filter design

If the filters are time invariant (the coefficients vary slowly compared to the timescale of the response of the system to be controlled), then $e_l(n)$ may be written as

$$e_l(n) = d_i(n) + \sum_{m=1}^{M} \sum_{i=0}^{I-1} \sum_{j=0}^{J-1} h_{imi}(n - j)v_R(n - i - j)$$
Onwards to phantom source creation by means of adaptive filtering

\[ d_l(n) + \sum_{m=1}^{M} \sum_{i=0}^{I-1} w_{mi} t_{lm}(n-i) \]

in which

\[ t_{lm}(n) = \sum_{j=0}^{J-1} h_{lmj} v_R(n-j) \]

Introducing vector notation gives

\[ e_l(n) = d_l(n) + r_l^T(n) w \]

with

\[ r_l^T(n) = [r_{l1}(n) \cdots r_{l1}(n-I+1) | \cdots | r_{lM}(n) \cdots r_{lM}(n-I+1)] \]

and

\[ w^T = [w_{10} \cdots w_{1I-1} | \cdots | w_{M0} \cdots w_{MI-1}] \]

and so

\[ e(n) = d(n) + R(n) w \]

with \( e(n), d(n), R(n) \) and \( w \) as defined in section 5.1.

By using as an error criterion

\[ J(n) = E\{\sum_{l=1}^{L} e_l^2(n)\} \]

\[ = E\{e^T(n)e(n)\} \]

\[ = E\{d^T(n)d(n)\} + 2w^T E\{R^T(n)d(n)\} + w^T E\{R^T(n)R(n)\}w \]

The optimum Wiener solution is

\[ w^* = -E\{R^T(n)R(n)\}^{-1} E\{R^T(n)d(n)\} \]

which gives for the minimum error

\[ J_{min} = J_0 - E\{d^T(n)R(n)\} E\{R^T(n)R(n)\}^{-1} E\{R^T(n)d(n)\} \]

in which \( J_0 = E\{d^T(n)d(n)\} \), the value for \( J(n) \) with no control applied.
Chapter 6

Modelling

In this chapter, the model which is used throughout the research is explained. First, two parts of the total model are explained, the emulation of a loudspeaker $\rightarrow$ ear (or microphone) transfer function and formulating an inverse of that transfer function. This is done since they form elementary parts of the total model and can be evaluated separately. By doing this, the performance of the adaptive filters can be measured.

The models are only applicable when there is a single output channel (microphone or ear). In the two-output-channel case the generalization of the filtered-$x$ algorithm must be used, since one filter will be updated with error terms from two microphones or ears.

The emulation of a transfer function is part of the filtered-$x$ algorithm, so this can be used in the two channel case.

The inverse forming can be seen as a way to get more information about the technique with respect to implementation and results.

6.1 Emulating a loudspeaker $\rightarrow$ ear transfer function

An acoustical path, $H_{R_e}$, has to be emulated electrically. This is done using a system as depicted in figure 6.1, with the adaptive filter $W$.

Since the acoustical path contains an acoustical delay, the electrical equivalent must also contain one, called $\Delta$ in figure 6.1.
The adaptive filter needs to minimize the difference signal between its own output and the desired output. In an ideal situation, $W$ will adapt to $H_{Rr}^o$, but certain practicalities cause a deviation from the ideal solution:

- $W$ must be of sufficient length (have enough taps) to form the most relevant part of $H_{Rr}^o$. This is given by: $I = f_{\text{sample}} \cdot T_H$, in which $T_H$ is the impulse response time duration of $H_{Rr}^o$ minus the acoustical delay and $I$ is the filter length of the adaptive filter $W$.

![Figure 6.1: Adaptation of a loudspeaker → ear acoustical path](image)

![Figure 6.2: The choice of $\Delta$ is critical due to its influence on $w^*(n)$](image)
Onwards to phantom source creation by means of adaptive filtering

- $\Delta$ must be chosen carefully: $W$ should be placed at the most relevant part of $H_{Rr}^a$. When $\Delta$ is too big, the first part of $H_{Rr}^a$ can not be emulated and much important information is lost. If, however, $\Delta$ is too small, the adaptive filter $W$ will emulate a useless delay and the tail of $H_{Rr}^a$ is cut off. (see figure 6.2)

- The measurement microphones in the artificial head have a finite signal to noise ratio, so at the end of $H_{Rr}^a$, a disturbing noise source is introduced.

6.2 Forming an inverse loudspeaker $\rightarrow$ ear transfer function

![Figure 6.3: Adaptation of an inverse loudspeaker $\rightarrow$ ear acoustical path](image)

This situation, depicted in figure 6.3, is slightly more complex than the previous one, as there is no direct emulation of an unknown system, but an approximation of the inverse function.

When the transfer function has a zero, then the inverse will have a pole, which results in an infinite impulse response of the latter. Therefore truncation errors will be made.

If such a zero lies outside the unit circle, then this function is non-minimum-phase. The inverse of such a function is non-causal and will have a non-causal (two-sided in time) infinite impulse response. By using an artificial delay, the adaptive filter can be shifted in time and emulate a non-causal function. Again truncation errors will be made due to the finite filter length.
Convolution of the two functions will result in a delayed Dirac pulse. Again, certain facts cause deviations from the ideal solution:

- Truncation errors will be made, since a two-sided infinite impulse response cannot be emulated with a limited number of coefficients.

- The delay, $\Delta$, needs to be chosen carefully so that the adaptive filter realizes the part of $H_{rf}^a$ containing the most energy. The optimum value can be determined by experimentation, as it is dependent on the function.

- The noise introduced by the microphone is added directly to the input of the adaptive filter, which may influence the final solution.
Chapter 7

Theoretical analysis

By using the models from the previous chapter, the most interesting properties of the adaptive filters, such as the influence of

- disturbances (noise present at reference and primary input)
- filter parameters (adaptation constant, modelling delay and filter length)

will be investigated. Theoretically obtained relationships can be verified by simulations.

Calculations will be made with generally used variables for easy readability. (So the mathematics will be slightly different to that in previous chapters, which was done with actual variables. However, one can easily convert between the two.)

7.1 AFIR in the direct model

In figure 7.1, the adaptive filter $W$ is placed in a configuration emulating the acoustical path from loudspeaker $\rightarrow$ ear (or microphone). For the sake of simplicity, this impulse response is modelled as an FIR. This approximation is fair, since the real impulse response is of finite duration, but its length rises to several $\text{msec}$.

Furthermore, the acoustical delay of the path is not taken into account, since it is not crucial for the analysis of the adaptation process.

The following assumptions hold (wide-sense stationarity)

\[
\begin{align*}
E\{i(n)\} & = 0 & i = x, s, u \\
E\{i^2(n)\} & = \sigma_i^2 & j = s, u \\
E\{i(n)j(n - \tau)\} & = 0 & \tau \neq 0 \\
E\{j(n)x(n - \tau)\} & = 0 & \forall \tau
\end{align*}
\]
Onwards to phantom source creation by means of adaptive filtering

There are two noise sources in the first schematic, which may influence the behaviour of the adaptive filter:

- \( s(n) \): not actually present, but added for educationary purposes
- \( u(n) \): noise due to the microphone’s finite signal to noise ratio and noise due to possible reflections

7.1.1 Optimum Wiener filter

If there are no disturbances, the adaptive filter will reach the optimum Wiener solution, the explanation of which follows.

The adaptive filter \( W \), shown in figure 7.1, will be treated as a fixed FIR filter of length \( I \). The plant \( H \) has filter length \( J \). Here, it is assumed that \( J = I \). For FIR filter \( W \), the optimum Wiener solution will be calculated with no additional noise sources present (so for the second schematic of figure 7.1).

The Wiener filter theory is based on minimizing the mean-square error

\[
J(n) = E\{e^2(n)\}
\]

in which \( e(n) \) is termed the error (or residue) signal, \( E\{\cdot\} \) the expected value in time and \( w = (w_0 \cdots w_{I-1})^T \) the vector with coefficients of \( W \). Generally, \( J(n) \) can be expressed as follows

\[
J(n) = \sigma_a^2 - \mathbf{R}^T_{xa} \cdot w - w^T \cdot \mathbf{R}_{xx} + w^T \cdot \mathbf{R}_{x} \cdot w
\]
Onwards to phantom source creation by means of adaptive filtering

with
\[ \sigma_a^2 = E\{a^2(n)\} \]
\[ \mathbf{R}_{xa}^T = E\{x(n) \cdot a(n)\} \]

\( \mathbf{R}_x = E\{x(n) \cdot x^T(n)\} \)

\( \mathbf{x}^T(n) = [x(n) \cdots x(n-I+1)] \)

For the optimum solution \( w^* \) this error needs to be minimal, so the derivative, with respect to \( w \), has to be set to zero. This gives
\[ w^* = \mathbf{R}_x^{-1} \cdot \mathbf{R}_{xa} \]

for the optimum Wiener filter. For the MSE this gives
\[ J_{min} = \sigma_a^2 - \mathbf{R}_{xa}^T \cdot \mathbf{R}_x^{-1} \cdot \mathbf{R}_{xa} \]

\( x(n) \) is wide-sense stationary and \( a(n) = h^T \cdot x(n) \). The next equations hold
\[ w^* = h \quad h = [h_0 \cdots h_{I-1}]^T \]

and
\[ J_{min} = 0 \]

Therefore, the optimum Wiener filter solution for the adaptive filter \( W \) in the second schematic of figure 7.1 will be exactly the same as the unknown transfer function which it has to realize.

7.1.2 The LMS algorithm

Least Mean-Squares is the name of the algorithm being used. It adapts the coefficients of \( w \) iteration-by-iteration. The final solution of the adaptive process will be of the form of a Wiener solution, since the same optimization criterion is used [18]. The algorithm for the coefficient update is
\[ w(n+1) = w(n) + 2\alpha x(n)e(n) \]

with \( \alpha \) being the adaptation constant which regulates the speed of convergence and the accuracy of the adaptation process.

On average, and after convergence of the algorithm \( (n \to \infty) \)
\[ E\{w(n)\} = h = w^* \]

holds, so the LMS algorithm converges to the optimum Wiener solution (on average).
7.1.3 Determining the filter quality

The schematic with noise sources, shown in figure 7.1, can be seen as a noise canceller: leave away the noise source $s(n)$ and introduce the desired signal $u(n)$, disturbed by the signal $a(n)$. The relative misadjustment ($\tilde{J}(n)$) is defined as [18]

$$\tilde{J}(n) = \frac{E\{e^2(n)\}}{E\{u^2(n)\}} = \frac{E\{(a(n) - \hat{a}(n))^2\}}{E\{u^2(n)\}} = \frac{J(n) - J_{\text{min}}}{J_{\text{min}}}$$

with

$$J_{\text{min}} = J(n)|_{w(n) = w} = \sigma_u^2$$

The fractional amount by which the steady state misadjustment exceeds the minimum attainable misadjustment $J_{\text{min}}$ is called the final misadjustment $\bar{J}$. After convergence

$$\bar{J} = I \alpha \sigma_x^2$$

$I$ is the adaptive filter length

holds. A necessary condition for convergence of the algorithm is that the adaptation constant, $\alpha$ is not too big. For the region of convergence

$$0 < \alpha < \frac{1}{\eta \sigma_x^2}$$

holds, with

$$\eta = I - 1 + \kappa_x \quad \sigma_x^2 = E\{x^2(n)\} \quad \kappa_x = \frac{E\{x^4(n)\}}{E^2\{x^2(n)\}}$$

In signal estimation problems, the rate of convergence ($\nu_{20}$) is defined as a quantity related to the number of iterations needed to decrease the quantity $10 \cdot \log J(n)$ by 20 dB. The next equation holds

$$\nu_{20} \approx \frac{1.15}{\alpha \sigma_x^2} \quad (\alpha \sigma_x^2 \ll 1)$$

A large $\alpha$ will result in a fast convergence, and also large values of $\bar{J}$ (thus decreasing accuracy).

If, however, the noise source $u(n)$ is not present, a different criterion will be chosen to judge the quality

$$\hat{J}(n) = \frac{J(n)}{E\{x^2(n)\}} = \frac{E\{e^2(n)\}}{E\{x^2(n)\}} = \frac{E\{(a(n) - \hat{a}(n))^2\}}{\sigma_x^2}$$

and so a steady-state error $\bar{J}$. When the length of $H(z)$ and $W(z)$ are equal, so $J = I$, then $\bar{J} = 0$ holds (a derivation can be found in the appendix, page 105).
Onwards to phantom source creation by means of adaptive filtering

The convergence region and rate of convergence will be the same.

The last quality criterion, $\bar{J}$, will be used when $J \neq I$. However, when the influence of the adaptation constant $\alpha$ is examined, $J = I$, this criterion is useless, since its value is always zero. Therefore, to test $\alpha$, the noise cancellation setup will be used with quality criterion $\bar{J}$.

**Adaptation constant $\alpha$**

It follows from the relationship $\bar{J} = I \alpha \sigma_x^2$, that with increasing $\alpha$ the accuracy after convergence will decrease.

**Adaptive filter length $I$**

If $J > I$ holds, the plant $H$ can never be emulated exactly, as there are too few taps. The optimum solution for a white noise signal $x(n)$ will be

$$ w^* = h_f \quad \text{with} \quad h_f = (h_0 \cdots h_{I-1})^T $$

The $\text{MSE}$ will be (a complete derivation can be found in the appendix, page 105)

$$ \bar{J} = \sigma_x^2 (h_I^2 + \cdots + h_{I-1}^2) $$

The influence of tap shortage will be noticed in the frequency domain, see the appendix, page 114 for an explanation.

**Noise source at the primary AFIR input**

This is also the noise cancellation setup, and the optimum Wiener solution

$$ E\{w(n)\} = h = w^* $$

holds. Since the noise source $u(n)$ is uncorrelated with the white noise signal $x(n)$, the adaptive filter will be unable to cancel it as it enters the residue signal $e(n)$, see the first schematic of figure 7.1. For the $\text{MSE}$

$$ J(n) = E\{e^2(n)\} = E\{(a(n) - \hat{a}(n))^2\} $$

is found. Which leads to

$$ \bar{J} = \frac{\sigma_u^2}{\sigma_x^2} = h_I^2 + \cdots + h_{I-1}^2 \quad \text{with} \quad \sigma_u^2 = \sigma_x^2 (h_I^2 + \cdots + h_{I-1}^2) $$

The convergence range and rate of convergence will not change due to $u(n)$. 
Onwards to phantom source creation by means of adaptive filtering

Noise source at the reference AFIR input

Since analysis of the MSE leads to an impractical result [19], the average final solution is observed. With s(n) present (see the first schematic of figure 7.1), it follows (see appendix, page 107 for a complete derivation) that

$$E\{w(n)\} = \frac{\sigma_z^2}{\sigma_s^2 + \sigma_z^2} \cdot h$$

and so

$$E\{w(n)\} \approx h \text{ if } \sigma_s^2 \ll \sigma_z^2$$

So there will be a multiplication factor in the optimum solution $w^*$, which is frequency independent.

This gives a biased solution for the frequency response, as the whole characteristic will move up- or downwards.

### 7.2 AFIR in the inverse model

![Figure 7.2: AFIR in inverse model](image)

The main difference with the direct model is that the input signal of the adaptive filter is the signal coming from the microphone rather than directly from the source $x(n)$. The original signal, $x(n)$ is white noise, but this is coloured by the "unknown" system, $H$. This has consequences for the convergence behaviour of $W$, but also means that the mathematics will be more complicated.
The expression for the optimum Wiener solution to the deconvolution problem of figure 7.2 is [20]

$$W^*(z) = \frac{z^{-\Delta} \Phi_{xx}(z) H(z^{-1})}{\Phi_{xx}(z)|H(z)|^2 + \Phi_{uu}(z)}$$

with $\Phi_{xx}(z)$ the power spectrum of $x(n)$, and $\Phi_{uu}(z)$ the power spectrum of $u(n)$.

As can be seen, the noise $u(n)$ gives an extra frequency dependent factor. (In the appendix, page 111, it is shown that if the noise source $u(n)$ was located before the plant, its influence would be a frequency-independent scaling factor).

If the situation is ideal (with sufficient delay, $W$ of sufficient length and no noise $u(n)$), then the average solution after convergence will be

$$W^*(z) = \frac{z^{-\Delta}}{H(z)}$$
Chapter 8

Digital signal processing using the XFIR-card

8.1 Introduction

The complete system being used for performing digital signal processing is depicted in figure 8.1. It consists of a PC, with the program for interactive control of user applications such as down-loading of DSP code, changing parameters and debugging, a host interface, A/D and D/A converters, a clock card, the XFIR card, the XMOT card and the LDA-ASP bus (which forms the connection between converters, clock, XFIR and XMOT cards).

![Figure 8.1: Complete digital signal processing system](image-url)

Of major importance are, of course, the XFIR (see figure 8.2) and XMOT (see the next chapter) cards, on which the actual processing takes place.

Each sample period is divided into 128 time slots. Each DSP may read from the bus during all of those time slots, but may only write to the bus during one time slot, in order to avoid bus conflicts. The best solution to avoid the
problem of reading data from a bus address, while another DSP is writing to the same bus address at the same time, is to do all I/O at the beginning of the interrupt routine (see page 54 for more explanation).

The heart of the XFIR card is formed by two Motorola DSP 56001 processors. There are also eight Motorola DSP 56200 CAFIR processors (cascadable adaptive FIR) placed on the board. For communication between the two DSP 56001 processors, there is 1 kbyte of dual ported RAM memory available. The host (PC) can communicate with the two DSP 56001 processors via the host interface and an external controller. There are 2 kbytes of dual ported RAM memory for the two DSP 56001 processors for a maximum of 16 times LDA-ASP bus access during one sample period. ($f_{\text{sample}} = 44,100 \text{ Hz}$.)

Furthermore, one DSP 56001 processor has 32 kbyte external memory, whereas the other has 16 kbyte memory and the control of the eight CAFIR chips. Three general purpose I/O pins are available for information about calculation time, to generate a trigger pulse, etc.

Hardware reset is done by pressing the button on the front side of the board. Each DSP can also be manually reset.

Hereafter, brief descriptions of the two types of Motorola processors on the
Onwards to phantom source creation by means of adaptive filtering

XFIR card will be given. Extra attention will be paid to the CAFIR chip because of the fact that this chip performs the actual filtering process.

8.2 Motorola DSP 56001

There are two processors of this type placed on the card. One is referred to as DSP 0 and the other as DSP 1. These are the processors on the card which are mainly used to control and to steer the processing. DSP 0 is in control of the eight CAFIR chips. The control and data transfer registers (16 per chip) are placed in the higher Y-memory. DSP 1 is generally used for calculations on input and output characteristics, such as mean power etc.

The clock frequency of the DSP 56001 is 27 MHz, the data bus size is 24 bit.

The DSP 56001 has three ports, which can be configured as follows:

- port A: I/O, external X, Y and P memory communication
- port B: host (PC) communication
- port C: general purpose I/O

8.3 Motorola DSP 56200 CAFIR

Special hardware filters of type DSP 56200 are being used. They have a maximum clock frequency of 10 MHz and a maximum of 256 coefficients. There is 24 bit coefficient RAM, 16 bit data RAM and output may be 32 or 16 bit.

These chips have several operational modes: single or cascadable (A)FIR. When it operates in AFIR mode, then the LMS algorithm is used. It is also possible to enable a "DC-tap" option (AC-coupling) and to enable "leakage" to prevent coefficient drift when no input is available. More about AC-coupling can be found in section 8.6. For the communication with its host (DSP 56001), 16 registers are available [23], [24] and [25]. Via the configuration register of the CAFIR, the mode can be adjusted by software, but some hardware settings must also be changed.

In figure 8.3 these modes can be seen.
Onwards to phantom source creation by means of adaptive filtering

A different layout can be drawn for the modes, as shown in figures 8.4 and 8.5. The CAFIR output signal, the negative error signal, is fed back into the LMS algorithm. This is shown by the dotted line. Since this output signal is the negative residue signal, a multiplication with "-1" is necessary.

8.4 Maximum number of taps available

There is a rule for the maximum number of taps that each CAFIR chip may use [24]

$\text{max. \#taps} \leq \frac{j_{ck}^{\text{CAFIR}}}{f_s}$

In which $j_{ck}^{\text{CAFIR}} = 10 \text{ MHz}$ and $f_s = 441 \text{ kHz}$ for audio purposes. To test if a certain tap number is possible, the previous formula has to be checked, with the following equations

$\text{max. \#taps} = \begin{cases} 12 + I + q & \text{FIR mode} \\ 17 + 2I + r & \text{AFIR mode} \end{cases}$
Onwards to phantom source creation by means of adaptive filtering 47

\[ q = \begin{cases} 
30 + n - I : & (29 - n + I) > 0 \\
0 : & \text{otherwise}
\end{cases} \]

\[ r = \begin{cases} 
30 + n - I : & (29 - n + I) > 0 \\
0 : & \text{otherwise}
\end{cases} \]

\( n \) = number of cascaded chips, \( 0 < n < 8 \)

\( I \) = number of taps used on each chip, \( 0 < I < 256 \)

Secure values for \( I \), when \( f_s = 44.1 \ kHz \), are:

- 100 taps in AFIR mode
- 200 taps in FIR mode
8.5 Cascade interface

As previously mentioned, the CAFIR chips can be placed in cascade, but for this the configuration on the XFIR card must be set in a special way [4] and the cascade interface of the chip will be used. This interface is depicted in figure 8.6.

![Cascade interface of DSP 56200](image)

The signals from the cascade interface are the following:

- SDO: serial data output, pass last data sample to next chip
- SDI: serial data input, connected with SDO of previous chip
- SSO: serial sum output, pass serial sum to next chip
- SSI: serial sum input, connected with SSO of previous chip
- SEI: serial error input, pass error term output of last in cascade to all previous chips (only in AFIR mode)

The XFIR card has three jumpers per CAFIR, to ensure easy switching between the possible modes (single, cascaded, (A)FIR). For detailed explanation see [24], [23] and [25].

The jumper settings and the resulting CAFIR modes are shown in table 8.1.

<table>
<thead>
<tr>
<th>Jumper</th>
<th>Set</th>
<th>Set</th>
<th>Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fix0</td>
<td>y</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>Fix1</td>
<td>n</td>
<td>y</td>
<td>n</td>
</tr>
<tr>
<td>FixP</td>
<td>n</td>
<td>n</td>
<td>y</td>
</tr>
<tr>
<td>Mode</td>
<td>FIR</td>
<td>AFIR</td>
<td>AFIR</td>
</tr>
<tr>
<td>Single</td>
<td>single</td>
<td>single</td>
<td>cascaded, not last</td>
</tr>
</tbody>
</table>
Onwards to phantom source creation by means of adaptive filtering

The CAFIRs are placed in such a way, that it is possible to cascade from two up to eight chips. The order from first to last is:
0 → 4 → 1 → 5 → 2 → 6 → 3 → 7 (see figure 8.2, the XFIR layout and the schematics in [4]).

8.6 DC-tap

Input signals to an adaptive filter may have a DC-bias due to A/D converter differential offset. These DC-components can cause problems in an unprotected adaptive filter. This is mathematically shown in the appendix, page 108, for an adaptive filter with one coefficient. The optimum Wiener solution may not be reached and the convergence region may decrease. The adaptive filter cannot synthesize in some situations, therefore a DC-tap is added, which is forced to the DC-bias value. The DC-tap option can be chosen when configuring the CAFIR.

However, if a DSP 56001 or 56002 is used as an adaptive filter, via an implemented program, then AC-coupling is a good solution. An AC-coupling filter is a virtual allpass filter and only very low frequencies are cut off. Since audio-frequencies start at 20 Hz, the AC-coupling does not alter frequency characteristics of the signals.

Experiments showed, that offset can alter the adaptation process when using a DSP 56001 as an adaptive filter. However, this offset-effect can effectively be eliminated by applying AC-coupling to the input signals.

When using digital noise as an input signal (in simulations), no offset is present in the system.
Chapter 9

Digital signal processing using the XMOT-card

9.1 Introduction

A second type of signal processing card is used, named XMOT. This card is shown in figure 9.1. It can be placed in the same digital signal processing system as shown in figure 8.1. The heart of it is formed by two Motorola DSP 56002 processors. For communication between the two DSP 56002 processors, there is 1 kbyte of dual ported RAM memory available. The host (PC) can communicate with the two DSP 56002 processors via the host interface and an external controller. There are 2 kbytes of dual ported RAM memory for the two DSP 56002 processors for a maximum of 16 times LDA-ASP bus access during one sample period. \( f_{\text{sample}} = 44,100 \text{ Hz.} \)

Furthermore, each DSP 56002 processor has 2 x 32 kbyte external memory available.

If the count mode [12] is switched on, the LDA bus interface of the XMOT is equal to that of the XFIR card.

Hardware reset is done by pressing the button on the front side of the board. Each DSP can also be manually reset.

Port C is used as a general purpose I/O port, so information about processing time is available and a trigger pulse is generated. The best option is to use it unbuffered, because practice shows that a buffered version of the processing time is corrupted by noise.
The processor DSP 56002 is slightly different from the DSP 56001: it has a larger instruction set and the main advantage is, that it runs at a clock frequency of at least 40 MHz. This clock frequency can be adjusted by software, up to a maximum of 90 MHz, but each application running on the DSP has its own maximum clock frequency, which has to be determined by empirically, for the hardware is not guaranteed to function correctly at clock frequencies above 40 MHz. (So build a program and increase the speed of the DSP until the program crashes.)

9.2 Special application: filtered-x

The filtered-x algorithm needs two adaptive FIR filters and one fixed FIR filter, as can be seen in figure 3.6. AFIR1 emulates the plant and its coefficients are copied into the fixed filter. This can easily be done by using CAFIR chips: only the coefficients of the adaptive filter are needed, not its output signal. However, AFIR2 is a special kind of adaptive filter, since the input signals to its update algorithm are different from the normal situation: one input is formed by the filtered AFIR input signal, and the other one is formed by the output signal of the plant (so not the direct difference between the primary and reference input of the filter). Furthermore, its output signal is fed to the plant.
Onwards to phantom source creation by means of adaptive filtering

If now the layout of the CAFIR chip is examined, see figure 8.3, then it can easily be seen that it is not capable of performing the desired task:

- input to filter part and update algorithm are always the same signals (connections internally), \( x(n) \)
- the direct difference of primary and reference input, \( e(n) \), is in the feedback loop (internal process)
- the output signal of the adaptive filter, \( y(n) \), is needed, and just the residue signal, \( e(n) \), is available

These factors limit the use of a CAFIR chip. It is clear that when using the filtered-x algorithm, for AFIR2 a CAFIR chip cannot be used. A good solution is using a DSP with a software implemented adaptive filter program.

Assume a sampling frequency \( f_s = 44,100 \) Hz.

A DSP 56001 can handle up to a maximum of 128 taps when using it as an FIR filter and up to 70 taps when using it as an adaptive FIR filter. A DSP 56002 can handle at least 400 taps when using it as an FIR filter and about 140 taps when using it as an adaptive FIR filter.

(These numbers may change, depending on the processor's clock frequency.) When deconvolution of a loudspeaker → ear (or microphone) transfer function for audio purposes is the desired application, then filter lengths of at least 200 taps are necessary [19], so an XMOT card must be used.
Chapter 10

Simulations using the standard LMS algorithm

Before the adaptive filter will be used in practice, it is important to gain more knowledge about the behaviour. That is why simulations are carried out.

By using several DSP cards a complete system is built, which is used to emulate and invert an “unknown” system. By doing this, it is possible to predict how an adaptive filter should behave in practice (so there is less chance of doing wrong measurements).

10.1 Simulation method

As stated, first simulations will cover emulation and inversion of a plant. Therefore, a loudspeaker → ear transfer function is implemented in an FIR filter. (loudspeaker specifications are given in the range of 45–20,000 Hz.

For the FIR filter up to 400 taps will be used, for the AFIR filter up to 300 taps will be used.

10.1.1 Generation of the input signals

The inputs to the signal processing system \((x(n), v_L(n)\) and \(v_R(n))\), are white noise signals.

White noise can be generated in several ways:

- internal Gaussian white noise source from HP 3562A signal analyzer
- Brüel & Kjær Gaussian white noise source
- digital pseudo-random noise routine from [20]

The last mentioned noise source generates digital noise from within a DSP. The other two sources give an analog signal which is fed to the processing system via the A/D converter. This may lead to problems when using
adaptive filters, for A/D and D/A converters add an offset to their input signal. Sometimes, this leads to non-cancellable components which disturb the adaptation process. (see appendix, page 108).
A solution is to let all output signals from the A/D converters pass an AC-coupling filter (high-pass filter with extreme low passband).

Uncorrelated white noise \((u(n)\) and \(s(n)\)) can be made by delaying the signal \(x(n)\) over 8,192 samples (by putting \(x(n)\) into a queue). The mentioned delay assures virtual uncorrelated signals.

### 10.1.2 A simulation run

A simulation run is as follows:

- initialization from the host (PC)
  - down-loading a program
  - transmitting filter coefficients and parameters
- start initialization interrupt routines in DSP
  - initialization of pointers and counters
  - data transmitting to hardware filter chips
  - initialization of memory locations
- allow 44.1 kHz. interrupt
  the DSP is in wait state ("no operation" loop), when a 44.1 kHz. interrupt arrives, the real-time signal processing takes place (the interrupt routine), when this routine ends, the DSP returns into the wait state
- after convergence of the adaptation process, data may be saved onto disk and a new simulation can be started
Onwards to phantom source creation by means of adaptive filtering

10.2 Real-time processing

The simulation system uses four (or five) DSPs, meaning one (or two) XFIR cards and one XMOT card. The names DSP 00 and DSP 01 are used for the DSPs of the XMOT card. The DSPs are named DSP 10, DSP 11, DSP 20 and DSP 21 for the XFIR cards.

The following tasks have to be done in real-time:

- DSP 00 generates white noise
- DSP 01 forms the "unknown" system
- DSP 10 performs the adaptive processing
- DSP 11 calculates mean values
  - mean-squared values for signals
  - mean values for coefficients
- DSP 20 forms the "unknown" system

The DSP 20 is only used when performing deconvolution, its input will be the output of the inverse filter. The output signal should be an estimation to the original white noise input signal to the whole system. This is a way of checking the quality of the deconvolution process.

Mean value calculations are performed by averaging over a block of data (212 values for coefficients and 216 values for signals). Steady-state situation and stationary signals are assumed, for calculating ensemble mean values takes an enormous amount of experiments. By approximation,

$$E\{e(n)\} \approx \overline{e}(n) = \frac{e_1}{N} + \cdots + \frac{e_N}{N}$$

holds ($e_i$ a measurement of signal $e(n)$ and averaging over $N$ values). This procedure is done for all desired mean value calculations.

10.3 Emulating a loudspeaker → microphone transfer function

For the "unknown" system, $H$, a measured transfer function is taken. This is in fact the path from a greycube loudspeaker to a measurement microphone, see page 88 for more information. Its length is shortened to 400 taps, so $J = 400$. 
10.3.1 Measured system parameters

For the simulations, the digital noise source is used, the following parameters were measured

\[ \sigma_x^2 = 0.083 \]
\[ \kappa_x = 1.8 \]

10.3.2 Adaptation constant \( \alpha \)

As previously explained, the noise cancellation set-up will be used to examine the influence of \( \alpha \) on \( J \).

For the filter length of both the FIR and AFIR filters a value of 200 taps is chosen \((J = I = 200)\). This gives

\[ \eta = I - 1 + \kappa_x = 200.8 \]

which means for the convergence region

\[ 0 < \alpha < \frac{1}{\eta \sigma_x^2} = 6 \cdot 10^{-2} \]

The adaptation constant \( \alpha \) determines whether or not the adaptation process converges, the speed at which this happens and the (steady state) accuracy. A tiny value for \( \alpha \) gives –theoretically– the best approximation, but also has two disadvantages:

• it takes a long time to converge

• due to finite word length, the update algorithm will stop even when the optimum solution is not yet reached (the update term falls out of reach)

So an \( \alpha \) has to be chosen in such a way that the accuracy is acceptable.

The theoretical value for \( J_{\text{theory}} \) \((J_{\text{theory}} = I \alpha \sigma_x^2)\) can be compared with those of the simulations.

The results of the variation of \( \alpha \) are shown in table 10.1. Though for each \( \alpha \) the theoretical and measured values for \( J \) differ, it is clear that the overall behaviour \((10 \times \) larger \( \alpha \) gives a 10 dB larger \( J \)) is good.
Onwards to phantom source creation by means of adaptive filtering

Table 10.1: Results of $\alpha$-variation

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$\overline{J}$ [dB]</th>
<th>$J_{\text{theory}}$ [dB]</th>
<th>convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2 \cdot 10^{-2}$</td>
<td>-4.7</td>
<td>-4.8</td>
<td>yes</td>
</tr>
<tr>
<td>$2.5 \cdot 10^{-3}$</td>
<td>-13.3</td>
<td>-13.8</td>
<td>yes</td>
</tr>
<tr>
<td>$2.5 \cdot 10^{-4}$</td>
<td>-24.2</td>
<td>-23.8</td>
<td>yes</td>
</tr>
<tr>
<td>$2.5 \cdot 10^{-5}$</td>
<td>-32.7</td>
<td>-33.8</td>
<td>yes</td>
</tr>
</tbody>
</table>

Deviations in the obtained value for $\overline{J}$ are caused by:

- mean value calculation is performed on a finite data block
- mean value calculation gives bit loss, see the formula on page 55
- a rounded value for $\alpha$ is send to the CAFIR chips (only multiples of $1/32,767$ are possible)

An aspect which can be viewed, is the variance of the coefficients: with increasing $\alpha$, the coefficients start to fluctuate more and more.

Also, the convergence region for a given adaptive filter length can be measured. No additional noise is used, so the schematic is that of figure 7.1.

This dependence of the convergence region on the adaptive filter length, $I$, can be seen in table 10.2. In this table the border of the convergence region, $\alpha_{\text{critical}}^{\text{theory}}$, is calculated by using the relationship

$$\alpha_{\text{critical}}^{\text{theory}} = \frac{1}{\eta \sigma_x^2}$$

With increasing adaptive filter length, the boundary of the convergence region deviates more from its theoretical value, but the table shows that there is a well-defined convergence region for each $I$, which shows a clear dependence on the adaptive filter length: $\alpha_{\text{critical}} \propto I^{-1}$. The deviations can be explained as follows:

- $\alpha$ is not very small, so $w(n)$ is not independent from $x(n)$
- finite word length has influence on the stability of the algorithm
Onwards to phantom source creation by means of adaptive filtering

10.3.3 Adaptive filter length $I$

It is, of course, of great interest to obtain a good result with as few taps as possible, bearing in mind financial and practical considerations. Now $H$ is formed with $J = 400$ taps, $\alpha = 2.5 \cdot 10^{-5}$.

The length of the adaptive filter is changed and $J'$ is determined. The obtained value can be compared with a theoretical one, namely

$$J'_{\text{theory}} = \sigma_x^2 \sum_{i=1}^{J} h_i^2$$

The results are shown in the next table 10.3

Table 10.3: Results of $I$-variation

<table>
<thead>
<tr>
<th>$I$</th>
<th>$J'$</th>
<th>$J'_{\text{theory}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>$1.4 \cdot 10^{-4}$</td>
<td>$0.9 \cdot 10^{-5}$</td>
</tr>
<tr>
<td>200</td>
<td>$2.2 \cdot 10^{-4}$</td>
<td>$2.8 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>150</td>
<td>$4.9 \cdot 10^{-4}$</td>
<td>$5.3 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>100</td>
<td>$1.4 \cdot 10^{-3}$</td>
<td>$1.2 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>50</td>
<td>$3.5 \cdot 10^{-3}$</td>
<td>$3.3 \cdot 10^{-3}$</td>
</tr>
</tbody>
</table>

An explanation for the deviations between theoretical and measured values of $J'$ can be found on page 57.

The influence of a smaller tap length on the magnitude characteristic is shown in figures 10.2, 10.3 and 10.4.
Onwards to phantom source creation by means of adaptive filtering

Figure 10.2: Magnitude characteristic depending on $I$

Figure 10.3: Magnitude characteristic depending on $I$
Onwards to phantom source creation by means of adaptive filtering

When \( I = 50 \), the characteristic is only roughly emulated. With increasing filter length, the emulation becomes better. Still, even for \( I = 200 \), there is a clear difference between the desired characteristic with 400 taps and the obtained one (especially for the lower frequency region). This means, for good emulation at least 200 taps have to be used for the adaptive filter. As can be seen, 300 taps gives the best result.

In [19] an \( I \)-value of 256 is taken, but when considering the hardware possibilities, 300 taps can be obtained with the same amount of CAFIR chips. (100 taps per CAFIR chip is the maximum in adaptive mode, so 256 or 300 taps can both be obtained by using three CAFIR chips)

Also for forming an inverse, as many taps as possible are desired, up to a maximum within limits of costs and implementation. Therefore, an adaptive filter length of 300 taps is taken.
10.3.4 Additive noise at the primary input of the adaptive filter

The setup used for this simulation is that of a noise canceller, but now, the signal $u(n)$ functions as an unwanted signal component (see figure 7.1). The signal $u(n)$ is formed by delaying the input signal $x(n)$, and multiplying it with a constant before adding it to the output signal of $H$. By doing so, a signal to noise ($X/U$) ratio is calculated.

$$u(n) = C \cdot x(n - \tau)$$

with $\tau = 8,196$ samples and $0 < C < 1$. Thus

$$\sigma_u^2 = C^2 \cdot \sigma_x^2$$

holds. Applying this on the equation for $\overline{J}$ gives

$$\overline{J} = \frac{\sigma_u^2}{\sigma_x^2} = C^2$$

The influence of $u(n)$ on the magnitude characteristic is shown in figure 10.5.

![Figure 10.5: Magnitude characteristic depending on $u(n)$](image)
Onwards to phantom source creation by means of adaptive filtering

From figure 10.5 can be concluded, that for different values of $\sigma_n^2$ hardly noticeable deviation in the frequency characteristic is created when comparing these characteristics with each other.

The results of varying $C$ can be found in table 10.4. $J = I = 200$ and $\alpha = 2.5 \cdot 10^{-5}$.

Table 10.4: Results of $C$-variation

<table>
<thead>
<tr>
<th>$X/U$ [dB]</th>
<th>$\bar{J}$</th>
<th>$\bar{J}_{\text{theory}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>98 \cdot 10^{-2}</td>
<td>1 \cdot 10^{0}</td>
</tr>
<tr>
<td>-6</td>
<td>25 \cdot 10^{-2}</td>
<td>25 \cdot 10^{-2}</td>
</tr>
<tr>
<td>-12</td>
<td>6.31 \cdot 10^{-2}</td>
<td>6.25 \cdot 10^{-2}</td>
</tr>
<tr>
<td>-18</td>
<td>1.63 \cdot 10^{-2}</td>
<td>1.57 \cdot 10^{-2}</td>
</tr>
<tr>
<td>-24</td>
<td>4.16 \cdot 10^{-3}</td>
<td>3.91 \cdot 10^{-3}</td>
</tr>
</tbody>
</table>

Table 10.4 shows that the simulations are in accordance with the theory.

An explanation for the deviations between theoretical and measured values of $\bar{J}$ can be found on page 57.

10.3.5 Additive noise at the reference input of the adaptive filter

The influence on the final solution was

$$ E\{w(n)\} = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_s^2} \cdot h $$

as derived earlier. The schematic of figure 7.1 is used.

By using as disturbing component $s(n) = C \cdot x(n - \tau)$ (with $\tau = 8,192$ samples), this final solution yields

$$ E\{w(n)\} = \frac{1}{1 + C^2} \cdot h $$

Depending on the signal to noise ratio, $(X/S)$, the error made can be found to be a bias in the frequency domain. This bias will be defined as

$$ \Delta A = 20 \cdot \log \frac{1}{1 + C^2} $$
Onwards to phantom source creation by means of adaptive filtering

Figure 10.6: Magnitude characteristic depending on $s(n)$

Figure 10.7: Magnitude characteristic depending on $s(n)$
Onwards to phantom source creation by means of adaptive filtering

Again, \( J = I = 200 \) and \( \alpha = 2.5 \cdot 10^{-5} \).
The results of this simulation are shown in table 10.5.
The measured values for \( \Delta A \) are taken from figures 10.6 and 10.7.

\[
\begin{array}{|c|c|c|}
\hline
X/S [dB] & \Delta A [dB] & \Delta A_{\text{theory}} [dB] \\
\hline
0 & -6.0 & -6.0 \\
-6 & -2.0 & -1.94 \\
-12 & -0.5 & -0.52 \\
-18 & -0.13 & -0.134 \\
-24 & -0.03 & -0.034 \\
\hline
\end{array}
\]

The values for \( \Delta A \) are taken from figures 10.6 and 10.7. The results from the simulations are in accordance with the theory. An explanation for the deviations between theoretical and measured values of \( \Delta A \) can be found on page 57.

10.4 Inverse filtering

When equalizing an "unknown" system with transfer function \( H \), the convolution of the impulse response of the inverse function (formed by the adaptive filter \( W \)) with the impulse response of the "unknown" system should give a \( \delta \)-function, with a certain delay \( \Delta \).

In the frequency domain \( W \cdot H = 1 \) should hold.

The magnitude of the "unknown" systems transfer function plays an important role in the equalizing process and the attainable steady-state accuracy. If the development of \( H \) with frequency is erratic (many strong and weak components), then it is very hard to find \( H^{-1} \) due to hardware limitations. Each filter coefficient has technical bounds: \(-1 \leq w_i \leq +1\).

If coefficients need to be larger than \(|1|\), they will grow to their technical maximum value and keep it throughout the adaptation process. Therefore, it is necessary to scale the input signals of the adaptive filter in such a way that even for the largest filter coefficient \( w_{\text{max}} \), \(-1 \leq w_{\text{max}} \leq +1 \) holds.

By using a few examples, the problematic nature of scaling will be illustrated.
Onwards to phantom source creation by means of adaptive filtering

- $h = a$ (one coefficient)
  - $|a| \leq 1 \quad \rightarrow \quad w = h^{-1} = 1/|a| \geq 1$
  - scaling is necessary in such a way that $w \leq 1$
  - $|a| \geq 1 \quad \rightarrow \quad w = h^{-1} = 1/|a| \leq 1$
  - scaling is not necessary since $w$ is in the correct region

- $H$ is minimum-phase
  - form $\hat{H}$, by emulating $H$
  - the inverse can be found by direct inversion of $\hat{H}$ because of the minimum-phase (so the inverse is stable)
  - by performing an inverse DFT, $\hat{h}^{-1}$ can be found
  - the scale factor is obtained by observing the largest coefficient of $\hat{h}^{-1}, h_{\max}$, and setting $|h_{\max}| = 1$

- $H$ is non-minimum-phase
  - direct inversion will lead to an unstable system
  - the schematic for forming the inverse function is available, but the appropriate scale factor can only be determined empirically

Finding a good scale factor is but one problem which arises, for there are two other very important problems when equalizing a plant:

- noise at the reference input of the AFIR
- finiteness, since the optimum inverse is IIR where only FIR is available

10.5 Equalizing a loudspeaker → microphone transfer function

Again, a measured transfer function from a greycube loudspeaker to a measurement microphone is taken as the "unknown" system, $H$, which is modeled as an FIR filter, with filter length $J = 400$ taps.

10.5.1 Adaptation constant $\alpha$

First, a suitable value for the adaptation constant $\alpha$ will be obtained. The main demands are:

- acceptable accuracy
- acceptable rate of convergence

The second demand is of great importance here, to keep the measurement and simulation time within reasonable limit. When the magnitude characteristic of $H$ is weak for a certain region, then the input for that frequency
band is weak and as a consequence will be the update term $\alpha x(n)$. Therefore the coefficient will not change a lot. (Remember that due to finite word length the update term may be rounded to zero.) This process increases as $e(n)$ decreases, so with increasing convergence. For very small values of $\alpha$, this process goes even faster, so the obtained accuracy need not be better than for larger values of $\alpha$. So it is very well possible that the adaptation process will not stop at the optimum solution, which means, that now the choice for $\alpha$ is more critical when compared to emulation.

The adaptive filter length is taken 200 and 300 taps. For the delay, $\Delta$, $\Delta = \text{AFIR length}/2$ is taken, which is a sub-optimum solution according to [20]. Results are shown in tables 10.6 and 10.7.

### Table 10.6: Results of $\alpha$-variation, $I = 200$

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$\overline{J}$</th>
<th>convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2.5 \cdot 10^{-1}$</td>
<td>$\ldots$</td>
<td>no</td>
</tr>
<tr>
<td>$2 \cdot 10^{-1}$</td>
<td>$3.0 \cdot 10^{-3}$</td>
<td>yes</td>
</tr>
<tr>
<td>$1 \cdot 10^{-1}$</td>
<td>$4.3 \cdot 10^{-4}$</td>
<td>yes</td>
</tr>
<tr>
<td>$2.5 \cdot 10^{-2}$</td>
<td>$3.4 \cdot 10^{-4}$</td>
<td>yes</td>
</tr>
<tr>
<td>$2.5 \cdot 10^{-3}$</td>
<td>$3.6 \cdot 10^{-4}$</td>
<td>yes</td>
</tr>
</tbody>
</table>

### Table 10.7: Results of $\alpha$-variation, $I = 300$

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$\overline{J}$</th>
<th>convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2 \cdot 10^{-1}$</td>
<td>$\ldots$</td>
<td>no</td>
</tr>
<tr>
<td>$1.5 \cdot 10^{-1}$</td>
<td>$8.3 \cdot 10^{-4}$</td>
<td>yes</td>
</tr>
<tr>
<td>$2.5 \cdot 10^{-2}$</td>
<td>$3.1 \cdot 10^{-4}$</td>
<td>yes</td>
</tr>
<tr>
<td>$2.5 \cdot 10^{-3}$</td>
<td>$2.9 \cdot 10^{-4}$</td>
<td>yes</td>
</tr>
<tr>
<td>$2.5 \cdot 10^{-4}$</td>
<td>$7.5 \cdot 10^{-4}$</td>
<td>yes</td>
</tr>
</tbody>
</table>

For $I = 200$, a good value for $\alpha$ is $1.375 \cdot 10^{-3}$, whereas for $I = 300$, a good value is $1.25 \cdot 10^{-2}$. (with $\overline{J} = 3.2 \cdot 10^{-4}$ and $\overline{J} = 2.9 \cdot 10^{-4}$ respectively)
For increasing values of $\alpha$, fluctuation in the coefficients is observed. For very small values of $\alpha$, the rate of convergence is very slow, leading to far larger adaptation times, when compared to emulation (up to several minutes).

The convergence region decreases with increasing adaptive filter length, just the same as was the case when emulating an "unknown" system.

As can be seen, the steady-state error $\bar{J}$ will never reach zero, for the applied adaptive filter has finite filter length. For very small $\alpha$, the error starts to increase again, due to finite word length.

In order to evaluate the quality of the formed inverse filter, $H \cdot W$ will be observed. The result of the analyzer's plant emulation measurement, which was shortened to 400 taps, and named $H_{HP\text{3562A}}$, is taken as an FIR representation for $H$. (So, actually $H_{HP\text{3562A}} \cdot W$ is evaluated.) Ideally, this function should be a horizontal line throughout all frequencies (equal magnitude). Due to scaling $H_{HP\text{3562A}} \cdot W \ll 0 \, dB$.

For two values of $\alpha$, the magnitude characteristics are drawn in figure 10.8.

Diagrams of $H_{HP\text{3562A}} \cdot W$ are always on one and the same page to make easy comparison possible.
Onwards to phantom source creation by means of adaptive filtering

As can be seen, for smaller $\alpha$, deviation in the higher frequency region (about 9.5 kHz.) occurs. This can happen because of finite word length effects: the update term becomes zero.

Figure 10.8: $H_{HP3562A} \cdot W = f(\alpha)$
10.5.2 Modeling delay $\Delta$

Since the impulse response of the inverse filter is non-causal, it will be two-sided in time (for the plant is a non-minimum-phase system). The most important part of the inverse impulse response must be present in the adaptive weight vector, which makes the choice for $\Delta$ important.

The usage of $\Delta$, see figure 6.2, enables the adaptive filter to form a delayed, truncated, two-sided impulse response. For each function which side will last the longest and therefore contain most energy (left-handed or right-handed), is different.

By taking $I = 300$ and $\alpha = 1.25 \cdot 10^{-2}$ the dependence of $\bar{J}$ as a function of $\Delta$ is determined.

(The usage of $\Delta$ can be explained as follows: the adaptive filter functions as a rectangular window of length $I$, which thus takes only a part of length $I$ of the two-sided inverse filter's impulse response into account. By changing $\Delta$, the position of the window can be varied.)

Results are shown in table 10.8. Inverse filter frequency responses are shown in figures 10.9 and 10.10. The quality of those inverse functions can be judged by observing figures 10.11 and 10.12.

($\Delta = 150$ can be seen as a reference, when comparing the several pictures.)

<table>
<thead>
<tr>
<th>$\Delta$</th>
<th>$\bar{J}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>$4.3 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>100</td>
<td>$3.1 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>150</td>
<td>$2.9 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>200</td>
<td>$3.5 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>280</td>
<td>$9.3 \cdot 10^{-4}$</td>
</tr>
</tbody>
</table>

The table shows that it is more important to take the right-hand part into account, since $\bar{J}_{\Delta=20} < \bar{J}_{\Delta=280}$.

For a region $100 \leq \Delta < 200$, $\bar{J}$ stays almost constant. $\Delta$ may be optimized, but a good choice is to pick it as half of the adaptive filter length according to [20], as can be seen in figures 10.11 and 10.12.

Although the differences between the values for $\bar{J}$ in table 10.8 are very close to each other, the effect in the frequency domain is very clear.
Onwards to phantom source creation by means of adaptive filtering

Figure 10.9: Magnitude characteristic depending on $\Delta$

Figure 10.10: Magnitude characteristic depending on $\Delta$
Onwards to phantom source creation by means of adaptive filtering

Figure 10.11: $H_{HP3562A} \cdot W = f(\Delta)$
Onwards to phantom source creation by means of adaptive filtering

Figure 10.12: $H_{HP3562A} \cdot W = f(\Delta)$
10.5.3 Adaptive filter length $I$

In order to invert a loudspeaker $\rightarrow$ ear transfer function, principally an infinite number of taps is required for the adaptive filter. The more coefficients used, the better the approximation will be. Simulations have been carried out, to examine the behaviour of the inverse filter when its length is changed.

Again, $J = 400$ taps and $\Delta = \text{AFIR length}/2$. The value for $\alpha$ is optimized for each adaptive filter length. Results can be seen in table 10.9. Figure 10.13 shows the inverse filter frequency response. The quality of the inverse filter can be judged by observing figure 10.14.

<table>
<thead>
<tr>
<th>$I$</th>
<th>$\alpha$</th>
<th>$\overline{J}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>$1.25 \times 10^{-2}$</td>
<td>$2.9 \times 10^{-4}$</td>
</tr>
<tr>
<td>200</td>
<td>$13.75 \times 10^{-3}$</td>
<td>$3.4 \times 10^{-4}$</td>
</tr>
<tr>
<td>100</td>
<td>$13.75 \times 10^{-4}$</td>
<td>$5.9 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

Figure 10.13: Magnitude characteristic depending on $I$
Onwards to phantom source creation by means of adaptive filtering

Figure 10.14: $H_{HP3562A} \cdot W = f(I)$
Onwards to phantom source creation by means of adaptive filtering

From the previous figure, it is clear that with increasing adaptive filter length, the quality of the inverse filter increases (starting for the higher frequencies). A length of 300 taps is enough.

10.5.4 Additive noise at the reference input of the adaptive filter

The least-squares solution is very sensitive to additive noise at the reference adaptive filter input and could render ineffective as a plant inverse. The spectrum of the additional noise is non-white, coloured by $H$. This means that the signal to noise ratio per frequency band will differ, and so will the quality of the final solution of the deconvolution process. (All this can be concluded from Section 10.4.)

The additive noise signal $u(n)$ is formed by $u(n) = C \cdot x(n-\tau)$, with $\tau = 8,192$ samples and $0 < C < 1$.

For different signal to noise ratios, $X/U$, $\overline{J'}$ is determined in table 10.10 and as a quality check, $H_{HP3562A} \cdot W$ as a function of $C$, so of this ratio, is shown in figure 10.15. (Remember, $X/U = 10 \cdot \log C^2$)

<table>
<thead>
<tr>
<th>$X/U$ [dB]</th>
<th>$\overline{J'}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-12</td>
<td>$7.7 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>-18</td>
<td>$4.1 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>-24</td>
<td>$2.2 \cdot 10^{-3}$</td>
</tr>
</tbody>
</table>

As can be seen, the steady-state error $\overline{J'}$ decreases with decreasing additional noise power.
Figure 10.15: $H_{HP3562A} \cdot W = f(C)$
The result of the deconvolution process is very sensitive to noise at the primary input. First deviations are noticed in the higher frequency region. When a large noise component is present, the solution of the equalization process is no good.

10.6 Conclusions from the simulations

Simulations were carried out to get an idea of how the DSP 56200 adaptive hardware filter behaves when emulating and equalizing an "unknown" plant: the acoustical path from loudspeaker to microphone. Furthermore, points of interest for measurements were found and a starting point for these measurements was created.

From those simulations, the following conclusions can be drawn:

- Emulating an acoustical path from loudspeaker to microphone, is a well-performed task for the adaptive filter, if the "unknown" system's impulse response is not longer than the adaptive filter's length, or if that "unknown" impulse response has a tail which contains only a small amount of energy (which is the case for loudspeaker → ear or microphone plants). The adaptation constant, $\alpha$, can be chosen in a wide range (limited by convergence region).

- Deconvolution gives results which are not as good as emulation, but still good enough, mainly, since the filter length is limited. The adaptation constant must be chosen from a smaller region, and its choice is more critical (too small values are bad). The inclusion of a delay $\Delta$, ensures better performance. Noise at the reference input alters the final solution, for good adaptation, a high signal to noise ratio is desired. Scaling of the input signals is necessary, the sufficient factors must be determined empirically. The rate of convergence is very low, so the adaptation process takes quite a long time.

- Determining the quality of the solution found by the adaptive filter is sometimes a problem due to finite word length (\(J(n)\) consists out of tiny values which are summed).

- Adaptive filters are sensitive to offset in input signals, AC-coupling or DC-tap are good solutions to overcome troubles when adapting (shown in other, more basic simulations, which are not reported here).
Chapter 11

Simulations using the pseudo filtered-x algorithm

11.1 Introduction

In order to gain more insight in the filtered-x algorithm, a modified schematic is used for simulations of plant deconvolution. This schematic is depicted in figure 11.1. It is given the name "pseudo filtered-x".

The difference to the filtered-x schematic of figure 3.6 is obvious: now a direct copy of the plant, $H_{Fr}^p(z)$, is placed in an FIR filter to generate $r(n)$, instead of using the approximated version, $\hat{H}_{Fr}^p(z)$, obtained by emulation. The adaptive filter left, is used for the equalization process.

The sampling frequency is lowered to $f_s = 7.35$ kHz., this ensures enough processing time for the complete algorithm to be implemented on one DSP 56002.
Onwards to phantom source creation by means of adaptive filtering

The plant, $H_{pR}$, is modeled as a 294 tap FIR filter (the coefficients are obtained via measurements with the signal analyzer). This shortened version is called $H_{HP3562A}$.

The plant noise, $u(n)$, is simulated by an additional, uncorrelated, white noise source. It is only used for determining the influence of plant noise on the final solution of the adaptation process.

Now the influence of the most important parameters of the adaptive filter can be examined.

11.2 Real-time processing

The simulation system uses three DSPs, meaning one XMOT and one XFIR card. The DSPs on the XMOT card are called DSP 00 and DSP 01. The DSP on the XFIR card is called DSP 11.

The following tasks have to be done in real-time:

- DSP 00 generates white noise
- DSP 01 forms the pseudo filtered-x system
- DSP 11 calculates mean values
  - mean-squared values for signals
  - mean values for coefficients

11.3 Equalizing a loudspeaker → microphone transfer function

11.3.1 Adaptation constant $\alpha$

Again, a suitable value for $\alpha$ is desired, ensuring good accuracy as well as acceptable rate of convergence. $J = I = 294$ and $\Delta = \text{AFIR length}/2$ are taken. Results are shown in table 11.1.

As can be seen, the steady-state error $\overline{J}$ will never reach zero, for the applied adaptive filter has finite filter length.

For small $\alpha$, this error starts to increase again, for an explanation see page 66. Though, the values do not differ a lot. The best possible solution is to compare the frequency characteristics.

The value for $\alpha$ of 0.025 gives best results within fair adaptation time.
Onwards to phantom source creation by means of adaptive filtering

Table 11.1: Results of \( \alpha \)-variation

<table>
<thead>
<tr>
<th>( \alpha )</th>
<th>( \mathcal{J}' )</th>
<th>convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 7.5 \cdot 10^{-2} )</td>
<td>---</td>
<td>no</td>
</tr>
<tr>
<td>( 5 \cdot 10^{-2} )</td>
<td>( 3.9 \cdot 10^{-3} )</td>
<td>yes</td>
</tr>
<tr>
<td>( 2.5 \cdot 10^{-2} )</td>
<td>( 3.6 \cdot 10^{-3} )</td>
<td>yes</td>
</tr>
<tr>
<td>( 5 \cdot 10^{-3} )</td>
<td>( 3.7 \cdot 10^{-3} )</td>
<td>yes</td>
</tr>
<tr>
<td>( 2.5 \cdot 10^{-3} )</td>
<td>( 4.1 \cdot 10^{-3} )</td>
<td>yes</td>
</tr>
</tbody>
</table>

11.3.2 Modeling delay \( \Delta \)

By taking \( J = I = 294 \) and \( \alpha = 0.025 \), the dependence of \( \mathcal{J}' \) as a function of \( \Delta \) is determined, for the most important part of the inverse impulse response must be present in the adaptive weight vector. Table 11.2 shows the results.

Table 11.2: Results of \( \Delta \)-variation

<table>
<thead>
<tr>
<th>( \Delta )</th>
<th>( \mathcal{J}' )</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>( 4.1 \cdot 10^{-3} )</td>
</tr>
<tr>
<td>100</td>
<td>( 3.8 \cdot 10^{-3} )</td>
</tr>
<tr>
<td>147</td>
<td>( 3.6 \cdot 10^{-3} )</td>
</tr>
<tr>
<td>200</td>
<td>( 3.7 \cdot 10^{-3} )</td>
</tr>
<tr>
<td>270</td>
<td>( 4.3 \cdot 10^{-3} )</td>
</tr>
</tbody>
</table>

For a region \( 100 < \Delta < 200 \), \( \mathcal{J}' \) stays almost constant. Again, \( \Delta \) may be optimized, but a value of \( \Delta = \text{AFIR length}/2 \) is good enough.

Inverse filter frequency responses are shown in figures 11.2 and 11.3. The quality of those inverse filters can be judged by observing figures 11.4 and 11.5. (\( \Delta = 147 \) can be seen as a reference, when comparing the several pictures.)
Onwards to phantom source creation by means of adaptive filtering

Figure 11.2: Magnitude characteristic depending on $\Delta$

Figure 11.3: Magnitude characteristic depending on $\Delta$
Onwards to phantom source creation by means of adaptive filtering

Figure 11.4: \( H_{HP3562A} \cdot W = f(\Delta) \)
Onwards to phantom source creation by means of adaptive filtering

Figure 11.5: $H_{HP3562A} \cdot W = f(\Delta)$
11.3.3 Adaptive filter length $I$

Since the inverse filter of the plant is an IIR filter, an increase in adaptive FIR filter length should give better results.

Simulations have been carried out, to examine the behaviour of the inverse filter when its length is changed.

Again, $J = 294$ and $\Delta = 147$. The value for $\alpha$ is optimized for each adaptive filter length.

The results for varying $I$ are shown in table 11.3. The quality of the inverse filter can be judged by observing figure 11.7.

Table 11.3: Results of $I$-variation

<table>
<thead>
<tr>
<th>$I$</th>
<th>$\alpha$</th>
<th>$J'$</th>
</tr>
</thead>
<tbody>
<tr>
<td>294</td>
<td>$2.5 \cdot 10^{-2}$</td>
<td>$3.6 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>200</td>
<td>$2.5 \cdot 10^{-2}$</td>
<td>$4 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>100</td>
<td>$2.5 \cdot 10^{-2}$</td>
<td>$4.4 \cdot 10^{-3}$</td>
</tr>
</tbody>
</table>

Figure 11.6: Magnitude characteristic depending on $I$
Onwards to phantom source creation by means of adaptive filtering

Figure 11.7: $H_{HP3562A} \cdot W = f(I)$
11.3.4 Additive noise at primary adaptive filter input

Plant noise was the reason for the development of the filtered-x algorithm. By changing the power of the signal \( u(n) \), the influence of \( u(n) \) on the adaptive filter's behaviour can be examined.

\( I \) was taken 294, \( \Delta \) was 147 and \( \alpha \) had a value of 0.025.

The additive noise signal \( u(n) \) is formed by \( u(n) = C \cdot x(n - \tau) \), with \( \tau = 8,192 \) samples and \( 0 < C < 1 \).

For different signal to noise ratios, \( X/U \), \( \overline{J} \) is determined. Results are in table 11.4.

<table>
<thead>
<tr>
<th>( X/U ) [dB]</th>
<th>( \overline{J} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>-6</td>
<td>280 \times 10^{-3}</td>
</tr>
<tr>
<td>-12</td>
<td>73 \times 10^{-3}</td>
</tr>
<tr>
<td>-18</td>
<td>20 \times 10^{-3}</td>
</tr>
<tr>
<td>-24</td>
<td>8 \times 10^{-3}</td>
</tr>
</tbody>
</table>

Each signal \( u(n) \) gives an extra contribution to \( \overline{J} \).

As can be seen, the steady-state error increases with increasing additional noise power, since \( u(n) \) forms an uncorrelated component.

The influence of \( u(n) \) on the optimum solution is depicted in figure 11.8.

The expected value of the adaptive weight vector \( w^* \) is unaffected by the independent plant noise, though practice shows that the variance of the coefficients grows (more fluctuations) with increasing noise power.
Figure 11.8: Magnitude characteristic depending on $C$

11.4 Conclusions from the simulations

Equalizing a loudspeaker $\rightarrow$ microphone transfer function, using the filtered-x algorithm gives good results.
The adaptation constant can be chosen from a fair region, and again, too small values do not give best results.
The inclusion of a delay $\Delta$, ensures better performance, though its influence is less compared to that in the inversion method using standard LMS.
With increasing adaptive filter length, the quality of the formed inverse filter improves.
Scaling of the input signals is necessary, the sufficient factors must be determined empirically.
The most important advantage when compared to standard LMS is, that plant noise has no effect on the average optimum solution.
Chapter 12

Standard LMS experiments, using a microphone and the NEUMANN dummy head

Simulations were carried out in such a way, as to be very close to actual measurements and results could be used directly.

However, the situation is far too ideal when doing simulations compared to doing measurements. Therefore, experiments with a loudspeaker, microphone and a dummy head were carried out to see if the simulated methods are useful in practice.

12.1 Measurements in the anechoic chamber

Experiments were undertaken in a testroom, an anechoic chamber. It contains the greycube loudspeaker, type Philips Pro 410 50 W. dual cone 100 mm., 45–20,000 Hz., the microphone, Brüel & Kjær type 4133 free field 1/2" and a dummy head, NEUMANN type KU811.

The measurement setup is depicted in figure 12.1. A white noise signal is generated by the signal analyzer, an HP 3562 A. It is fed to the loudspeaker via an audio amplifier. A microphone (stand alone or inside the dummy head) picks up the sound from the loudspeaker and, via a microphone amplifier, the signal is passed on to an A/D converter (16 bit resolution), which feeds the signal back into the processing cards. It is also possible to pass an output signal from the DSP rack via a D/A converter (16 bit resolution) to the loudspeaker amplifier.

The microphone is used as a comparison with the dummy head, in order to
Onwards to phantom source creation by means of adaptive filtering

Onwards to phantom source creation by means of adaptive filtering

Figure 12.1: Experimental arrangement

examine the influence of the outer ear, the auditory duct, the head and the torso.

A measurement adaptation run is identical to a simulation adaptation run, except for a minor adjustment in the adaptation programs: the acoustical delay has to be taken into account. The loudspeaker → ear distance is 1 m., so a delay of 129 samples is taken for \( \Delta_{\text{plant}} \).

12.2 Measurement of a loudspeaker → microphone transfer function

12.2.1 Without adaptive filtering

Figure 12.2 shows the setup used for measurements of the transfer functions. The signal analyzer (HP 3562A) generates white noise, which is fed to the
Onwards to phantom source creation by means of adaptive filtering

loudspeaker. The microphone (ear) then picks up a signal, which is fed back to the analyzer. (Microphones may be used in place of the artificial head.) The analyzer can determine the impulse response and transfer function of the acoustical path.

First, a microphone, rather than the artificial head, is used to pick up the output signal. The obtained impulse response is depicted in figure 12.3. The analyzer measures with 2,048 taps and it samples at 51.2 kHz, so in order to compare, the results have to be 1,764 taps, since the clock frequency of the DSP system is 44.1 kHz. (decimating the analyzer’s sampling frequency down to the audio sampling frequency, a decimation factor $R = \frac{441}{512}$).

The analyzer measures the impulse response as follows: An analog signal passes A/D conversion. Then AC-coupling and lowpass filtering ($f \leq f_{\text{chosen}}$) are applied. After windowing, an inverse FFT takes place, since the analyzer measures in the frequency domain (cross spectrum and power spectrum are measured and divided to obtain the frequency response). This procedure is carried out several times to obtain an average characteristic.
12.2.2 With adaptive filtering

Figure 12.4 shows the setup used for the measurements of the transfer functions. The signal analyzer (HP 3562A) generates white noise, which is fed to the loudspeaker. The microphone (ear) then picks up a signal which is fed to
Onwards to phantom source creation by means of adaptive filtering

the adaptive filter. (All input signals for the adaptive filter first pass an A/D converter.) The filter will adapt the acoustical path and when the adaption is completed, the output signal will resemble that of the microphone. (The analyzer can perform a check on the estimated transfer function, \( \hat{P}_r(z) \), from the adaptive filter to compare \( P_r^o(z) \) and \( \hat{P}_r(z) \), as a simple way of error detection.)

The impulse response shown in figure 12.5, is obtained by using a microphone without the artificial head. The response from the adaptive filter has 300 taps.

The time scales of figures 12.3 and 12.5 differ, because the used filter lengths differ: 300 taps corresponds to a maximum time of 6.8 msec. and 1,764 taps corresponds to a maximum time of 40 msec., with an applied sampling frequency of 44.1 kHz. The time scale of figure 12.3 is deliberately shortened, to make its contents clearer (the omitted part contains the tail of the impulse response, values near to zero).

In order to obtain the impulse response from the adaptive filter, each tap is averaged over \( 2^{12} \) values. Those average tap values are then read out directly from DSP memory.

![Impulse response](image)

Figure 12.5: Impulse response loudspeaker \( \rightarrow \) microphone using AFIRs
12.2.3 Results

The obtained transfer functions are depicted in figure 12.6.

![Figure 12.6: Transfer function loudspeaker → microphone](image)

It is very clear that the frequency responses in figure 12.6 differ for the lower frequencies, but this can be explained by the fact that using less taps results in truncation of the impulse response, causing smoothing in the lower frequency region.

For the higher frequencies, the transfer functions are very similar.
12.3 Measurement of an inverse loudspeaker → microphone transfer function

As stated previously, direct inversion of a non-minimum-phase system will lead to an unstable system. This is shown in figure 12.7, in which direct inversion (via a signal processing program) is applied to an emulation measurement, done by the analyzer.

![Figure 12.7: Impulse response inverse loudspeaker → microphone using direct inversion](image)

Figure 12.8 shows the setup used for the measurements of the inverse transfer functions. Primary and reference inputs of the adaptive filters are switched when compared to the emulation measurement setup.

When making an inverse function using adaptive filters, the applied method with a delay leads to good results. The inverse functions formed could equalize the transfer function from loudspeaker → microphone well. The result can be seen in figure 12.9.

The transfer functions obtained are depicted in figures 12.10 and 12.11.

The plant's and adaptive filter's impulse responses can be convolved to get a quality check. Ideally, this convolution should be a horizontal characteristic in the frequency domain, at 0 dB. The actual change of this function with frequency is depicted in figure 12.12.
There is a difference in magnitude between the expected and the obtained convoluted signals, the obtained horizontal characteristic lies beneath 0 dB, due to the fact that scaling is necessary to avoid adaptive coefficients reaching their technical maximum value of $|1|$. 

Figure 12.8: Equalization measurement with adaptive filters
Onwards to phantom source creation by means of adaptive filtering

Figure 12.9: Impulse response inverse loudspeaker → microphone using AFIRs

Figure 12.10: Transfer function inverse loudspeaker → microphone using direct inversion
Figure 12.11: Transfer function inverse loudspeaker → microphone using AFIRs

Figure 12.12: $H_{HP3562A} \cdot W$
12.4 Measurement of a loudspeaker → ear transfer function

Instead of the microphone, the dummy head is used. The measurement is almost the same, only the microphone amplifier has to be adjusted, as the dummy head microphone gives a stronger signal and the acoustical delay has to be altered to 180 samples (the dummy head is placed at a larger distance from the loudspeaker). Transfer functions measured with the analyzer and the adaptive filters are depicted in figure 12.13.

```
Figure 12.13: Transfer function inverse loudspeaker → ear
```

12.5 Measurement of an inverse loudspeaker → ear transfer function

The microphone amplifier has to be adjusted to ensure good adaptation. The impulse response of the adaptive filter, used to form an inverse plant, is depicted in figure 12.14. The corresponding transfer function is depicted in figure 12.15.
Onwards to phantom source creation by means of adaptive filtering

Figure 12.14: Impulse response inverse loudspeaker → ear using AFIRs

Figure 12.15: Transfer function inverse loudspeaker → microphone using AFIRs
Onwards to phantom source creation by means of adaptive filtering

The quality check is performed by the convolution between the impulse responses of the plant and the adaptive filter. The frequency domain plot is shown in figure 12.16.

![Frequency Domain Plot](image)

**Figure 12.16: $H_{WP_{3562A}} \cdot W**

### 12.6 Discussion

#### 12.6.1 A loudspeaker → microphone transfer function

When comparing the impulse responses from figures 12.3 and 12.5, it becomes clear that they are slightly different. One reason is that the adaptive filter length is much lower than that from the analyzer; 300 vs. 1,764 taps. Another reason is that the analyzer uses different A/D converters, lowpass filtering etc. to perform measurements in the frequency domain. Later on, the results are transformed back into time domain, whereas the adaptive filter does its processing in time domain. The difference at the beginning of the impulse responses can be explained by the value for the acoustical delay having been altered.

The emulated transfer function approximates the measured one quite well, as can be seen in figure 12.6.
12.6.2 An inverse loudspeaker → microphone transfer function

From figure 12.12, it can be concluded, that the inverse function formed is quite good. For a few frequencies, the adaptive filter has difficulties, as can be seen (peaks in the plot). Deviations from an ideal horizontal characteristic are noticed for the lower frequency region.

12.6.3 A loudspeaker → ear transfer function

As can be seen in figure 12.13, the adaptive filter can emulate the desired function quite well. Deviation appears only for the lower frequency region.

Differences between a transfer function to ear and to microphone are definitely in the higher frequency region, starting at about 15 kHz. When using an ear, the transfer function decreases dramatically (compare figures 12.6 and 12.13).

12.6.4 An inverse loudspeaker → ear transfer function

As can be seen from figure 12.16, the adaptive filter has more problems when deconvolving the plant's transfer function. Since the plant's transfer function has zeroes nearby the unit circle for frequencies about 9 and 16 kHz (sharp dips), the inverse should cancel these. Therefore, a large filter length is necessary. The quality decreases for low frequencies, again due to filter length shortage.

12.7 Conclusions from the measurements

The following conclusions can be drawn:

- Emulation gives good results
- Deconvolution gives good results, though less good than emulation
- A transfer function to ear is more complex a case than a transfer function to microphone
- Amplification factors have to be adjusted in such a way that no overflow occurs and that the filter coefficients do not reach their technical limit
- Measurements are easily reproduced
Chapter 13

Conclusions

• Emulating a loudspeaker → ear (or microphone) transfer function
  If the length of the impulse response of the plant and the adaptive filter
  are equal, or if the tail of the plant’s impulse response does not contain
  much energy, then a good reproduction can be formed. An adaptive
  filter length of 300 taps is sufficient for emulating a loudspeaker → ear
  or microphone transfer function.
  The adaptation constant $\alpha$ can be chosen within fair limits.
  The signal to noise ratio at the reference input should be large enough,
  otherwise, a bias will occur in the final transfer function obtained.

• Deconvolving a loudspeaker → ear (or microphone) transfer function
  Due to the limited adaptive filter length, the results are less good.
  For good equalizing of a loudspeaker → ear (or microphone) transfer
  function, an adaptive filter length of 300 taps is necessary.
  A good modeling delay in the primary input ensures better approxi­
  mations.
  In comparison with emulation, the adaptation constant must be cho­
  sen from within a smaller region (not too small nor too big).
  The signal to noise ratio at the reference input is crucial for good in­
  version.
  Special attention must be paid to scaling factors for the inverse filter
  coefficients are doomed to reach their technical limit.

• Filtered-x
  Since even on XMOT cards with high clock frequency software adap­
  tive filters with more than 140 taps cannot be implemented, a plant’s
  transfer function cannot be equalized good enough.
  For multi-channel deconvolution, the situation is even worse, since this
  needs a bigger update part.
  So, for audio-purposes, the filtered-x algorithm cannot be used with
  the available adaptive filters.
• Pseudo filtered-x
When a lower sampling frequency is applied, good plant deconvolution is possible. The adaptive filter length can be large (up to 300 taps) with still enough calculation power present. This may be used for multi-channel calculations.
Chapter 14

Future research

Possible solutions for the problems with filtered-x algorithm for audio purposes, may be using IIR adaptive filters or using efficient block frequency domain adaptive filter (BFDAF) techniques.

For further research with filtered-x the following procedure is recommended: Try low sampling frequency, this ensures for larger filter lengths to be implemented and filters may be shorter for good emulation and deconvolution. First, the single-channel case must be examined. From thereon, the multi-channel case can be tackled. Finally, this multi-channel case can be applied to generate phantom sources, for it is the same algorithm as used in active noise reduction problems.

A very different approach, is to examine the use of a CAFIR chip as an FIR filter and to update its coefficients, by using a software update algorithm. This is also a form of an adaptive filter, though only one coefficient can be updated each sample period. (This is not according to general adaptive filter techniques, but perhaps it is a possible way of trying to use an XFIR card to implement a modified form of the filtered-x algorithm.)
Chapter 15

Appendix

Bare in mind: \(x(n), s(n)\) and \(u(n)\) are all uncorrelated, white noise signals, with zero mean.

15.1 Obtaining \(\mathcal{J}'\) with \(I = J\)

\[
\mathcal{J}'(n) = \frac{E\{e^2(n)\}}{\sigma_x^2} \approx E\{d^T(n) \cdot d(n)\}
\]

From [18]

\[
E\{d^T(n + 1) \cdot d(n + 1)\} = (1 - 4\alpha \sigma_x^2 + 4\alpha^2 \eta \sigma_x^4) \cdot E\{d^T(n) \cdot d(n)\}
\]

holds, which leads to

\[
\mathcal{J}'(n + 1) = (1 - 4\alpha \sigma_x^2 + 4\alpha^2 \eta \sigma_x^4) \cdot \mathcal{J}'(n)
\]

Solving for \(n\) gives

\[
\mathcal{J}'(n) = (1 - 4\alpha \sigma_x^2 + 4\alpha^2 \eta \sigma_x^4)^k \cdot \mathcal{J}'(0)
\]

For \(0 < \alpha < \frac{1}{\eta \sigma_x^2}\) this converges to

\[
\lim_{n \to \infty} \mathcal{J}'(n) = \mathcal{J}' = 0
\]

15.2 Adaptive filter with tap shortage, \(I < J\)

The same procedure as in [19] is used. Define

\[
x^T_I(n) = [x(n) \cdots x(n - I + 1)]
\]

and

\[
x_{I-J}^T(n) = [x(n - I) \cdots x(n - J + 1)]
\]
Onwards to phantom source creation by means of adaptive filtering

\[ a(n) = h_J^T \cdot x_J(n) = x_J^T(n) \cdot h_J \]

and

\[ \hat{a}(n) = w^T(n) \cdot x_I(n) = x_I^T(n) \cdot w(n) \]

hold. The optimum Wiener solution is

\[ w^* = R_x^{-1} \cdot R_{xa} \]

with

\[ R_x = E\{x_I(n) \cdot x_I^T(n)\} = \sigma_x^2 \cdot \mathbb{I} \quad (\mathbb{I} \text{ is the } I \times I \text{ identity matrix}) \]

\[ R_{xa} = E\{x_I(n) \cdot a(n)\} = E\{x_I(n) \cdot x_J^T(n) \cdot h_J\} = R_{xIJ} \cdot h_J \]

in which

\[ R_{xIJ} = E\{x_J(n) \cdot x_J^T(n)\} = [R_x | E\{x_I(n) \cdot x_J^T(n-I)\}] \]

which can be simplified to

\[ R_{xIJ} = [R_x | O] \quad (O \text{ is the null matrix}) \]

for a white noise signal \( x(n) \). This all yields for the optimum Wiener solution

\[ w^* = R_x^{-1} \cdot R_{xa} = R_x^{-1} \cdot [R_x | O] \cdot h_J = h_I \]

with

\[ J_{\text{min}} = \sigma_a^2 - R_{xa}^T \cdot R_x^{-1} \cdot R_{xa} \]

This expression can be simplified by using

\[ \sigma_a^2 = E\{(h_0 \cdot x(n) + \cdots + h_{J-1} \cdot x(n-J+1))^2\} = \sigma_x^2(h_0^2 + \cdots + h_{J-1}^2) \]
Onwards to phantom source creation by means of adaptive filtering

Furthermore,

\[ R^{-1}_{xx} \cdot R_{xa} = w^* = h_I \]

and

\[ R_{xa}^T \cdot h_I = h_f^T \cdot R_{xx}^T \cdot h_I = h_f^T \cdot \sigma_x^2 \cdot \left( \frac{I}{O} \right) \cdot h_I = \sigma_x^2 (h_0^2 + \cdots + h_{J-1}^2) \]

All leading finally to

\[ J_{\text{min}} = \sigma_x^2 (h_0^2 + \cdots + h_{J-1}^2) \]

15.3 Additional noise at the reference input of the adaptive filter

Assume \( J = I \), and both \( x(n) \) and \( s(n) \) are white noise signals, then

\[ d(n) = h - w(n) \]

holds, for the difference channel. The LMS update formula is

\[ w(n + 1) = w(n) + 2\alpha \cdot (x(n) + s(n)) \cdot e(n) \]

Which leads to

\[ d(n + 1) = (I - 2\alpha(x^T(n) + s^T(n))(x(n) + s(n)))d(n) + 2\alpha(x^T(n)s(n) + s^T(n)x(n))h \]

for the difference channel, which gives

\[ E\{d(n + 1)\} = (1 - 2\alpha(\sigma_x^2 + \sigma_s^2))E\{d(n)\} + 2\alpha \sigma_s^2 h \]

on average. This leads to

\[ E\{w(n)\} = h - E\{d(n)\} \]
for the coefficients, which can be simplified to
\[ E\{w(n)\} = (1 - 2\alpha(\sigma_x^2 + \sigma_s^2))E\{w(n - 1)\} + 2\alpha\sigma_x^2 h \]
So, for the steady state,
\[ \lim_{n \to \infty} E\{w(n)\} = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_s^2} \cdot h \]
holds, if \(|1 - 2\alpha(\sigma_x^2 + \sigma_s^2)| < 1\) holds, so for the convergence region
\[ 0 < \alpha < \frac{1}{\sigma_x^2 + \sigma_s^2} \]
is obtained. And so the final Wiener solution is
\[ w^*(n) = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_s^2} \cdot h \]

15.4 On the influence of offset upon the AFIR behaviour

When applying an adaptive filter program in a DSP, rather than using the hardware filter chips, special attention must be paid to offset caused by A/D and D/A-converters. This offset can be of great influence on the behaviour of the adaptive filter, as will be shown.

An echo canceller with one coefficient, shown in figure 15.3 is examined (in order to keep the mathematics simple).

The input signal, \(x(n)\), is white noise, and via the echo path, \(\hat{a}(n)\) can be observed. This signal, \(\hat{a}(n)\), can be modelled as the desired, uncorrelated white noise signal \(u(n)\), corrupted with the echo \(a(n)\).
From the LMS algorithm, it follows for the coefficients

\[ w(n+1) = w(n) + 2\alpha x(n)e(n) \]
\[ = w(n) + 2\alpha x(n)(u(n) + a(n) - \hat{a}(n)) \]
\[ = w(n) + 2\alpha x(n)(u(n) + hx(n) - w(n)x(n)) \]
\[ = w(n) + 2\alpha x(n)u(n) + hx^2(n) - w(n)x^2(n)) \]
\[ = (1 - 2\alpha x^2(n))w(n) + 2\alpha x(n)u(n) + 2\alpha hx^2(n) \]

Which gives, on average

\[ E\{w(n+1)\} = (1 - 2\alpha E\{x^2(n)\})E\{w(n)\} + 2\alpha E\{x(n)\}E\{u(n)\} + 2\alpha hE\{x^2(n)\} \]

assumed that \( x(n) \) and \( u(n) \) are statistically independent.

Now suppose the models for the input and desired signals are

\[ x(n) = b + m(n) \quad E\{m(n)\} = 0 \quad E\{m^2(n)\} = \sigma_m^2 \]
\[ u(n) = a + p(n) \quad E\{p(n)\} = 0 \quad E\{p^2(n)\} = \sigma_p^2 \]

then for the average coefficient update

\[ E\{w(n+1)\} = (1 - 2\alpha(b^2 + \sigma_m^2))E\{w(n)\} + 2\alpha ab + 2\alpha h(b^2 + \sigma_m^2) \]
\[ = (1 - 2\alpha(b^2 + \sigma_m^2))E\{w(n)\} + 2\alpha(ab + h(b^2 + \sigma_m^2)) \]

holds, which leads to

\[ E\{w(n)\} = (1 - 2\alpha(b^2 + \sigma_m^2))nE\{w(0)\} + 2\alpha(ab + h(b^2 + \sigma_m^2)) \sum_{j=0}^{n-1} (1 - 2\alpha(b^2 + \sigma_m^2))j \]

For \( n \to \infty \) the mean coefficient value will be

\[ E\{w(\infty)\} = E\{w(0)\}( \lim_{n\to\infty} (1 - 2\alpha(b^2 + \sigma_m^2))^n ) + 2\alpha(ab + h(b^2 + \sigma_m^2)) \sum_{j=0}^{\infty} (1 - 2\alpha(b^2 + \sigma_m^2))^j \]

This equation exists when

\[ 0 < \alpha < \frac{1}{b^2 + \sigma_m^2} \]

holds, the limit has to converge. This interval is called the convergence region.
If $\alpha$ is within the convergence region, the mean endvalue for the adaptive filter will be

\[
E\{w(\infty)\} = 2\alpha(ab + h(b^2 + \sigma_m^2)) \cdot \frac{1}{1 - (1 - 2\alpha(b^2 + \sigma_m^2))} = h + \frac{ab}{b^2 + \sigma_m^2}
\]

The following conclusions can be drawn:

- When there is no offset present in both $x(n)$ and $u(n)$, meaning $a = 0$ and $b = 0$, then the final solution will be

  \[E\{w(\infty)\} = h \quad \text{with convergence region} \quad 0 < \alpha < \frac{1}{\sigma_m^2}\]

- When there is only offset in $u(n)$, meaning $a \neq 0$ and $b = 0$, then the final solution will be

  \[E\{w(\infty)\} = h \quad \text{with convergence region} \quad 0 < \alpha < \frac{1}{\sigma_m^2}\]

- When there is only offset in $x(n)$, meaning $a = 0$ and $b \neq 0$, then the final solution will be

  \[E\{w(\infty)\} = h \quad \text{with convergence region} \quad 0 < \alpha < \frac{1}{b^2 + \sigma_m^2}\]

So still the optimum Wiener solution will be reached, within a certain convergence region (which may be slightly smaller in the last case). However, if offset is present in both $x(n)$ and $u(n)$, then there will be a deviation from the optimum Wiener solution.
15.5 On the influence of noise upon equalization

Assume that the noise source $u(n)$ is placed at the front-end of the plant, as shown in figure 15.4, then its signal is coloured by the plant. The means that the optimum Wiener solution

$$W^*(z) = \frac{z^{-\Delta} \Phi_{xx}(z) H(z^{-1})}{\Phi_{xx}(z) |H(z)|^2 + \Phi_{uu}(z)}$$

can be rewritten as

$$W^*(z) = \frac{z^{-\Delta} \Phi_{xx}(z) H(z^{-1})}{(\Phi_{xx}(z) + \Phi_{uu}(z)) |H(z)|^2}$$

with $\Phi_{xx}(z) = \sigma_x^2$ and $\Phi_{uu}(z) = \sigma_u^2$

$$W^*(z) = \frac{z^{-\Delta} \sigma_x^2}{(\sigma_x^2 + \sigma_u^2) H(z)}$$

holds, which can be seen as

$$W^*(z) = \frac{C \cdot z^{-\Delta}}{H(z)}$$

a scaled, frequency-independent, version of the noise-less optimum Wiener solution.
Onwards to phantom source creation by means of adaptive filtering 112

15.6 Simulation and measurement sessions: used programs

15.6.1 Simulation sessions with standard LMS

Emulation

<table>
<thead>
<tr>
<th>DSP</th>
<th>program</th>
<th>data</th>
<th>function</th>
<th>card</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>whi.asm</td>
<td>datafile.ld2</td>
<td>white noise generator</td>
<td>XMOT</td>
</tr>
<tr>
<td>01</td>
<td>noi.asm</td>
<td>datafile.ld2</td>
<td>“unknown” system</td>
<td>XMOT</td>
</tr>
<tr>
<td>10</td>
<td>2big.asm</td>
<td>datafile ld2</td>
<td>adaptation program</td>
<td>XFIR</td>
</tr>
<tr>
<td>11</td>
<td>2bm.asm</td>
<td>datafile ld2</td>
<td>mean value calculator</td>
<td>XFIR</td>
</tr>
</tbody>
</table>

The file datafile ld2 contains the data of the “unknown” system, which is modeled FIR.

Deconvolution

<table>
<thead>
<tr>
<th>DSP</th>
<th>program</th>
<th>data</th>
<th>function</th>
<th>card</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>whi.asm</td>
<td>datafile.ld2</td>
<td>white noise generator</td>
<td>XMOT</td>
</tr>
<tr>
<td>01</td>
<td>noi.asm</td>
<td>datafile.ld2</td>
<td>“unknown” system</td>
<td>XMOT</td>
</tr>
<tr>
<td>10</td>
<td>2big.asm</td>
<td>datafile ld2</td>
<td>adaptation program</td>
<td>XFIR</td>
</tr>
<tr>
<td>11</td>
<td>2bm.asm</td>
<td>datafile ld2</td>
<td>mean value calculator</td>
<td>XFIR</td>
</tr>
<tr>
<td>20</td>
<td>ifir.asm</td>
<td>datafile ld2</td>
<td>“unknown” system</td>
<td>XFIR</td>
</tr>
</tbody>
</table>

The output of DSP 20 will show the approximation to the overall system’s input white noise signal.

15.6.2 Simulation sessions with pseudo filtered-x LMS

<table>
<thead>
<tr>
<th>DSP</th>
<th>program</th>
<th>data</th>
<th>function</th>
<th>card</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>whi.asm</td>
<td>datafile.ld2</td>
<td>white noise generator</td>
<td>XMOT</td>
</tr>
<tr>
<td>01</td>
<td>softy1m.asm</td>
<td>datafile ld2</td>
<td>“unknown” system</td>
<td>XMOT</td>
</tr>
<tr>
<td>11</td>
<td>hfm.asm</td>
<td>datafile ld2</td>
<td>mean value calculator</td>
<td>XFIR</td>
</tr>
</tbody>
</table>
15.6.3 Measurement sessions with standard LMS

Emulation

Table 15.4: Programs needed for plant equalization

<table>
<thead>
<tr>
<th>DSP</th>
<th>program</th>
<th>data</th>
<th>function</th>
<th>card</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>accoup.asm</td>
<td>datafile.ld2</td>
<td>AC-coupling filter</td>
<td>XMOT</td>
</tr>
<tr>
<td>10</td>
<td>m2big.asm</td>
<td>datafile.ld2</td>
<td>adaptation program</td>
<td>XFIR</td>
</tr>
<tr>
<td>11</td>
<td>2bm.asm</td>
<td>datafile.ld2</td>
<td>mean value calculator</td>
<td>XFIR</td>
</tr>
</tbody>
</table>

The input to the overall system is a white noise signal, generated by the HP 3562A signal analyzer. This signal passes A/D conversion, so an AC-coupling filter has to be used.

The difference between 2big.asm and m2big.asm is, that in measurements, an extra queue for the compensation of $\Delta_{plant}$ has to be included.

Deconvolution

Table 15.5: Programs needed for plant equalization

<table>
<thead>
<tr>
<th>DSP</th>
<th>program</th>
<th>data</th>
<th>function</th>
<th>card</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>accoup.asm</td>
<td>datafile.ld2</td>
<td>AC-coupling filter</td>
<td>XMOT</td>
</tr>
<tr>
<td>10</td>
<td>i2big.asm</td>
<td>datafile.ld2</td>
<td>adaptation program</td>
<td>XFIR</td>
</tr>
<tr>
<td>11</td>
<td>2bm.asm</td>
<td>datafile.ld2</td>
<td>mean value calculator</td>
<td>XFIR</td>
</tr>
</tbody>
</table>

The program i2big.asm may be used for measurements as well, only the length of a queue has to be changed for the compensation of $\Delta_{plant}$. 
15.7 On the influence of the adaptive filter length in the frequency domain

A loudspeaker → ear (or microphone) impulse response has a smooth tail. In order to obtain a good emulation, the most important part of that impulse response must be present in the adaptive weight vector. The smooth tail will be ignored, and thus cut off. The error made, is caused by the energy present in the tail.

As an example, two 1024 point lowpass filters, with different cut-off frequencies are examined. Only the first part of the impulse response is shown, since the tail is smooth (the full length is approximately 0.023 sec.). When the length of the adaptive filter is only 32 taps, then it will only have the first part, indicated by the arrow, in its weight vector.

This means that the most important part of the impulse response of the filter with high cut-off frequency is present in the adaptive weight vector, the tail is very smooth.

Though its tail is smooth, the filter with low cut-off frequency still has an important part of its impulse response outside the adaptive weight vector.

![Figure 15.5: Impulse responses of lowpass filters](image)

An $I$-point DFT divides the frequency range $0 < f < f_s$ into $I$ equal frequency bands (equally spaced bins), each of them having a bandwidth $B = f_s/I$. For the frequency resolution $F_r = I/f_s$, points/Hz. holds. So for an adaptive filter with filter length $I$, low $I$ will give a low frequency
Onwards to phantom source creation by means of adaptive filtering

resolution and the frequency characteristic will thus show a global shape. By padding zeroes to a short impulse response, the resolution improves.

The loudspeaker → microphone (or ear) transfer function will be displayed on a logarithmic x-axis, because this is good related to the way of human perception. Because of representing data in this way, the frequency bins are not equally spaced along the x-axis (frequency) anymore, since the lower frequency part is stretched, and the higher frequency part is compressed. So more bins are present at high frequencies, and less bins at lower frequencies.

15.8 Temperature effects

When extensively using many chips on the card, the temperature of the inside rack (which holds the signal processing cards, signal converters and power supplies) increases. This may lead to incorrect functioning of the electronic parts. If adaptive filters are used, this effect can be seen as the adaptive filter coefficients passing through the optimum Wiener solution until all coefficients reach the technical maximum value of “+1”. A good solution to this problem is placing a fan in front of the rack.
Acknowledgments

I would like to acknowledge prof. Van Bokhoven and dr. Druyvesteyn for providing me with an industrial research final project. I would like to thank ing. Ronald Aarts and dr. Piet Sommen for their supervision, help, ideas and support throughout my task. Furthermore ing. Paul Boers for his great help during my efforts to handle the DSP system. Last but not least, all members and all stagiares of the acoustics group at NatLab, for the pleasant time.
Bibliography

*A novel filtered-x LMS algorithm and its application to active noise control*
Signal Processing 6
pp. 1709, 1992

*Active noise cancellation using a modified form of the filtered-x LMS algorithm*
Signal Processing 6
pp. 1053, 1992

*Spatial hearing*
MIT Press
1983

*DSPCON 6.1: a control program for user applications running on Motorola DSP 56001/DSP 56002*
NAT. LAB. REPORT NR. 6661

*Signal processing for binaural applications: adapting loudspeaker signals for headphone reproduction*
NAG Journaal nr. 104, September 1990

[6] P.M. Clarkson
*Optimal and adaptive signal processing*
CRC Press
London, 1993

*Head-related two-channel stereophony with loudspeaker reproduction*
Journal of the Acoustical Society of America
Vol. 50, pp. 1109, October 1971
Onwards to phantom source creation by means of adaptive filtering

   Ein Verfahren zur richtungstreuen Schallabbildung des oberen Halbraumes über zwei Lautsprecher
   Acustica

   Multiple-point equalization in a room using adaptive digital filters
   Journal of the Audio Engineering Society
   Vol. 37, No. 11, November 1989

    A multiple error LMS algorithm and its application to the active control of sound and vibration
    IEEE Transactions on Acoustics, Speech and Signal Processing
    Vol. 35, No. 10, October 1987

    Real-time digital signal processing applications with Motorola’s DSP56000 family
    Prentice-Hall
    Englewood Cliffs NJ, 1990

    XMOT users manual
    NAT. LAB. REPORT NR. AR29-R-57/92

    Adaptive loudspeaker system
    Journal of the Audio Engineering Society
    Vol. 37, No. 11, November 1989

    Active Control of Sound
    Academic Press
    London, 1992

    Adaptive inverse filters for stereophonic sound reproduction
    IEEE Transactions on Acoustics, Speech and Signal Processing
    Vol. 40, No. 7, July 1992

[16] P. Parker
    Acoustics source book
    McGraw-Hill
    New York, 1988
Onwards to phantom source creation by means of adaptive filtering

Computer simulation of sound transmission in rooms
IEEE Convention Records
Part 7, 1963

[18] P.C.W. Sommen
Tijddiscrete signaalbewerking
Diktaat nr. 5739, TU Eindhoven
1990 & 1993

[19] M.J. Verbunt
Adaptieve signaalbewerking voor luidsprekersimulatie bij hoofdtelefoonweergave
Afstudeerverslag, TU Eindhoven
Eindhoven, 1991

Adaptive Signal Processing
Prentice-Hall
Englewood Cliffs NJ, 1991

[21] DSP56000/DSP56001 digital signal processor user’s manual
Motorola Ltd, European Literature Centre

[22] Motorola DSP 56000 presentation
Philips diktaat

[23] Motorola semiconductor technical data
Advance information ADI257

[24] Motorola semiconductor technical data
Advance information ADI1257R1

[25] Motorola semiconductor technical data
Technical summary BR283
List of Figures

2.1 Set-up to generate a phantom source ........................................ 6
2.2 Block scheme of the desirable model ....................................... 7
2.3 The plant, the model and the real situation ................................. 11

3.1 Adaptive inverse model without delay ....................................... 14
3.2 Adaptive inverse model with delay ........................................... 15
3.3 Adaptive control system ....................................................... 16
3.4 Adaptive control system with plant noise ................................... 17
3.5 Adaptive inverse before the plant with additive noise ..................... 18
3.6 Filtered-\( x \) algorithm ..................................................... 19

4.1 Usage of an equalization filter in a sound-reproduction system ............ 21
4.2 Single-point equalization problem ........................................... 22
4.3 Reversion of transfer functions .............................................. 23
4.4 Adaptive loudspeaker system .................................................. 24

5.1 Multiple-point equalization problem ........................................ 25
5.2 Multi-channel system for random sound control ............................... 27
5.3 Block diagram of the multi-channel system for random sound control ...... 28
5.4 Multi-channel system for phantom source generation ........................ 29
5.5 Block diagram of the multi-channel system for phantom source generation . 29
5.6 Block diagram of a feed-forward active sound control system .......... 30

6.1 Adaptation of a loudspeaker \( \rightarrow \) ear acoustical path ..................... 33
6.2 The choice of \( \Delta \) is critical due to its influence on \( w^*(n) \) ............... 33
6.3 Adaptation of an inverse loudspeaker \( \rightarrow \) ear acoustical path ............ 34

7.1 AFIR in direct model, with and without noise ................................ 37
7.2 AFIR in inverse model ....................................................... 41

8.1 Complete digital signal processing system .................................... 43
8.2 XFIR card layout ............................................................. 44
8.3 Operational modes of the CAFIR chip ....................................... 46
Onwards to phantom source creation by means of adaptive filtering

8.4 FIR layout ........................................ 47
8.5 AFIR layout ..................................... 47
8.6 Cascade interface of DSP 56200 .................. 48

9.1 XMOT card layout ................................. 51

10.1 Flowchart of a simulation run ................. 54
10.2 Magnitude characteristic depending on I .... 59
10.3 Magnitude characteristic depending on I .... 59
10.4 Magnitude characteristic depending on I .... 60
10.5 Magnitude characteristic depending on u(n) .. 61
10.6 Magnitude characteristic depending on s(n) ... 63
10.7 Magnitude characteristic depending on s(n) ... 63
10.8 \(H_{HP3562A} \cdot W = f(\alpha)\) ..................... 68
10.9 Magnitude characteristic depending on \(\Delta\) .. 70
10.10 Magnitude characteristic depending on \(\Delta\) . 70
10.11 \(H_{HP3562A} \cdot W = f(\Delta)\) .................... 71
10.12 \(H_{HP3562A} \cdot W = f(\Delta)\) .................... 72
10.13 Magnitude characteristic depending on I ...... 73
10.14 \(H_{HP3562A} \cdot W = f(I)\) ....................... 74
10.15 \(H_{HP3562A} \cdot W = f(C)\) ....................... 76

11.1 Pseudo filtered-x algorithm ..................... 78
11.2 Magnitude characteristic depending on \(\Delta\) .. 81
11.3 Magnitude characteristic depending on \(\Delta\) .. 81
11.4 \(H_{HP3562A} \cdot W = f(\Delta)\) .................... 82
11.5 \(H_{HP3562A} \cdot W = f(\Delta)\) .................... 83
11.6 Magnitude characteristic depending on I ...... 84
11.7 \(H_{HP3562A} \cdot W = f(I)\) ....................... 85
11.8 Magnitude characteristic depending on C ...... 87

12.1 Experimental arrangement ....................... 89
12.2 Emulation measurement with the signal analyzer 90
12.3 Impulse response loudspeaker \(\rightarrow\) microphone using the analyzer 91
12.4 Emulation measurement with adaptive filters ... 91
12.5 Impulse response loudspeaker \(\rightarrow\) microphone using AFIRs 92
12.6 Transfer function loudspeaker \(\rightarrow\) microphone 93
12.7 Impulse response inverse loudspeaker \(\rightarrow\) microphone using
direct inversion .................................................. 94
12.8 Equalization measurement with adaptive filters 95
12.9 Impulse response inverse loudspeaker \(\rightarrow\) microphone using
AFIRs .............................................................. 96
12.10 Transfer function inverse loudspeaker \(\rightarrow\) microphone using
direct inversion .................................................. 96
Onwards to phantom source creation by means of adaptive filtering

12.11 Transfer function inverse loudspeaker → microphone using AFIRs .......................... 97
12.12 \( H_{HP3562A} \cdot W \) ..................................................................................... 97
12.13 Transfer function inverse loudspeaker → ear ......................................................... 98
12.14 Impulse response inverse loudspeaker → ear using AFIRs ............................... 99
12.15 Transfer function inverse loudspeaker → microphone using AFIRs ................. 99
12.16 \( H_{HP3562A} \cdot W \) ..................................................................................... 100

15.1 Noise canceller configuration ............................................................................. 106
15.2 Noise canceller configuration, with noise at the reference input 107
15.3 Noise canceller configuration, with noise at the primary input 108
15.4 Inverse filtering with noise at the front-end of the plant .................................. 111
15.5 Impulse responses of lowpass filters ................................................................. 114
List of Tables

8.1 CAFIR configuration dependence on jumper setting .......................... 48
10.1 Results of $\alpha$-variation ....................................................... 57
10.2 Critical values for $\alpha$ .......................................................... 58
10.3 Results of $I$-variation ............................................................. 58
10.4 Results of $C$-variation ............................................................. 62
10.5 Results of $C$-variation ............................................................. 64
10.6 Results of $\alpha$-variation, $I = 200$ .............................................. 66
10.7 Results of $\alpha$-variation, $I = 300$ .............................................. 66
10.8 Results of $\Delta$-variation .......................................................... 69
10.9 Results of $I$-variation ............................................................. 73
10.10 Results of $C$-variation ............................................................ 75

11.1 Results of $\alpha$-variation .......................................................... 80
11.2 Results of $\Delta$-variation .......................................................... 80
11.3 Results of $I$-variation ............................................................. 84
11.4 Results of $C$-variation ............................................................. 86

15.1 Programs needed for plant emulation ........................................... 112
15.2 Programs needed for plant equalization ....................................... 112
15.3 Programs needed for plant equalization ....................................... 112
15.4 Programs needed for plant equalization ....................................... 113
15.5 Programs needed for plant equalization ....................................... 113