Digital Microphone Array

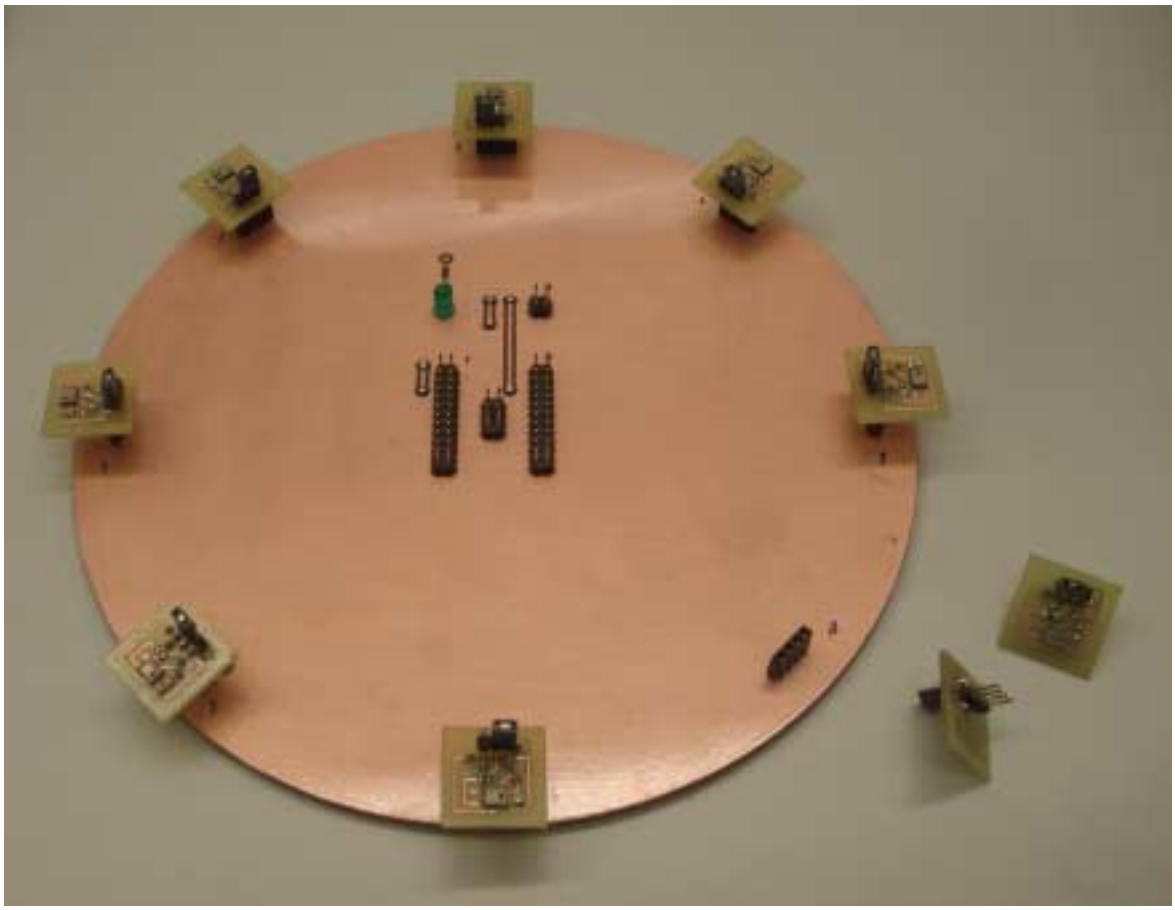
Design, Implementation and Speech Recognition Experiments

Erich Zwyssig

A thesis submitted for the degree of Master of Science

The University of Edinburgh

21st August 2009



Abstract

The instrumented meeting room of the future will help meetings to be more efficient and productive. One of the basic components of the instrumented meeting room is the speech recording device, in most cases a microphone array. The two basic requirements for this microphone array are portability and cost-efficiency, neither of which are provided by current commercially available arrays. This will change in the near future thanks to the availability of new digital MEMS microphones. This dissertation reports on the first successful implementation of a digital MEMS microphone array. This digital MEMS microphone array was designed, implemented, tested and evaluated and successfully compared with an existing analogue microphone array using a state-of-the-art ASR system and adaptation algorithms. The newly built digital MEMS microphone array compares well with the analogue microphone array on the basis of the word error rate achieved in an automated speech recognition system and is highly portable and economical.

Declaration of Originality

I hereby declare that the research recorded in this dissertation and the dissertation itself was composed and originated entirely by myself at the School of Philosophy, Psychology & Language Sciences (PPLS) at The University of Edinburgh. I have read and understood The University of Edinburgh guidelines on Plagiarism and declare that this written dissertation is all my own work except where I indicate otherwise by proper use of quotes and references.

Erich Zwyssig

Edinburgh, 21st August 2009

Acknowledgements

The idea of building a digital MEMS microphone arose during a meeting with Steve Renals when he showed me the latest analogue microphone array, the size of a tea cup and an amplifier ten times that size. I then asked why no one had produced a digital microphone array. Fifteen weeks later it is done. I wish to thank Steve Renals for being a good supervisor, providing me with somewhere to work and giving me his support.

Many thanks also to Mike Lincoln for his continuing support with the existing setup and tools and Rob Clark, Simon King and the team at the CSTR for their help with my endless questions and for being guinea pigs for the experiment.

I would like to acknowledge the feedback I received from Jens Krisitan Paulsen and Iain McCowan during the initial drafting of the specification and for sharing their microphone array knowledge with me.

I am very grateful to my past colleagues Erich Spahr and Bernhard Bäriswil from Bernafon Ltd. Switzerland, for putting me in contact with Knowles Electronics; and Birgid Rathbone and Alex Sioufi from Knowels for providing free samples of their digital MEMS microphones.

Thanks are also due to Dave Hamilton and Robert Macgregor from the University of Edinburgh School of Informatics workshop in Appleton Tower for building the digital microphone array PCB.

This project would not have been possible without the support of Wolfson Microelectronics plc and my past colleagues Nick Roche, Anthony Magrath, Mark Brown, Graeme Angus, Ian Smith and Andy Brewster.

I am grateful to Luu Tuan from TI and Chris Wymark from Select Software for their kind support with device and compiler questions.

My classmates Katrien, Kevin, Anna, Phil and Aggeliki provided support, encouragement and good sparring partners - it was a great year.

Finally my thanks go to my partner Lynda for supporting me “durch dick und dünn” and, once again English-ifying my dissertation.

Mountains should be climbed with as little effort as possible and without desire.
The reality of your own nature should determine the speed.
If you become restless, speed up.
If you become winded, slow down.
You climb the mountain in an equilibrium between restlessness and exhaustion.
Then when you're no longer thinking ahead,
each footstep isn't just a means to an end but a unique event in itself.
This leaf has jagged edges.
This rock looks loose.
From this place the snow is less visible, even though closer.
These are things you should notice anyway.
To live only for some future goal is shallow.
It's the sides of the mountain which sustain life, not the top.
Here's where things grow.

Robert M. Pirsig

Table of Contents

List of Figures

List of Tables

Abbreviations

AC'97	Audio Codec Specification 1997
ADC	Analogue Digital Converter
AMI	Augmented Multi-party Interaction
AMIDA	Augmented Multi-party Interaction with Distance Access
ASR	Automatic Speech Recognition
AV	Audio Visual
CIC	Cascaded Integrator-Comb
CMLLR	Constrained MLLR (Maximum Likelihood Linear Regression)
CMOS	Complementary Metal Oxide Semiconductor
CSTR	Centre for Speech Technology Research
DAC	Digital Analogue Converter
DMA	Digital Microphone Array / Direct Memory Access
DSP	Digital Signal Processing
DSR	Distant Speech Recognition
EEPROM	Electrically Erasable Programmable Read-Only Memory
FE	Front End
FIFO	First-In First-Out (memory)
FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array
FU	Functional Unit
FW	FirmWare
HDA	High-Definition Audio
HDI	Human Device Interface
HDL	Hardware Description Language
HiFi	High Fidelity
HMM	Hidden Markov Model
HTK	HMM Tool Kit
HW	HardWare
I^2S	Inter-IC Sound
IC	Integrated Circuit
IF	InterFace
IMR	Instrumented Meeting Room
I^2S	Inter-IC (Integrated Circuit) Sound

JFET	Junction Field Effect Transistor
LPC	Linear Predictive Coding
LPF	LowPass Filter
MAP	Maximum A Posteriori
MC	Multi-Channel
MEMS	Micro Electro Mechanical Device
MFCC	Mel Frequency Cepstrum Coefficient
MLLR	Maximum Likelihood Linear Regression
MLLRMEAN	Means-Only MLLR (Maximum Likelihood Linear Regression)
MS	Microsoft
MU	Mixer Unit
µC	microController
OS	Operating System
PC	Personal Computer
PCM	Pulse Code Modulation
PDM	Pulse Density Modulation
RAM	Random Access Memory
RC	Resistor Capacitor
RP	Received Pronunciation
SDM	Sigma-Delta Modulation
SLIMbus	Serial Low-power Inter-chip Media Bus
SMD	Surface Mount Device
SNR	Signal to Noise Ratio
SP	Signal Processing
S/PDIF	Sony/Philips Digital Interconnect Format
STC	STreaming Controller
SW	SoftWare
THD	Total Harmonic Distortion
TDM	Time-Domain Multiplexed
TI	Texas Instruments
TMF	Transform Model File
WER	Word Error Rate
WSJ	Wall Street Journal
USB	Universal Serial Bus

Introduction

The future, according to the European Union consortium AMI/AMIDA, will bring fewer but more productive meetings [97]. Physical meetings will be mixed with virtual meetings, i.e. past meetings, remote meetings or parallel meetings that collide with each other. Changes in technology and the smart meeting room will lead to new business processes for people to take advantage of these new tools, tools which will augment the meeting experience.
What people most want from meetings, according to a survey [97], is shared notes (minutes), the agenda and access to documents and presentations after the meeting. They want to search for decisions taken, participants and speakers and topics discussed during the meeting. When delayed or absent, people wish to see an automated summary, a list of action points and an overview of the content of the meeting, all at their fingertips by browsing smart minutes.

The Centre for Speech Technology Research (CSTR) of the University of Edinburgh is an academic consortium partner of AMI/AMIDA (Augmented Multi-party Interaction/Augmented Multi-party Interaction with Distance Access). The CSTR conducts basic and applied research for the AMI/AMIDA project and has built a fully-functional audio-visual meeting room (G3.07) for their research, equipping it with audio and video recording and projecting devices. A room of this kind is called an instrumented meeting room (IMR).
An IMR can be used to capture all the information that is used, said and presented during a meeting. The real benefits of the smart meeting room, however, are not in capturing the information but in processing it. The first basic task is to recognise the information produced in the meeting and to use applications to convert this information into human-readable form. The expertise of the CSTR lies in one such application, Distant Speech Recognition (DSR).

Distant Speech Recognition

Distant speech recognition is the discipline of speech recognition when the recording device is separated from the speaker and takes place in an enclosure, i.e. meeting room. Typical problems that need to be addressed are therefore dereverberation, noise and multiple speech sources. Some fields of distant speech recognition are shown in Figure 1.

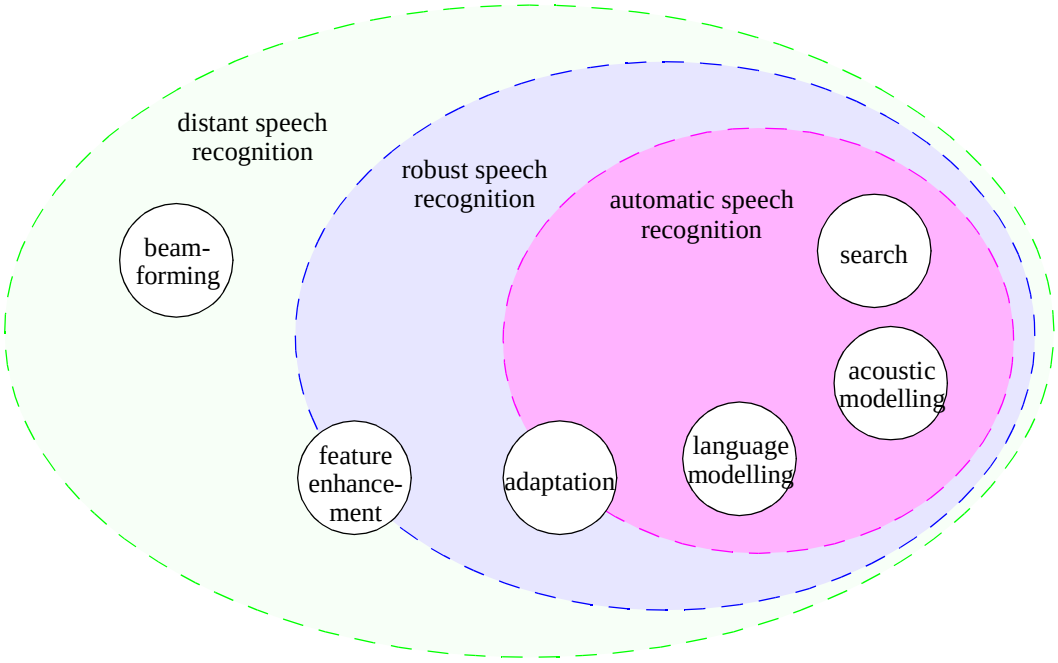


Figure 1 Fields of Speech Recognition (with kind permission of [8])

Speech recognition problems which are considered as having well developed solutions are [77]:

- search, using the Viterbi algorithm
- acoustic modelling, using HMMs
- language modelling, using large corpora
- adaptation, using algorithms such as MLLR or MAP
- feature enhancement, using MFCCs or LPCs
- beamforming, using microphone arrays

While each of the above disciplines on its own is considered to be completed task, the combination of all of the above into a DSR does not yet produce acceptable performances. The aim of the AMI/AMIDA projects is to develop the smart meeting room and the necessary applications. The current implementation of the IMR built by the CSTR has known limitations. These limitations include:

- portability
- cheap commodity HW

The existing IMR (G3.07) is not portable and is built using specially designed equipment housed in large racks.

The aim of this dissertation is to look at one device from the IMR, the microphone array. The microphone array used by the CSTR is an array built of eight expensive analogue

omnidirectional microphones and, in addition to the microphones, requires a rather large and expensive amplifier and A/D converter.

The purpose of this project is to build an inexpensive, more cost-effective and smaller digital MEMS microphone array and compare its performance in terms of ASR with the analogue microphone array.

Digital MEMS microphone array (DMA)

MEMS (Micro Electro Mechanical System) microphones are ultra small microphones that withstand reflow soldering in automatic manufacturing. MEMS microphones have only become commercially available very recently [95] while digital MEMS microphones are only available as samples [127]. The digital MEMS microphone array is the first of its kind to be built from scratch.

Overview

The overall structure of this dissertation is as follows: first a literature review covering MEMS microphones, microphone arrays and automatic speech recognition is presented. This is followed by some digital signal processing background which is required for understanding the building of the digital microphone array. In the second part of this dissertation, first the methodology for comparing the newly built digital microphone array with the existing analogue microphone array is presented, followed by the description of the setup and the presentation of the results. Finally these results are analysed and discussed. Copyright and trademark information and the AMI consent form are included in the appendices.

Review

Building a digital MEMS microphone array is a multi-disciplinary task, as the name already implies. The word digital implies computing and (digital) signal processing (D)SP. MEMS (Micro Electro Mechanical System) indicates a nano-scale system containing both electrical and mechanical components, while a microphone is a device that converts an acoustic signal into an electrical signal. If multiple (e.g. eight) microphones are built together it is called a microphone array.
This section first defines the terms analogue and digital. Next, reviews of MEMS microphones and microphone arrays are presented. Finally, speech recognition including distant speech recognition is reviewed.

Analogue vs. Digital

The real world is analogue. Why then are most devices digital? The answer is simple: processing analogue signals is infinitely more complex and difficult then working in the digital domain. It was only the invention of digital logic that led to the enormous technological progress of the last few decades.
The first problem a system designer using a digital core faces is how to communicate with the real world, which is still analogue. Analogue to digital conversion of input signals (e.g. audio signals) and the conversion from the digital to the analogue domain are therefore a critical system design factor.
The analogue to digital conversion of audio signals is a key issue for the digital microphone array. While analogue microphones were invented more than a century ago, digital microphones have only been around for about a decade and miniaturised MEMS digital microphones have only been available recently [95] [127].
The application for which the digital microphone array presented in this dissertation has been designed is speech recognition in meetings. Existing microphones and microphone arrays use expensive analogue microphones and off-the-shelf converters, i.e. the conversion of the acoustic signal into its digital representation is located several metres from the membrane which captures the acoustic wave. The digital microphone (array) addresses this by putting the conversion less than a millimetre away from the membrane [125]. This aims to simplify the microphone array and reduce costs.

MEMS microphones

Twenty years of research and development were required for the first silicon (MEMS) microphone to be commercially available ([62] in [63], [95]). In the first ten years research focused mainly on the sensor structure (piezoelectric vs. piezoresistive vs. capacitive) and the amplifier that follows the acoustic sensor (see Scheeper et al. [63] (1994) for a review). Manufacturing MEMS microphones involved many problems, e.g. poor uniformity of the microphone sensitivity on the same wafer, sticking of the membrane, non-linear frequency response and process choice (see Ning et al. [55] 1996). From the mid 1990s capacitive sensors were the dominant choice (Pederson et al. [58] 1998). A further ten years of research were necessary to produce MEMS microphones which would operate in a customer application environment. Such requirements are, for example:

- support of SMD mounting (Brauer et al. [23] 2001)
- use of standard CMOS processes (Neumann and Gabriel [54] 2002)
- good SNR performance (Neumann and Gabriel [53] 2003)
- operation with standard supply voltages (Weigold et al. [71] 2006)

Achievements such as listed above led to a breakthrough in MEMS microphone production and usage today. The uptake of MEMS microphones from the mobile phone market, for example, increased from annual sales worth $2 million in 2004 [78] to $140 million in 2006 and is estimated at $922 Million for 2011 [118].

Commercial interest in MEMS microphones has, as a consequence, increased significantly and research and development shifted away from the universities to industry, therefore leading to a change in the type of publication from academic papers to patents. Current commercial interest, for example, are:

- manufacturing and yield improvement [82],
- packaging [81],
- sensitivity improvement [83], and
- performance improvement (using calibration schemes [84])

Currently over twenty companies offer digital MEMS microphones with Akustica, Knowles Acoustics, Sonion MEMS A/S, MEMS Technology Bhd or Wolfson Microelectronics leading the field ([98], [102], [110], [113], [109]). Every single one of the companies mentioned above claims to have the best performing MEMS microphone with key competitive features being

signal to noise ratio (SNR), noise floor, small packaging, linearity, low total harmonic distortion (THD), low cost, clear sound or flat frequency response [58].

Details of MEMS microphone functionality, operation, manufacturing and packaging are presented in what follows.

The key problems manufacturers of (digital) MEMS microphones face are yield, linearity, sticking of the membrane, vibration, moisture sensitivity and calibration [54] [55] [71]. Possible integration of MEMS microphones are either single chip (with the membrane and amplifier on the same chip) or two chip (membrane and amplifier separated) solutions. The following Figure 2 shows a MEMS microphone membrane (i.e. sensor) and a package cross section.

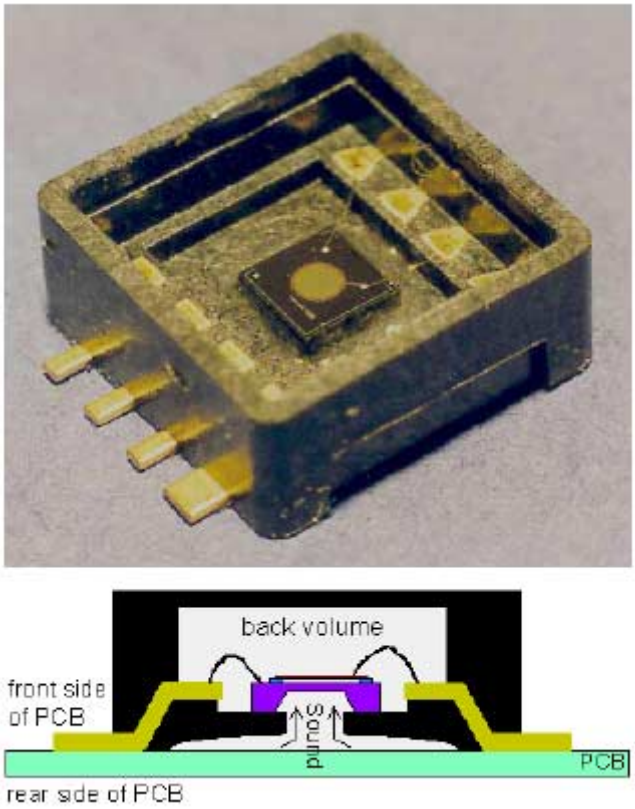


Figure 2 MEMS microphone in SMD package (with kind permission of [23])

In Figure 2 the package and contacts, the sensor on the chip and the bond wires connecting the package with the chip are clearly visible. Please note that in the above integration the sound coupling is from the bottom of the package. Some manufacturers claim that this results in better linearity, reduced moisture sensitivity and better SNR performance [23].

Two principal operations of the MEMS microphone are known. In both cases the MEMS part of the microphone is a membrane, acting as a capacitor.
In the first case the capacitor is sensed using a frequency modulation (FM) scheme [58]. The MEMS membrane, i.e. the capacitor, changes in value stimulated by the sound wave. Being part of an oscillator circuit, the variable capacitor affects, i.e. modulates the oscillation frequency (FM) which is first converted into an analogue signal and then into a digital PDM (pulse density modulation) stream using an oversampling ADC.
The second principle for a MEMS microphone (in wider use) uses a pre-charged capacitor, the MEMS membrane ([53], [54]). Modulation of the capacitor will lead to voltage changes if the charge is maintained, as

$$Q = V \cdot C = const \qquad \text{Eq. (1)}$$

with Q being the charge, V the voltage over the capacitor's terminals and C the capacitance (typically between 1 and 10 nF with an air gap as little as 1 to 5 µm [53] [58]).
Using an amplifier with an ultra-high-ohmic input (i.e. JFET) the voltage variations are amplified and this analogue signal again converted into a digital PDM stream using an oversampling ADC. The principal block circuit diagram of the MEMS amplifier is shown in Figure 3 below.

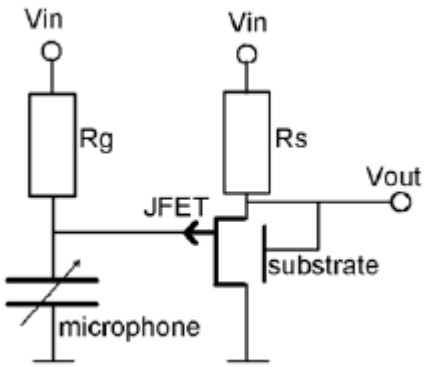


Figure 3 MEMS microphone and amplifier circuit (with kind permission of [23])

One of the greatest difficulties in the manufacturing of MEMS sensors is the process environment. While the CMOS manufacturing technique involves one kind of environment (i.e. specific temperatures and chemicals), the MEMS sensor processes (called micro machining [9])

uses another environment. Both processes remove or add parts of the chip while protecting other parts from been damaged or removed [3] [55].

The final working system needs both the MEMS sensor and the CMOS logic, and some manufacturers of MEMS circuits believe is preferable to have these two different circuits on the same piece of silicon while other prefer two different chips [3] [112].

The following Figure 4 shows the principle mechanical structures (left) and electrical structures (right) of a MEMS sensor, implemented on the same piece of silicon.

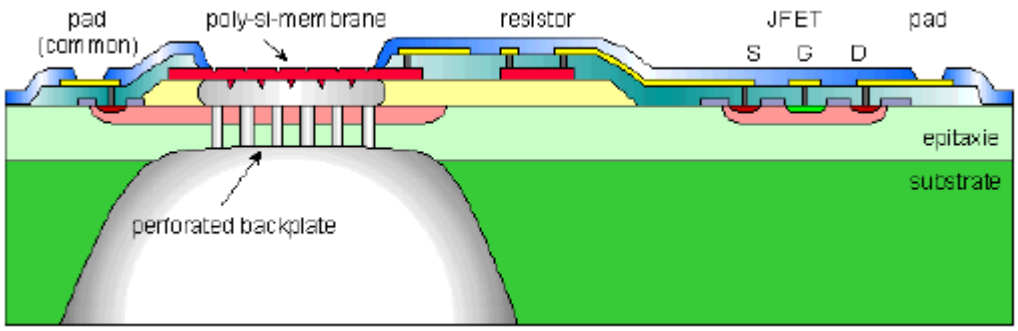


Figure 4 CMOS cross section of MEMS microphone (with kind permission of [23])

The membrane and the amplifier are shown in above Figure 4. Please also note the large amount of substrate material that is removed (using micro machining) in order to free the membrane.

A (MEMS) microphone, a system that converts acoustic waves into electrical signals inherently has to cope with three kinds of noise problems which are:

- acoustical (e.g. sound path to membrane)
- electrical (e.g. leakage, substrate noise, interconnects)
- mechanical (e.g. picking up vibrations)

MEMS microphones are very insensitive to mechanical vibrations due to their small scale and low weight. Electrical problems are minimised for example using one or two chip solutions or varied amplifier schemes. Acoustical problems are addressed as demonstrated in Figure 5.

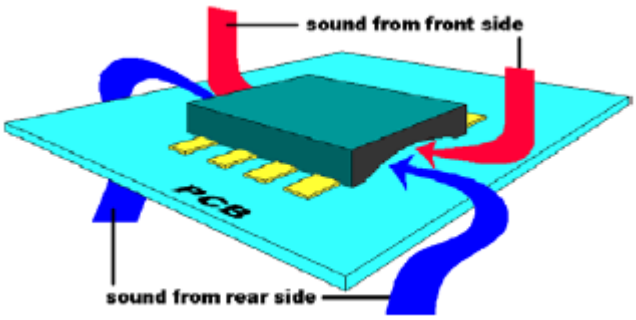


Figure 5 MEMS microphone sound path (with kind permission of [23])

While Akustica or Knowles Acoustics use front openings to the membrane, Analog Devices or Wolfson Microelectronics claim that picking up the sound from the back will give the best performing microphone ([98], [102], [110], [113], [109], [112]). Developing, designing and manufacturing MEMS microphones is an ongoing task and customers will expect the same performance from them as from a conventional microphone nowadays.

Microphone Arrays

Multiple microphones placed at different spacial location form a microphone array. The microphones may be placed in a line, circle of even onto the surface of a sphere [111]. Microphone arrays are able to provide noise robustness and hands-free signal acquisition and are therefore ideally suited for speech processing applications. For recording, even using two microphones instead of one can lead to a significant improvement in system performance. The main aim of increasing the number of microphones in the system is to improve the quality of the input signal, i.e. reduce the effect of typical recording problems. The two most serious problems to overcome when recording speech in a room are noise and echoes, i.e. reverberation.

Noise

Three different kinds of noise fields have been defined for microphone array applications [45]. These are:

- coherent noise fields
- incoherent noise fields
- diffuse noise fields

If the noise propagates to the microphone in a direct, undisturbed way, then it is defined as coherent. A high correlation is found when measuring coherent noise with multiple

microphones [101]. The noise field is incoherent if the noise measured at any spatial location is un-correlated with the noise measured at any other location. In a real environment the energy of the noise propagates in all directions simultaneously. Any location is therefore lowly correlated with any other location, but has approximately the same energy. This is defined as a diffuse noise. For most applications the noise environment can be characterised as diffuse. Diffuse noise is then treated like incoherent noise for simplification [45].

Reverberation

"Reverberation is the collection of reflected sounds from the surfaces in an enclosure" [39] and is defined as the length of time it takes for the reverberation, or echo, to decay to 60dB (one thousandth) from the level of the original sound. Many methods and algorithms have been devised to reduce this ever since speech has been recorded in enclosers (e.g. meeting rooms). For automatic speech recognition systems dereverberation can be addressed, for example,

- before capturing the signal (ie.g. by improving the room acoustics),
- while capturing the signal (e.g. signal processing either in the time or frequency domain),
- before running the speech recognition (e.g. cepstral coefficient manipulation),

or

- while training the speech model (e.g. training the HMMs with recordings in a noisy environment).

Huang et al. provide a good summary in [39].

Microphone array beamforming

In this section the working and benefits of using multiple microphones are presented after first defining a few terms.

Sound propagates in a room like waves. While wave theory is not discussed in this dissertation, some terms are still important. First, the wavelength λ of a signal can be calculated knowing the speed of sound in air c and the frequency of the acoustic signal f as defined in Eq. (2).

$$\lambda = \frac{c}{f} \qquad \text{with c = 343 m/s} \qquad \text{Eq. (2)}$$

The wavelength of speech (for ASR the range of interest is 50 Hz to 8 kHz) travelling in air at 343 m/s will therefore span 4.28 cm (for 8 kHz) to 6.86 m (for 50 Hz). While some researchers have built arrays that span several meters ([30]) another typical choice is to separate the microphones by about ϕ = 0.2 m, defined by the distance between our ears [25]. This

dissertation compares the performance of an analogue microphone array with the newly built digital microphone array. Both arrays have been built using 8 microphones, placed equidistant on a circle with diameter 0.2 m, as shown in [43] and presented in Figure 6.

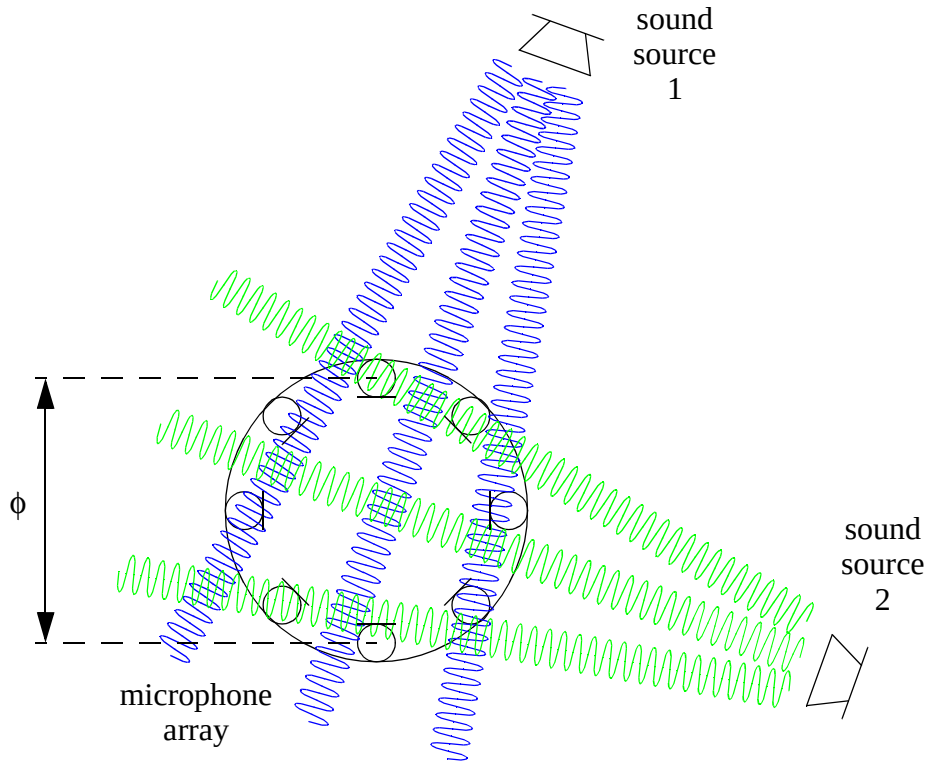


Figure 6 Microphone array setup

Acoustic wave propagation in a room is a function of space and time. In mathematical terms this is a four-dimensional function with three spatial dimensions and one time variable. Although the functionality of microphone arrays can be demonstrated well using formulae, this document explains the benefits and workings of a microphone array diagramatically. The interested reader is referred to [29] and [45] for details.

Looking at a stereo microphone compared to a single microphone it is easy to imagine that any signals coming from the front towards the two microphones are added, while signals from the side cancel each other out. A stereo microphone is therefore the simplest microphone array making use of simple beam steering, i.e. directional sensitivity. This method of adding the microphone signals is called sum beamforming and is shown in Figure 7 below.

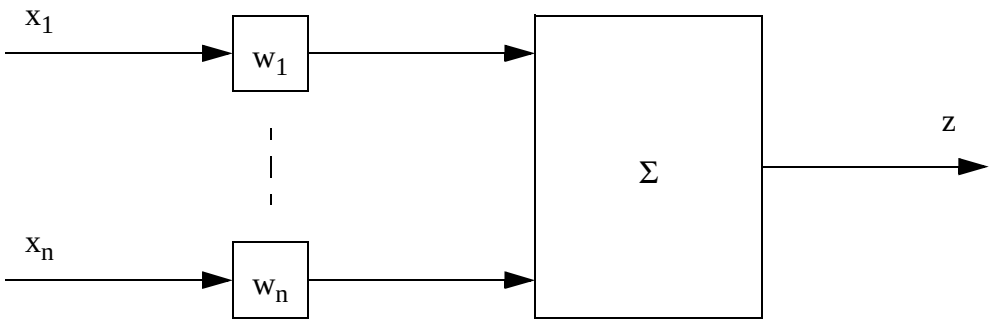


Figure 7 General beamformer structure

Looking at the sum beamforming structure shown in Figure 7, Eq. (3) demonstrates the mathematics of such a system with x_n being the input signal of the individual microphone and w_n being the scaling weight with which x_n is multiplied before being summed.

$$z = \begin{bmatrix} w_1 \\ \dots \\ w_n \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ \dots \\ x_n \end{bmatrix} = \boldsymbol{w} \cdot \boldsymbol{x} \qquad \text{Eq. (3)}$$

Adding delays to the weighted individual signals gives a delay-sum beamformer.

Delay-sum beamforming

The most basic microphone array beamformer is the delay-sum beamformer. Indeed, the best results in distant speech recognition (DSR) using microphone arrays are achieved with modified delay-sum beamformers [29]. The basic block diagram of a delay-sum beamformer is shown in Figure 8.

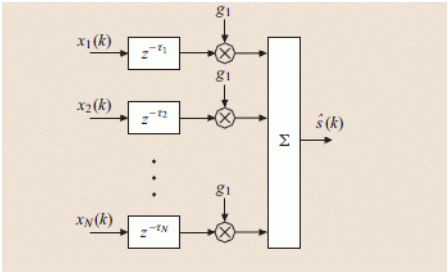


Figure 8 Delay-sum beamformer block diagram (with kind permission of [29])

Implementing the delay-sum beamformer requires nothing more than memory (either in RAM or flip-flop registers if implemented in Hardware or a stack or other kinds of memory if implemented in Software), a multiplier and an adder, as shown in Figure 9.

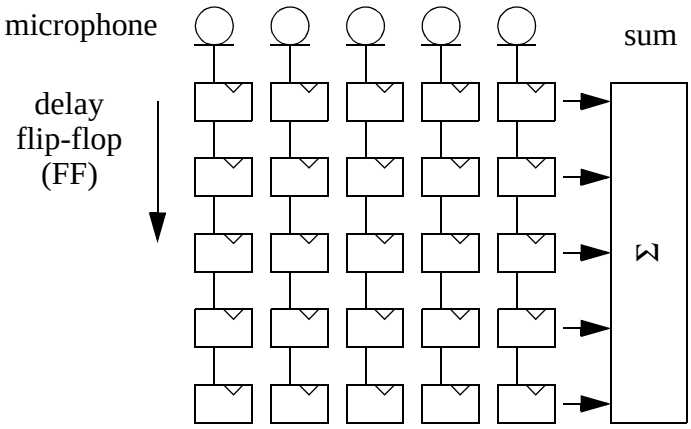


Figure 9 Delay-sum beamformer

The following Figure 10 and Figure 11 illustrate how a microphone array can look forward and sideways, depending on how the delays of the individual channels are set, i.e. depending on which delay element is selected for the sum operation.

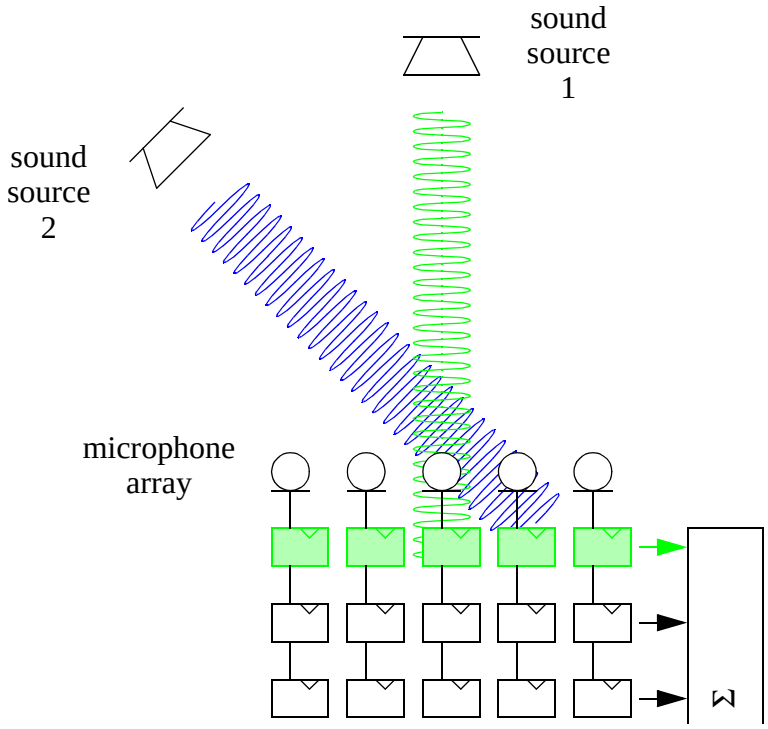


Figure 10 Delay-sum beamformer (looking forward)

In Figure 10 the beam is set to look forward and the delay elements highlighted in green are selected for the sum operation, i.e.their weights are set to 1 while all others are set to 0.

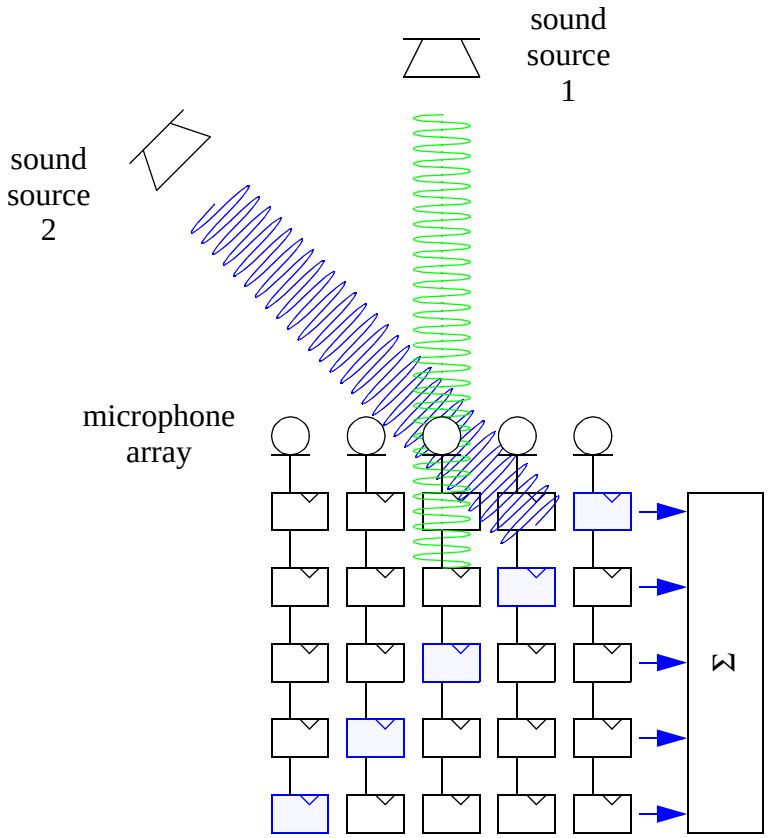


Figure 11 Delay-sum beamformer (looking sideways)

In Figure 11 the beam is steered 45º to the right and the delay elements highlighted in blue are selected for the sum operation, i.e.their weights are set to 1 while all others are set to 0.
This delay-sum scheme can easily be applied to the digital microphone array as shown in Figure 12.

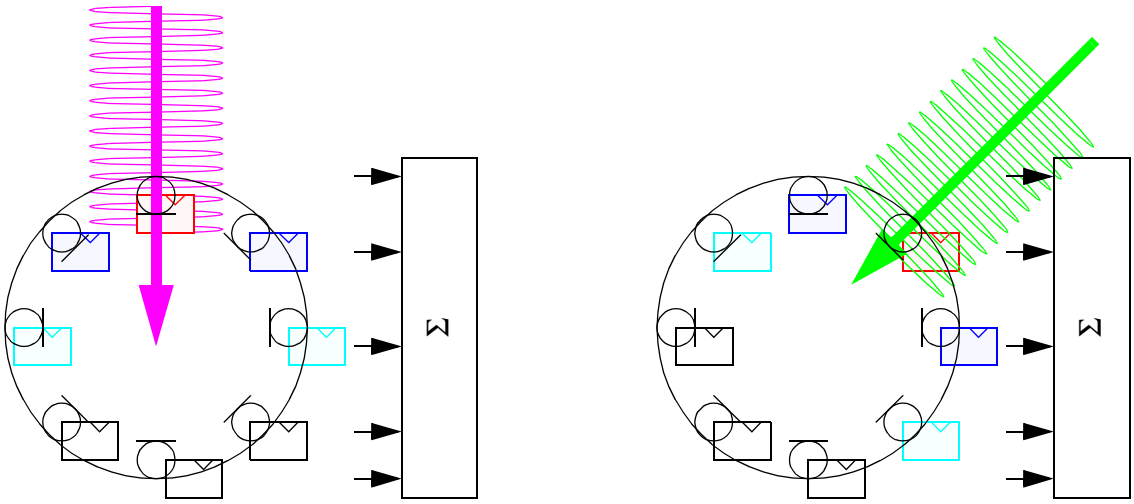


Figure 12 Steering the digital microphone array

In Figure 12 the delay elements highlighted in red point to the front and the individual weights and delays of the eight microphones on the microphone array circle are set accordingly, e.g. for both scenarios illustrated in Figure 12 $^{1}/_{5}$ of red, $^{1}/_{10}$ of both blues delayed by $t_{d(1)}$[1], $^{1}/_{10}$ of both cyans delayed by $t_{d(2)}$, etc.

Research in microphone arrays has benefited greatly from research in sonar, radar, seismology and radio technology which laid all the foundations of wave propagation [45]. Knapp et al. [42] defined the principle of beam steering as early as 1976. They state that “the direction of an input signal” can be “simply estimated by the abscissa value at which the cross-correlation peaks”. The principle block diagram of a circuit that determines the direction of an incoming sound, as presented by Knapp, is illustrated in below.

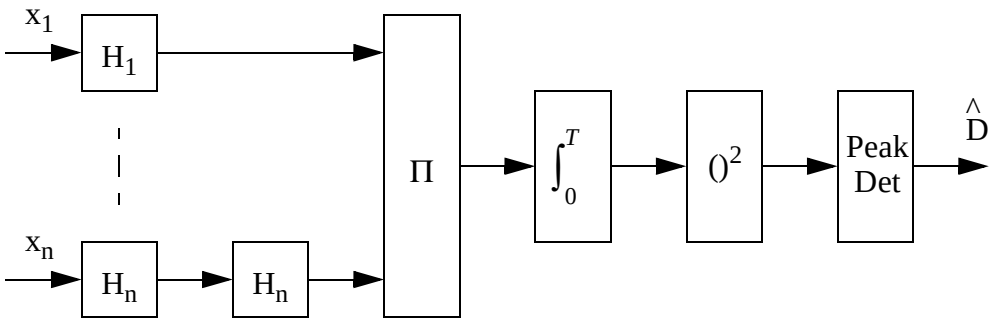


Figure 13 Circuit to determine input signal direction (as defined by [42])

1. $t_{d(n)}$ can be determined by the distance of the array microphones given the desired angle of the steered beam, the speed of sound c and the sample rate f_s.

The direction of the input signal is determined by filtering the different microphone input signals (only vowel sounds are of interest), delaying them (according to the direction the signal is presumed to come from), then multiplying, integrating and squaring them to determine the cross correlation. The direction is determined by the peak cross correlation by varying the delays.

A two-dimensional microphone array using two beams, a search beam and a recording beam was successfully implemented and tested by Flanagan et al. [30] in 1985. Cox et al. [26] proved that the delay-sum beamformer achieves near optimum performance and applied sideslope shading and oversteering of the beam to increase the array sensitivity. They later introduced the scaled projection algorithm to further fine-tune adaptive beamforming [27]. In 2001 McCowan [45] presented a very good overview of beamforming techniques and their advantages and disadvantages. Lincoln et al. [43] used the results and algorithms obtained by McCowan to design and test a multi-channel Wall Street Journal corpus for speech recognition for the AMI Meeting Corpus (see [43] and e.g. [34]). This system was improved further by Himawan et al. [37] who applied the principles to blind source speech separation.

Delay-filter beamforming

The delay-sum beamformer, as shown in Figure 9 above, can easily be modified into a delay-filter beamformer by weighting and summing not only one element per microphone delay chain but all of them, as in a two-dimensional FIR filter[1].

Superdirective microphone arrays

A sound source which is on the same axis as the aperture, i.e. straight in front of the centre of the linear array, is defined as an endfire source [45]. Conventional linear arrays have a directivity that is almost proportional to the number of sensors N. The theoretical directivity limit for linear endfire arrays in a diffuse noise field is N^2 [45]. Beamforming algorithms which exploit this capability are called superdirective beamformers. Superdirective beamformers aim to optimise the array weight vector to maximise the gain by using the Lagrange method as defined by Cox et al. [27]. Cox et al. also present a more practical solution by iteratively adjusting the Lagrange multiplier using the white noise gain constraint. Although computationally more complex, this method works well in practise as the beamforming filters

1. For an introduction to FIR filters please refer to "(Half-Band) FIR filter" on page 53

depend only on the array geometry and source location and so only need to be updated once for a given configuration.
Note: See McCowan [45] and Bitzer and Simmer [2] for a summary on superdirective microphone arrays and all the possible methods that can be applied to improve the performance of the basic delay-sum beamformer.

Post-filtering

In addition to the different techniques to improve the beam shape, beam sideslopes and the beam search algorithm, post filtering using Wiener filters has demonstrated significantly improved signal quality. The basic problem is that beamforming is an optimisation of a narrowband input, i.e. super directional microphone arrays find the best solution with respect to a narrowband input signal. Multi-channel Wiener filtering, a broadband multi-channel filtering technique, has been shown to give significant improvement in SNR (see Simmer et al. [7] for a summary) and is widely used in speech recognition systems (e.g. [43]).

A current, comprehensive summary of microphone arrays, beamforming, gain optimisation and adaptive microphone arrays is given by Elko and Meyer [29].

In the reminder of this section two features of the microphone array are reviewed more closely. These are

- spacial aliasing

and

- directivity

Microphone Array Spatial Aliasing

When considering a microphone array and the propagation of sound it should be borne in mind that the physical distance of the individual microphones has the same effect in space as the time distance has on the signal frequency when sampled. This effect is called aliasing [96].
Spacial aliasing, as defined in Eq. (4) (from [45]), is the effect of being unable to tell where the signal actually comes from if it is above a certain frequency.

$$d < \frac{\lambda_{min}}{2} = \frac{c}{2f} \qquad \text{Eq. (4)}$$

For our microphone d and f_{max} are:

$$d = \phi \cdot \sin\left(\frac{\pi}{n}\right) = 0.2m \cdot \sin\left(\frac{\pi}{n}\right) = 0.076m \qquad \text{Eq. (5)}$$

$$f_{max} = \frac{c}{2 \cdot d} = \frac{343\frac{m}{s}}{2 \cdot 0.076m} = 2.24kHz \qquad \text{Eq. (6)}$$

This implies that input signal components with f > 2.24 kHz can not be located with our microphone array. This is not an issue in so far as the beam search algorithm usually looks for phones in the received speech and the 1st and 2nd formants of either a man's, woman's or child's voice are well below this frequency [5].

Please note that the values of d and f_{max} in Eq. (5) and Eq. (6) are approximations, taking the distance of two microphones as the minimum distance of the array.[1]

The following Figure 14 shows spacial aliasing.

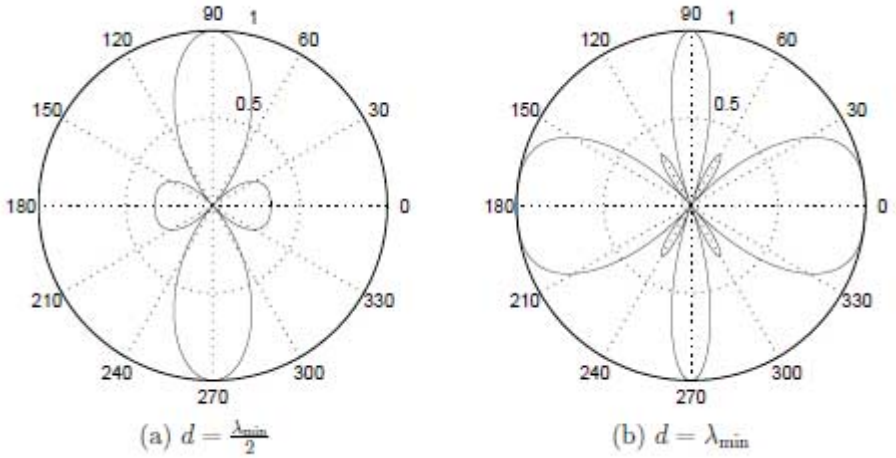


Figure 14 Spatial Aliasing in Beamformer (with kind permission of [45])

Figure 14 demonstrates how the side-slopes grow as d approaches λ until a signal wave entering the array at 90º cannot be distinguished from one entering at 180º.

1. For a 8 microphone array, with microphone 1 to 8 in a circle, the signal arriving at 90º from a line from microphone 1 and 8, the minimal distance d is the height of the trapeze mic1 - mic2 - mic7 - mic8.

Microphone Array Directivity

The combination of sound of a certain frequency travelling at the speed of sound in the air being captured with multiple microphones at a defined distance has another side effect, the effect of directivity. An example of directivity is presented in Figure 15 below.

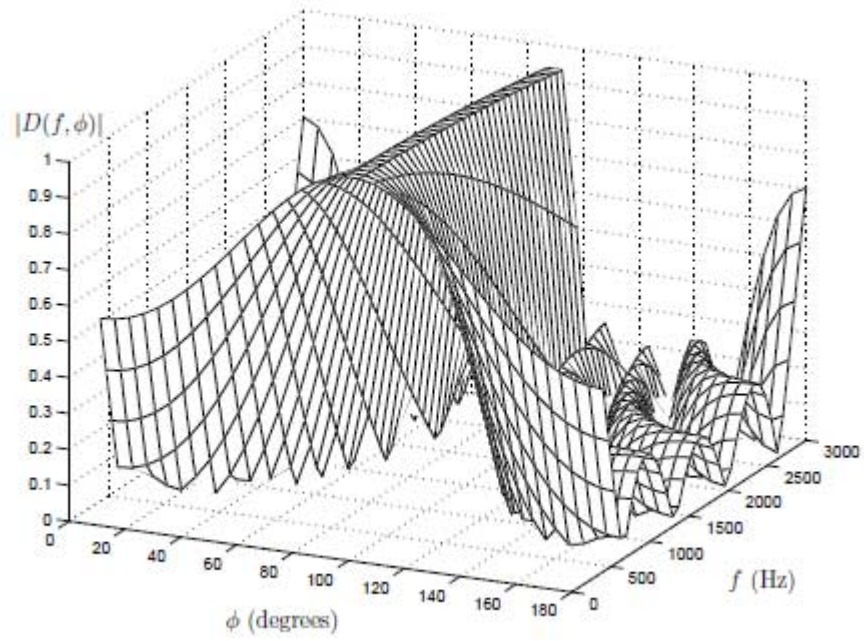


Figure 15 Directivity pattern for 400 Hz $\leq$ f $\leq$ 3 kHz (with kind permission of [45])

Figure 15 above shows the directivity pattern of sound from 400 Hz to 3 kHz, recorded with 5 microphones place in a line at 0.1 m spacing. Note that the width of the beam is not constant over the frequency. This might or might not be a problem for the system in which the array is being used.

A constant beamwidth microphone array can be designed using nested microphone arrays, as shown in Figure 16 below.

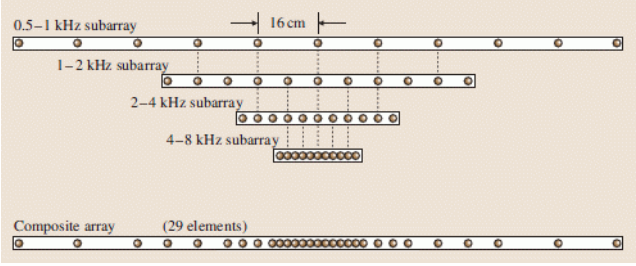


Figure 16 Nested microphone array (with kind permission of [29])

Using multiple arrays with different equi-distantly place microphones as shown in Figure 16 each array is optimised for a specific frequency band. This arrays can then be combined, saving in the number of microphones required and the space needed to build such an array. The combined constant-beamwidth microphone array system directivity pattern is presented in Figure 17 below.

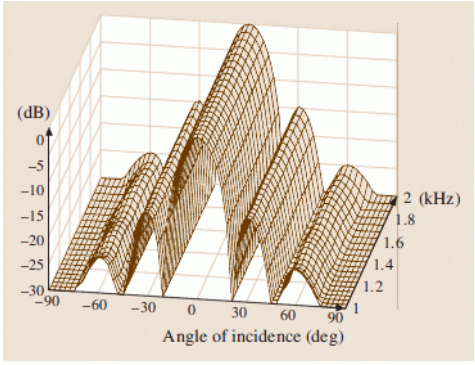


Figure 17 Nested microphone array frequency response (with kind permission of [29])

Please note that an array with an input frequency range of 500 Hz to 8 kHz as defined here is over 1m wide. This is quite impractical for a (portable) microphone array using in a meeting room. A full summary of constant-beamwidth microphone arrays can be found in Elko and Meyer [29].

Automatic Speech Recognition and Distant Speech Recognition

Hidden Markov models (HMMs) and their related technologies are at the core of almost all present-day automatic speech recognition (ASR) systems (Young 2009 [77]). Over the last two decades HMMs and their usage have been continuously refined and systems today use different kinds of feature enhancement and adaption techniques.

The usage of HMMs in speech processing originated in the 1980s [59]. Although the basic theory of Markov chains had been known for over 80 years, three problems and their corresponding solutions changed the world of speech recognition. These three problems and their solutions were:

- how do you calculate the probability of an observation sequence?
  → use a forward-backward algorithm
- how do you find the best path through an HMM?
  → use the Viterbi algorithm

- how do you maximise the HMM parameters?

  → use the Baum-Welch expectation-maximisation algorithm

Note that the Baum-Welch algorithm is a forward-backward algorithm used for calculating and maximising the HMM parameters, i.e. it can be applied to two of the three problems described above (see Rabiner and Juang for details [59]).

The invention of the Baum-Welch and Viterbi algorithms led to enormous progress in speech processing. In the phase that followed, research groups devised many detailed and sophisticated modelling techniques optimising feature extraction, phone modelling or state-tying techniques [77], though all with acoustic models at the core of the HMMs. Other methods such as the token passing algorithm or pruning were engaged in the Viterbi algorithm to optimise the HMM search. In the final step of decoding, a language model is engaged to further prune the decoding and to reoptimised the word lattice output of the Viterbi algorithm [77].

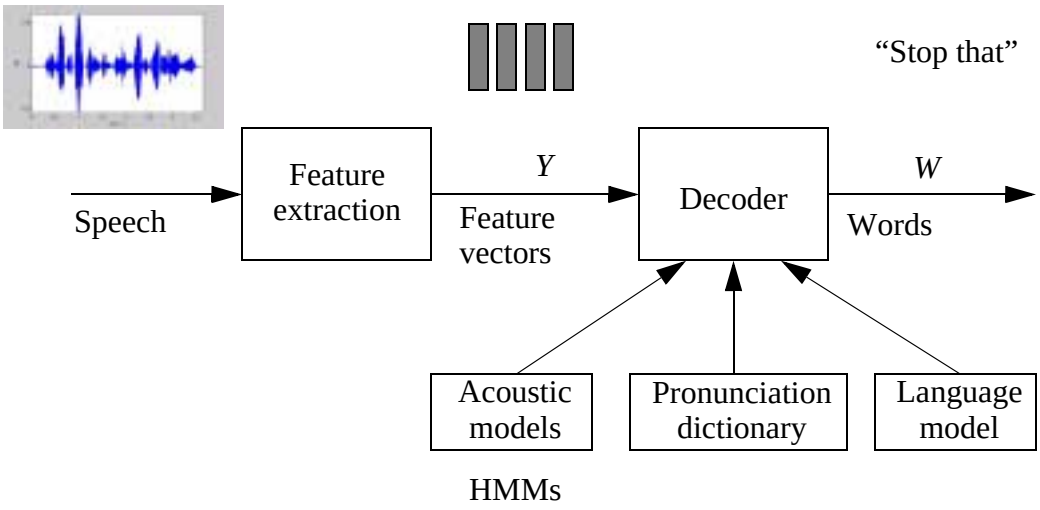


Figure 18 Architecture of HMM-based recogniser

Another fundamental problem of HMMs is that they represent the training data, i.e. not the test data. A speech recognition system will therefore not perform well if a new speaker needs to be recognised. The solution to this problem is HMM adaptation. Different adaptation techniques such as maximum likelihood linear regression (MLLR) or maximum a posteriori (MAP) adaptation algorithms have been developed [77]. MAP adaptation is a method of reinterpolating the original prior HMM parameters using supervised training sentences. All HMM parameters are updated by MAP and the major drawback is therefore sparsity of data. An alternative

adaptation approach is maximum-likelihood linear regression (MLLR) which is very successful with little or no (i.e. unsupervised adaptation) training data.

Adaptation

Both MAP and MLLR adaption are based on modifying the parameters of the HMMs, i.e. the observation probabilities μ and σ of the Gaussian distributions, as defined in Eq. (7).

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)} \quad \text{Eq. (7)}$$

Applied to multiple Gaussian mixture models (GMM) the observation probability P(Y) is a function of the acoustic input vector sequence Y and the mean and variance matrices μ and Σ of the GMMs, as shown in Eq. (8).

$$P(Y) = f(Y;(\mu, \Sigma)) \quad \text{Eq. (8)}$$

For MAP adaptation the individual μ and Σ of each GMM are reestimated. MAP is especially useful for porting well-trained models to a new domain [77]. If adaptation data is sparse then many model parameters will not be adapted. Several alternative approaches have been developed to overcome the above limitation of MAP adaptation.
One such method is MLLR adaptation, an incremental adaptation algorithm. Two variants of MLLR are of interest for this dissertation. Means-only and constrained MLLR adaptation. In both schemes transformation matrices are shared for a set of GMMs. Depending on the amount of adaptation data available, more or fewer transformation matrices are shared between the GMMs. For means-only adaptation, as the name implies, only the means are modified, as defined in Eq. (9).

$$\hat{\pi}_{jm} = G\mu_{jm} + b \quad \text{Eq. (9)}$$

Constrained MLLR is an adaptation technique that modifies the means and variances of the GMMs (see Eq. (10)), but constrained in such a way that the same matrix G is used for the means μ and the variances Σ, i.e. G = H.

$$\hat{\Sigma}_{jm} = H\Sigma_{jm}H^{T} \qquad \text{Eq. (10)}$$

Distant Speech Recognition

Distant Speech Recognition (DSR) combines the fields of acoustic array processing and automatic speech recognition (ASR). A complete DSR system includes the following distinct components [49]:

- a microphone array
- an algorithm to track the active speaker(s)
- a beamforming algorithm to focus on the desired speaker(s)
- post-filtering to enhance the beamforming
- a speech recognition engine and
- a speaker adaptation component

DSR therefore combines acoustic array processing and automatic speech recognition and will be looked at in detail in what follows (see also McDonough and Wölfel [48] [49]).

First, robust speech recognition is reviewed. Robust speech recognition is the discipline of ASR in adverse environments. The problem of speech recognition in adverse environments is basically noise such as environmental noise (e.g. a fan in an office or wind noise in cars) or reverberation. Other effects such as stress compensation (if the speaker changes his voice due to noise) or speech distortion (due to masking effects) also degrade the performance of an ASR system (Juang 1991 [40]). Acoustic ambient noise is usually considered additive to a system, i.e. what is received by the microphone is the sum of the speech and the noise. Speech enhancement therefore works on the principle of removing the noise from the speech so that the ASR system processes clean speech, where WERs of 5% or less can been achieved.
Early microphone arrays in ASR systems have already produced significant improvements in SNR performance, leading to a decreased WER (van Compernolle et al. 1990 [25]). Following van Compernolle et al. many researchers have attempted to improve the different components of a DSR system. Examples of this are:

- Stern et al. (1992) [69] reviewed the effect of multiple approaches to robust speech recognition, looking at acoustical pre-processing, microphone arrays and speech vector

enhancement techniques on the MFCCs or LPCs and demonstrated the combined improvement of these techniques on system performance

- Giuliani et al. [33] (1996) reproduced these results and also showed that HMM adaptation techniques have an additive effect
- Yamada et.al. [74] (1996) used pitch harmonics to locate the speaker in the room. This technique allows better steering of the microphone array beam and gives improved performance
- Kiyohara et al. [41] (1997) successfully demonstrated the positive effect of two-dimensional arrays on speech enhancement
- Seltzer et al. [65] (2002) combined the speech pre-processing, i.e. feature enhancement optimisation and HMM adaptation optimisation, using a maximisation-estimation (ME) algorithm which led to WER improvements
- Seltzer and Raj [66] (2003) later improved the system performance further by using speaker specific known utterances to adapt the enhancement filters and ASR system

These research examples focus on the robustness, i.e. the noise sensitivity of the DSR system. Another well researched problem of DSR systems is overlapping speech. Shriberg et al. [67] (2001) show that overlapping speech is not a distortion as such but an important inherent characteristic of speech. Overlapping itself contains important information and should be jointly modelled in speech processing. Microphone arrays can be used to separate overlapping speakers and therefore improve a meeting DSR system (Moore and McCowan [51] 2003, Lincoln et al. [43] 2005 and McDonough et al. [50] 2006).

A review of robust speech recognition, i.e. speech recognition systems which cope in a practical environment such as enclosures (i.e. meeting rooms) is provided in Renals et al. [60] (2007). Schuller et al. [64] (2009) give a current extensive survey and analysis of feature enhancement, model adaptation techniques and speech signal pre- and post-processing. The authors are working on DSR techniques to improve speech recognition inside cars.

DSR is a very complex problem associated with the recording of meetings. The wealth of information exchange in meetings is often lost because note taking is subjective and incomplete. The AMI/AMIDA ([35], [47], [60]) and ICSI projects ([52], [68]) aim to improve the efficiency of meetings and decrease their number. Multimodal recording of meetings is one solution to this

which enables the recognition, structuring, indexing and summarising of meetings content. The main aims of the AMI/AMIDA and ICSI projects are:

- ASR
- ASR from far-field microphones, e.g. microphone arrays
- speaker segmentation and turn detection
- speaker identification and prosody recognition (e.g. laughter)
- dialogue abstraction, e.g. identifying rules and creating a speaker timetable
- dialogue analysis
- summarisation

Numeral instrumented meeting rooms (IMR) were built for these projects and both projects have successfully recorded and annotated 75 hours (ICSI) and 170 hours (AMI/AMIDA) of meetings. Both databases are available publicly (http://www.idiap.ch/mmm) and can be used for further development.

Distant Speech Recognition problems apply to both conference room meetings and lecture room meetings. The ISL (Interactive System Laboratories, see Fügen et al. [32] 2006 and Wölfel et al. [72] 2006) demonstrate DSR in a lecturing environment. A toolkit and evaluation lecture recognition system has been developed and successfully tested using head microphones and single and multiple distant microphones.

Distant speech recognition is a new and exciting research area and “looking into the black boxes” of the individual DSR disciplines, i.e. bridging the gap between acoustic array processing and automatic speech recognition will bring progress for the IMR as McDonough and Wölfel ([48] [49] 2008) successfully demonstrated. The authors modified existing DSR components by enabling access to the internal states of e.g. the beamforming and post-filtering processes to the speech recogniser, i.e. the Viterbi search.

DMA - Background

The idea of the digital microphone array (DMA) originated with the latest analogue array available on the market which was designed specifically for IMR [107].When looking at the size, not of the microphone array itself, but the amplifier following the microphone array [119], the question arose as to why this microphone is not implemented with the latest available digital MEMS microphones and more economical digital signal processing. Using MEMS microphones with a digital PDM (pulse density modulation) output, some signal processing and the correct interface, a digital microphone array can be implemented using no more than the microphones, a DSP chip, an Interface chip and some passive components.
Microphone arrays have been in use for speech capture in meetings for speech recognition for over a decade. The main benefits of arrays are reduced sensitivity to noise and reverberation and the ability to capture multiple speakers without the need for head-mounted microphones. Speech recognition, speaker recognition and other speech processing techniques have become possible with digital signal processing and have undergone a tremendous boost with the immense increase in desktop computing power. The first basic problem therefore is the conversion of the speech signal from the analogue real world into its digital representation. The following section explains principles of analogue to digital conversion, signal processing and interfaces.

Analogue Digital Conversion

The conversion of the analogue signal into a digital value is a well known and intensively researched area. The principal block diagram of an ADC is shown in Figure 19 below.

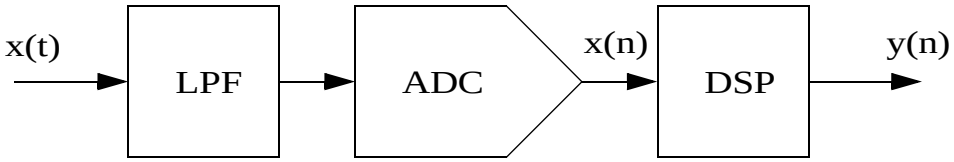


Figure 19 Analogue Digital Conversion (ADC)

The three major building blocks of an ADC are the low-pass filter (to remove all undesired frequency components above the Nyquist frequency), the actual analogue to digital converter (ADC) and some digital signal processing (DSP). Almost a dozen ADC principles are known (see [99] for a summary). The two main ADC (Analogue to Digital Converter) types are:

- Nyquist converter
- Oversampling converters

The two main problems that need to be handled properly when converting a time domain analogue signal into the discrete digital signal are aliasing and quantisation. Oversampling SDM (sigma delta/ΔΣ modulation) ADCs have been successful ever since they were invented because of their ability to handle aliasing and quantisation errors. The reason for this is best explained by comparing Nyquist converters with oversampling SDM ADCs.

Definitions

Three definitions are necessary before going into ADC principles in greater detail.

- The sample frequency f_s is the frequency at which the ADC produces digital values sampled from the analogue signal.
- Aliasing is the problem of higher frequencies folding into lower (audible) frequencies due to the sampling ADC not being able to distinguish between a signal that is e.g. 0.45 f_s compared to 1.45 f_s.
- Quantisation, i.e. quantisation errors, are errors introduced by the finite step size of a n-bit ADC compared to the infinite sensitivity of a real analogue signal.

Both problems are explained in details in what follows.

Nyquist and Oversampling ADC

A Nyquist converter is an ADC that samples the analogue signal at the sample frequency f_s, or double the Nyquist frequency[1] f_n ($f_n = f_s/2$). An oversampling SDM ADC, on the other hand, samples the analogue signals at a much higher multiple of the sample frequency, e.g. $F_s = 64\ f_s$. The first principal problem of analogue to digital convertion, the problem of aliasing, is illustrated in Figure 20 below. If, for example, a 16-bit Nyquist converter were to be implemented, the required analogue low-pass filter (LPF) would need to have a brick-wall cut-off frequency response at the Nyquist frequency from 0dB to 96dB[2]. Such a filter cannot be implemented in practise (see [57] for details).

1. The Nyquist frequency is also know as the folding or cut-off frequency of a sampling system.
2. A 1-bit ADC would produce a 1 for a signal > 0.5[u], and 0 for a signal < 0.5[u], therefore giving 6dB of input signal resolution. 10 bits therefore give 60dB, and 16 bits would give 96dB resolution, or 2^{16}, i.e. 65'536 steps, or, on a 1Vrms signal, 15μV step size (see [76]).

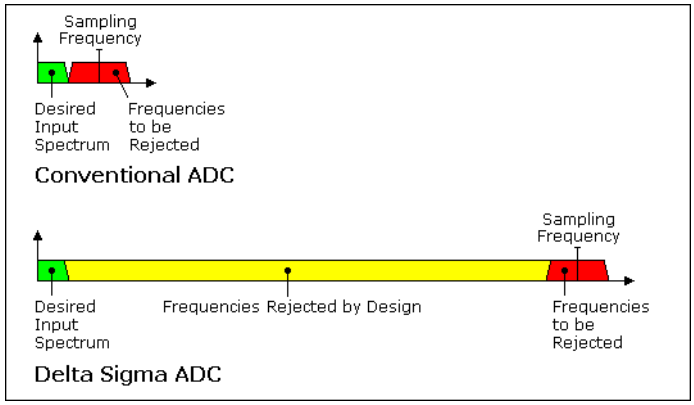
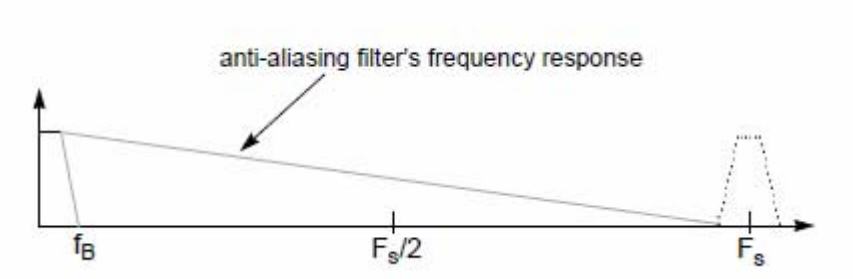


Figure 20 Aliasing and frequency bands (with kind permission of [22])

Using an oversampling SDM ADC the requirements for the analogue low-pass filter (that is placed before the converter) can be greatly reduced, as shown in Figure 21 below.



Note: f_b is used in this figure for the Nyquist frequency f_n

Figure 21 Analogue anti-aliasing filter requirements (from [57])

Instead of a high-quality analogue brick-wall filter, the anti-aliasing filter that is necessary can be a simple first-order RC filter. This is feasible, cheap and easily implemented in a standard microchip using CMOS technology.

Oversampling or Sigma-Delta Modulation Converters

While Nyquist Converters are the only feasible solution for high-speed real-time FE Converters (e.g. Telecom applications) they are usually limited to resolutions of a few bits. For HiFi applications the Sigma-Delta Modulation (SDM) Analogue Digital Converter (ADC) is a prime choice. It is used in most modern electronic components [106] due to its cost effective implementation. The principle top-level block diagram of an SDM ADC is presented in Figure 22.

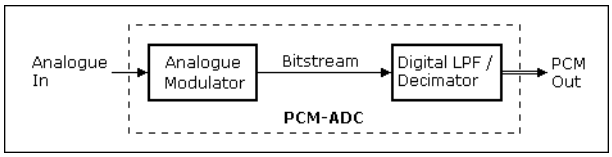


Figure 22 SDM ADC top-level block diagram (with kind permission of [22])

Using a n^{th} order sigma-delta modulator (SDM) an analogue signal is converted into a PDM bitstream at the oversampling rate, e.g. $F_s = 64\ f_s$. This pulse density modulated (PDM) bitstream is then down-sampled using decimators and digital lowpass filters into a pulse code modulated (PCM, see [129] for details) signal at f_s.

The principle of sigma-delta (or delta-sigma) modulation is shown in Figure 23 and explained in what follows below.

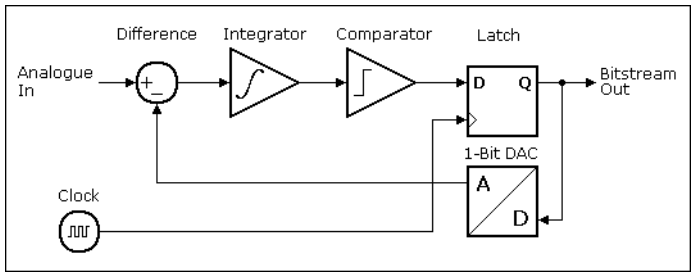


Figure 23 SDM block diagram (with kind permission of [22])

SDM ADC handle the aliasing and quantisation problem in one step. Aliasing is addressed by sampling at a much higher rate than the Nyquist rate f_n and quantisation errors are minimised by dividing the analogue input signal into only two levels, zero ‘0’ and one ‘1’. This therefore requires only a simple one-bit ADC converter.

The SDM is built of a one-bit DAC, one bit of memory, an integrator and a comparator. The output of the SDM is determined by comparing the integrated difference signal of the input signal with the previous output. If the signal is above the threshold (the middle of Vref+ and Vref-, i.e. 0V in this example), then the output is 1, if it is below then the new output is 0. The SDM keeps track of the analogue input signal by executing this comparison many hundred thousand times a second, as demonstrated in Figure 24 below.

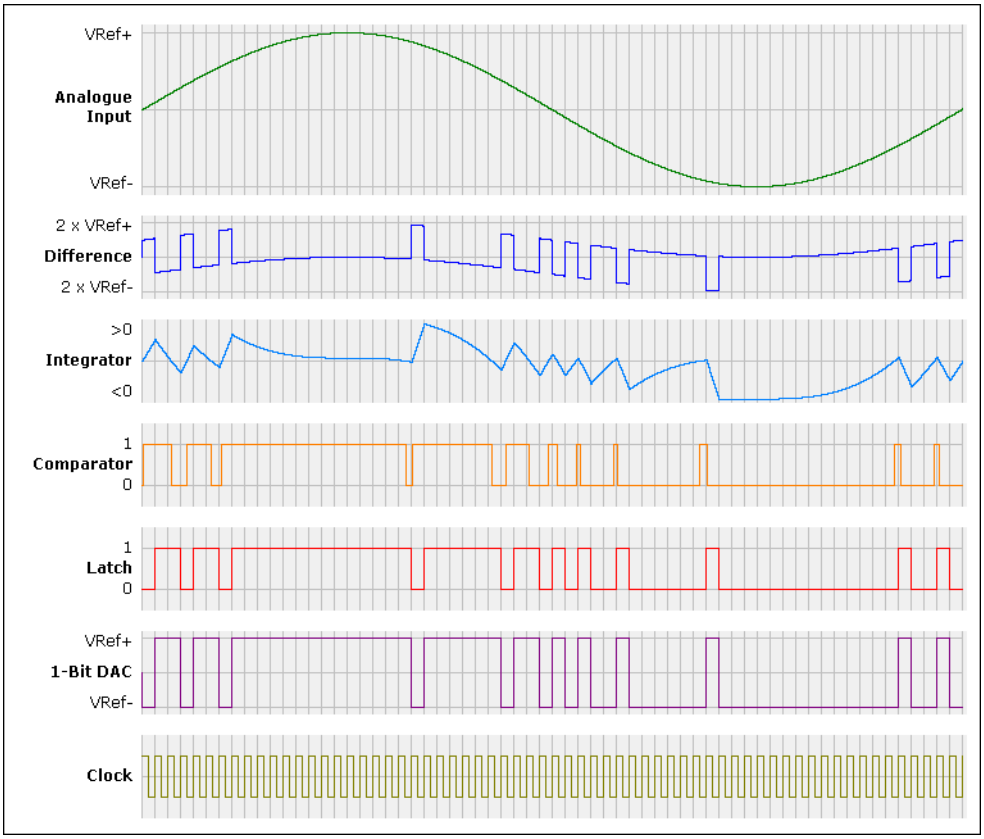


Figure 24 SDM principle (with kind permission of [22])

The error signal, i.e. the resulting quantisation error in the SDM ADC demonstrated above, is shown in Figure 25 below.

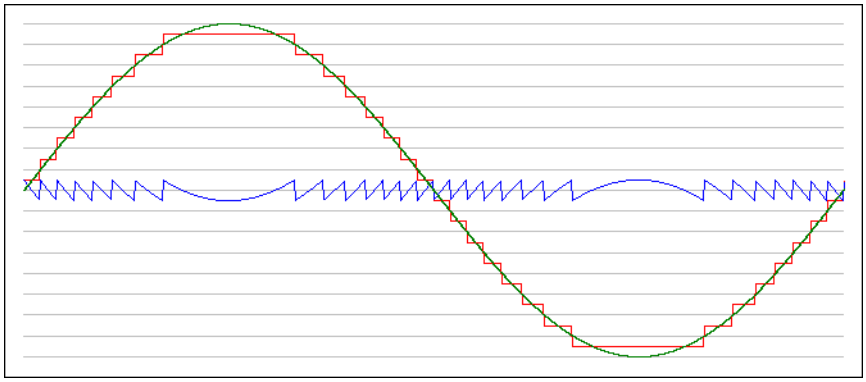


Figure 25 Sampling/Quantisation error (from [22])

The sigma delta modulator is designed in such a way that the quantisation error is significantly less than half an LSB of the PCM output of the ADC.

Advanced schemes are added to the basic SDM which deal with the many problems that the SDM has. The one significant advantage of the SDM is its noise shaping characteristic. Using

higher-order SDMs, performance of an SDM ADC can be tuned to the maximum. For a detailed introduction to SDM ADCs see [57], [70] and [123].

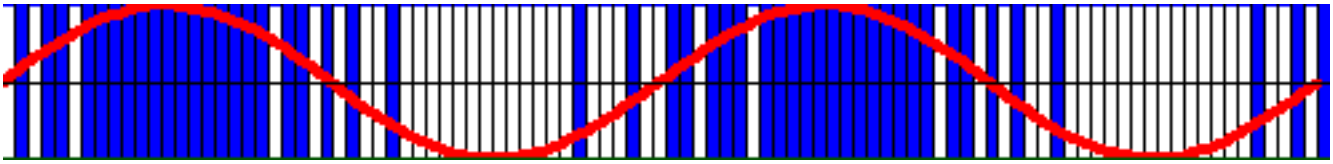
Interfaces

The following section provides an introduction to the interfaces between the major building blocks of the digital microphone array:

- PDM interface, connecting the MEMS microphone with the DSP
- AC'97 interface, connecting the DSP with the USB streaming controller
- USB interface, connecting the DSP with the PC

PDM interface

The output of the digital MEMS microphone is a digital bit-stream at 64 f_s. The digital MEMS microphone contains an analogue digital converter that converts the air density modulation first into an analogue electrical signal and then into a digital discrete signal using an SDM ADC. The output of the SDM ADC is a pulse density modulated (PDM) signal, as shown in Figure 26.



0101101111111111111101101010010000000000000100010011011101111111111111011010100100

Figure 26 Pulse Density Modulation (from [121], Wikipedia "fair rules" apply [130])

Figure 26 shows that a rising analogue input signal leads to a larger (delayed) number of 1s in the output bit stream, while lowering the inputs signal results in a series of 0s. Looking at the above signal a trained eye can recognise that a simple first order RC lowpass filter connected to a PDM stream is enough to regenerate the analogue signal. The conclusion from this is that a HiFi DSP that converts the PDM MEMS microphone output into a PCM signal is simply a high quality lowpass filter (LPF).

AC'97 interface

While a PDM output for a digital MEMS microphone is a de facto industry standard, choosing a way to transfer eight channels of PCM data is not straight forward. The best-known interfaces are:

- I^2S: Inter-IC Sound, is a 4-wire (clock, left/right signal, input and output data) interface, allowing many data-only channels to be transmitted using a TDM scheme [88]
- AC'97: Audio Codec 97, is a 5-wire (clock, synchronisation, input and output data and reset) interface, using 13 TDM channels. One house-keeping, two control and ten data channels are available [85] as shown in Figure 27 below

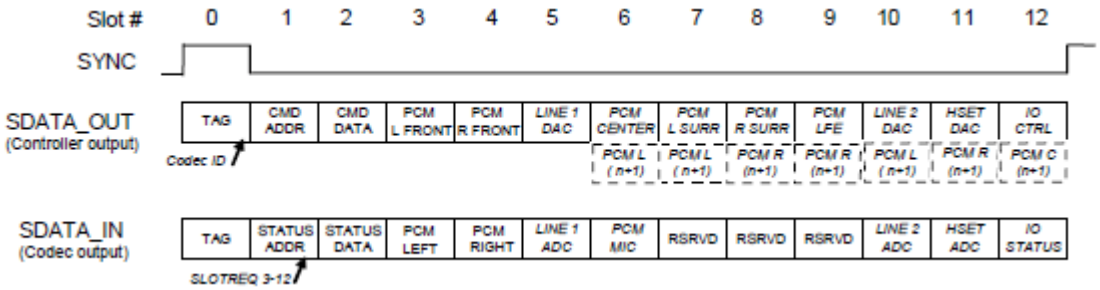


Figure 27 AC'97 TDM scheme (from [85])

- S/PDIF: Sony/Philips Digital Interconnect Format, is a one wire (signal and shield/ground) serial interface combining control and 4 data channels [90]

Recently two more interfaces have emerged, which are:

- HDA: High Definition Audio, an expansion of AC'97 [87]
- SLIMbus: Serial Low-power Inter-chip Media Bus, a standard interface between processors and peripheral components such as microphones and loudspeakers [89]

Getting standard components and support for the HDA and SLIMbus interfaces was considered to involve too many risks for the implementation of the digital microphone array and in the end the AC'97 interface was been chosen because of the simple access to components and support.[1] I^2S would be an alternative interface but it was not chosen because it does not allow the transfer of control data.

USB (Universal Serial Bus)

While the HDA interface would allow bypassing an interface chip between the DSP and the PC it would require the presence of an HDA interface on the PC. This is not usually the case and only the latest IBM motherboards give the user access to the HDA interface which directly plugs into the Southbridge device [44]. Such a scheme is only expected to become available later this year or next.

1. It should be noted that I designed an AC'97 codec for Wolfson Microelectronics a few years ago which allowed me to design the interface in a very short time.

Because there is no direct link between the signal processing and the PC, an interface device needs to be added. The options are:

- USB: Universal Serial Bus, a two-wire packet based interface using half-duplex differential signalling which offers control and data communication, e.g. guaranteed isochronous audio channels [92]
- IEEE 1394: Firewire interface, a four wire packet based interface using half-duplex signalling which offers data and control communication as well as guaranteed isochronous audio transfer very similar to the USB bus [86] [116]

The Universal Serial Bus is believed to be easier to use for the digital microphone array partly because of easy access to the TAS1020 and TUSB3200A AC'97/I^2S to USB streaming controllers. Modern PCs contain several USB interfaces. In addition, the wide availability of existing FW drivers should aid the development of the digital microphone array.

DMA - Building

The following section describes the building of the digital microphone array (DMA). First, the specification and design of the digital microphone array is shown in a top-down manner. The blocks components and interfaces are presented starting with the system definition. The implementation of the array is followed by the Software (SW), or Firmware (FW) requirements and designs.

System Design

The main components of the digital microphone array are:

- Microphone array
- Digital Signal Processing
- Interfacing from the microphones to the DSP and from the DSP to the PC
- Application for capturing raw audio data
- Post-signal processing (e.g. noise reduction, beamforming, etc.)

and

- Automatic Speech Recognition

The timing constraints for this dissertation, i.e. the limited time and resources for the design, implementation and verification of the digital microphone array put significant restrictions on the system design. The aim was therefore to find the best-fitting components available on the market to build the array.

One available analogue array [107] uses standard industrial ADCs and DACs and a general purpose multi-core DSP processor with block transfer capabilities into a Firewire interface (IEEE 1394). Reproducing such a scheme would require the implementation of significant amounts of Firmware (FW) and Software (SW) and is not feasible for this project. Alternatively, USB streaming devices should allow a simpler system implementation.

The following components were selected for the implementation of the digital microphone array after a two-week feasibility study and market analysis:

- TUSB3200A USB streamer [126]
- Xilinx Spartan 3A FPGA [131], [132]

Both devices are available with an evaluation board. The TI USB streamer comes with example designs and Firmware ([128]) while the Xilinx Spartan 3A (XC3S700A-FG484) FPGA is a reasonably sized programmable chip large enough for the digital signal processing necessary

but which can still be simulated and programmed with freely available Software. Choosing these devices also solves the second biggest problem for the system implementation, i.e. the choice of the interfaces. These interfaces are:

- AC'97 interface [85] from DSP to PC IF device
- USB interface [92] from digital microphone array to recording SW (on PC)

The system resulting from these choices is illustrated in Figure 28 below.

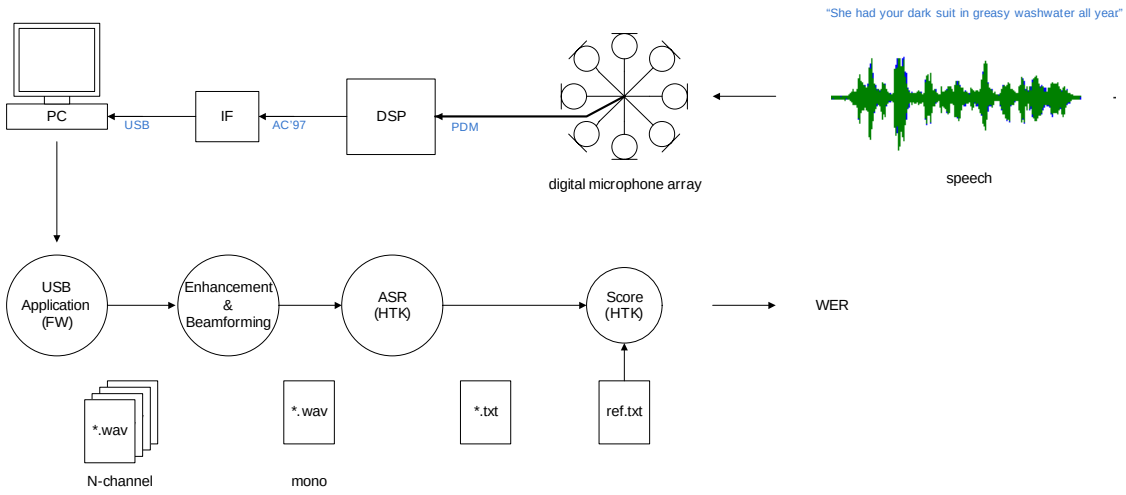


Figure 28 System

The individual components and building blocks of the digital microphone are presented in what follows.

Signal Processing

Using a Nyquist ADC the output of the converter is a PCM signal at f_s, i.e. a 2's complement binary signal of typically 16 or 24 bit width. This converter does not need any signal processing. The output of an oversampling ADC is a 64 f_s PDM signal. Although a loudspeaker could very easily be supplied with this signal (the speaker acts as a low pass filter, as it is used in hearing aids [24]), the data rate and therefore the amount of data to transfer in such a system is not feasible. Therefore signal processing is necessary to convert the 64 f_s PDM signal into a PCM signal at f_s.

The signal processing required for this conversion can be described as a low pass filter (LPF), the same LPF required as in the Nyquist converter. Fortunately, in contrast to the Nyquist converter, this filter is placed after the ADC and can therefore be implemented in the digital domain. While it would be feasible to build a single filter removing all the spectrum from f_n to

F_s, such a filter would be very costly in terms of processing power and size. It would need to run at F_s, and be an FIR filter of approx. 3400^{th} order.

Over the decades engineers have come up with clever schemes to implement the down- (and up) conversion efficiently, as shown in Figure 29 below.

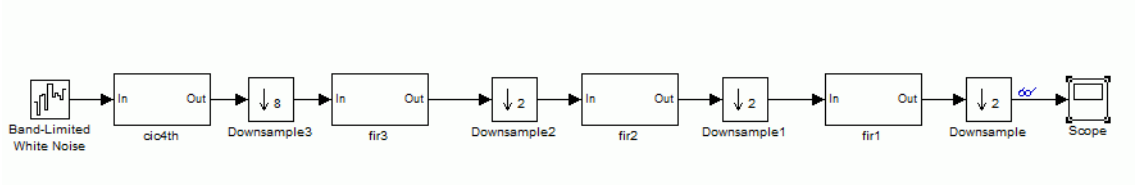


Figure 29 DSP

The basic trick is to down-convert in stages, i.e. from e.g. 64 f_s to 8 f_s to $4f_s$, 2 f_s and finally f_s. The following Figure 30 illustrates this.

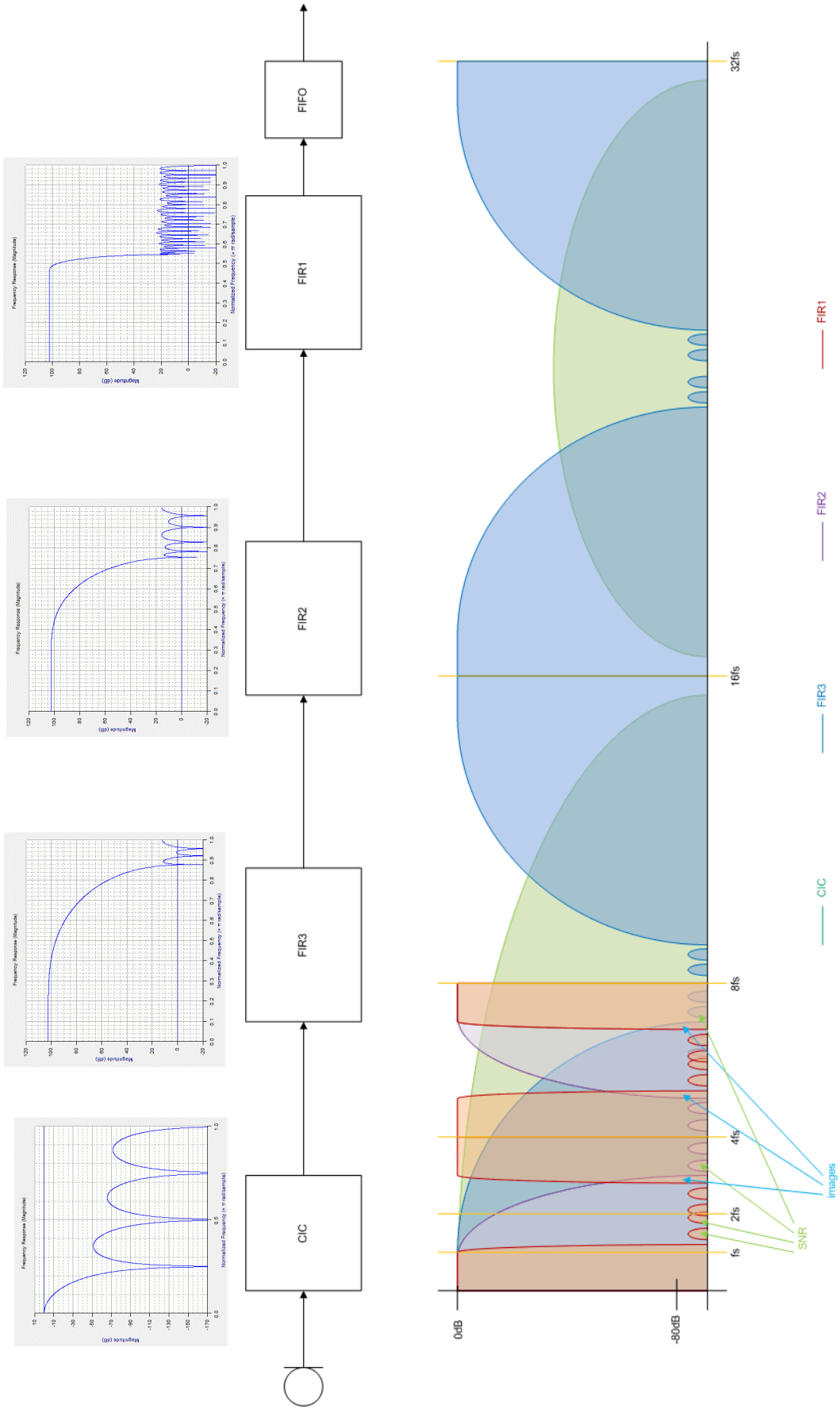


Figure 30 System Filter Response

Figure 30 above presents a very efficient method to downsample an over-sampled signal. The two basic building blocks are:

- CIC (Cascaded-Integrator-Comb) filter
- Halfband FIR filter.

The principal function, features and advantages of these filters are presented in detail in what follows. Please note the SNR and images shown in Figure 30. Both the SNR specification of a digital filter and the effect of the images will also be explained in detail in this section.

CIC (Cascaded-Integrator-Comb) filter

Hogenauer [38] found CIC filters, an "economical class of digital filters for decimation and integration", in the very early 1980s. Decimation is the process of down-sampling a digital signal from e.g. 64 f_s to f_s, while interpolation is the inverse, i.e. upsampling from f_s to e.g. 64 f_s. Only decimation is of interest for the purpose of this dissertation (and the digital microphone array).

CIC filters can be used for ultra economical decimation; economical in the sense that they are small and fast because they do not require multipliers but are built with add/sum and delay elements only. The principal building blocks of CIC filters are integrators and decimators. These filters have the advantage that they are digital filters with all their coefficients, i.e. poles and zeros, on the unit circle, therefore only requiring summation operations. The basic design of a CIC filter is presented in Figure 31 below.

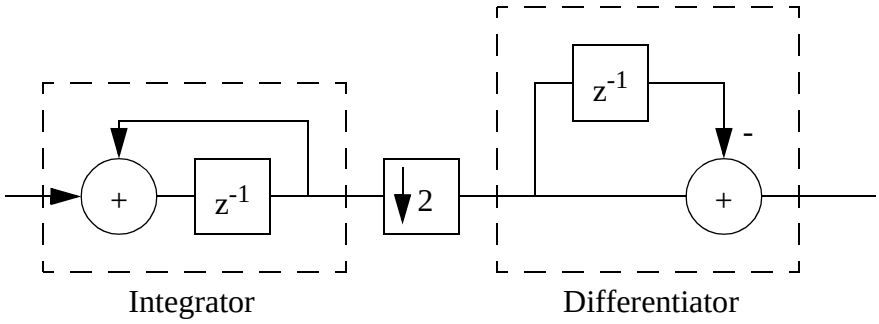


Figure 31 CIC block diagram

First the input signal is integrated using one (or more) integrators, then downsampled by n (with n being a power of 2, i.e. $2^1 = 2$), and then differentiated using the same number of differentiators as integrators. This simple scheme results in a filter response as show in Figure 32 and makes them a perfect fit for signal decimation.

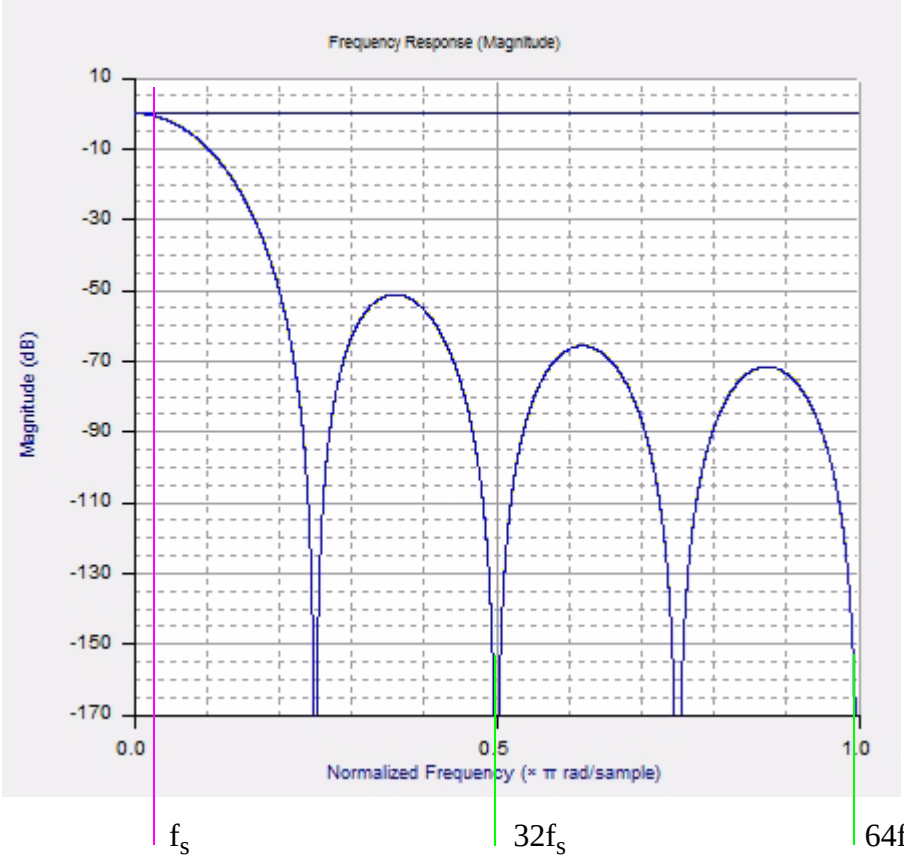


Figure 32 CIC Filter

CIC filters are very common, so common that Xilinx allow you to integrate most CIC decimators and interpolators using their CORE Generator tool. If the system designer knows the specification for the CIC filter then the CIC filter, the FPGA implementation and a model can be generated with very few operations. More information can be found in [18], [38] and [108].

(Half-Band) FIR filter

Although CIC filters are very efficient filters used for down-sampling signals, they are only efficient to a certain degree, specifically when used to downsample higher rate signals. For down-sampling lower rate signals Finite Impulse Response (FIR) filters, as defined in Eq. (11), are needed.

$$y_n = \sum_{m=0}^{M-1} b_m \cdot x_{n-m} \qquad \text{Eq. (11)}$$

Looking at Eq. (11) closer, considering a two-tap (m = 2) FIR filter with $b_m = b_1 = b_2 = 0.5$ and therefore $y = b_1 x_n + b_2 x_{n-1} = x_n/2 + x_{n-1}/2$, a new output sample y_n is calculated adding half of

the current input signal and half of the previous one. This is simply a moving average filter, which is the most basic high-pass filter. A more complex filter implementation is presented in Figure 33 below.

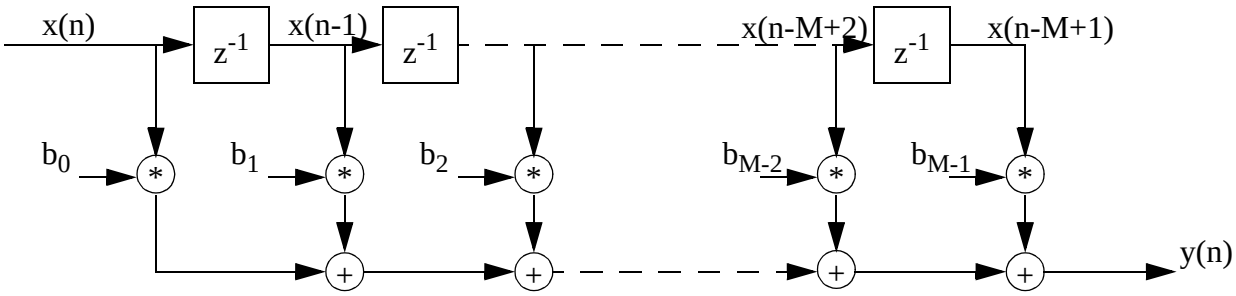


Figure 33 Direct form FIR filter

Half-band filters are filters that let half of the frequency band of the input signal pass. In the case of a decimation LPF, if the filter runs at 8 f_s, any signal above 2 f_s is discarded[1]. On close inspection half-band filters show two very special characteristics. The first is that the coefficients b_m are symmetrical (i.e. $b_0 = b_{M-1}$, $b_2 = b_{M-3}$, etc.) and the second feature is that every second coefficient is zero (i.e. $b_1 = b_{M-2} = b_3 = b_{M-4} = 0$). Using this special feature of the half-band filters, a filter such as FIR3 (as shown in Figure 34) with 11 coefficients b_m only requires 4 multiplications using a pre-add-multiply-accumulate structure or 6 multiplications using a multiply-accumulate structure.

1. With the Nyquist frequency $f_n = f_s / 2$, a filter running at f_s processes signals of the input spectrum from 0 to $f_s / 2$, a half-band filter therefore filters everything above or below $f = f_b / 2 = f_s / 4$.

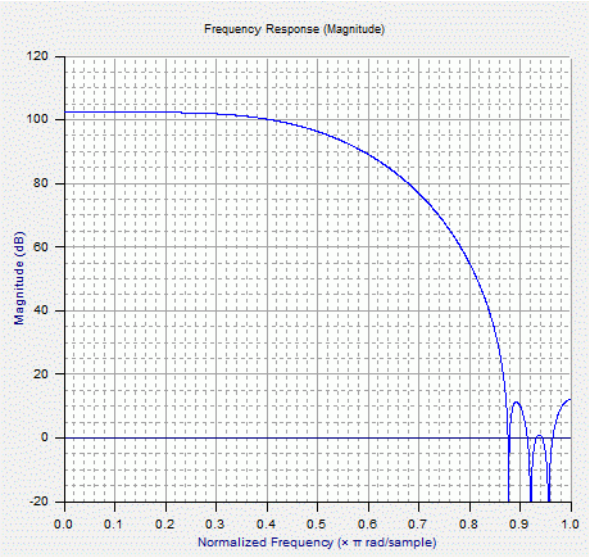


Figure 34 FIR3

As shown in Figure 30, three half-band filters FIR3, FIR2 and FIR1 are used to downsample from 8 f_s to $4f_s$, $2f_s$ and f_s. Down-sampling in steps of 2 using half-band filters has been shown to be the most efficient way of implementing decimation filters (and vice versa for interpolation filters).

Two major building blocks are used for the implementation of an FIR filter. These are:

- (pre-add-)multiply-accumulate unit
- storage register

The FIR filters FIR3, FIR2 and FIR1 have been designed using the Matlab Filter Design and Analysis tool fdatool which is part of the filter design toolbox. Using the fdatool the filter can be specified, or, as in the case of the microphone array, pre-calculated filter coefficients can be imported[1]. After defining specific parameters (such as fixed point data widths) using the Target generator, Xilinx Coefficient (.COE) Files can be generated and then imported into the Xilinx CORE Generator (see [19] for details). Implementation files and simulation models are generated with the Xilinx CORE Generator and can be used to verify and build the design.

The system as shown in Figure 30 is completed with FIFOs, first-in-first-out memories, which collect eight parallel samples. Once each FIFO contains at least one sample the AC'97 interface block announces the presence of data in the synchronisation slot (0) and the eight samples are sent to the interface chip. The most efficient way to implement memory for a Xilinx FPGA is

1. The filter coefficients have been kindly supplied by Anthony Magrath from Wolfson Microelectronics plc

again by using Xilinx CORE Generator, specifically the Block Memory Generator (i.e. Simple Dual-Port RAM [17]).
Two important system aspects, signal to noise ratio (SNR) and frequency images also need to be considered. Both are looked at in what follows.

Signal to Noise (SNR) ratio

The specified Signal to Noise ratio (SNR) is one parameter that defines the width and complexity of a DSP. As shown in [76], a signed digital signal of width n is able to represent an analogue signal with 2^{n-1} steps. 16 bits therefore allow 2^{15} steps, and 20 log 2^{15} is 90 dB[1]. The AC'97 interface allows a maximum data width of 20 bits [85], while USB works in segments of 8 bits, therefore giving the choice of either 16 or 24 bits [92]. When designing the system every component needs to be taken into account. The MEMS microphones used specify a typical SNR of 56dB [12], though at a sound pressure level of 1 dB (=1 dB_{SPL}). In reality the microphone can easily cope with up to 20dB_{SPL}, therefore allowing an SNR of around 80dB. This makes a system data width of 16 bits the obvious choice. In addition, Xilinx CORE Generator components allow only limited filter data and coefficient widths (i.e. 17 bits for the input data, 35 bits for the outputs, and 18 bits for the coefficients of the FIR filters and two bits for the CIC filter input and 18 bits for the output).
What is clear from the above description is that the maximum possible bit width has been chosen at the interfaces of each individual stage of the system while complying with the hard limits. In the end, the data arriving at the PC is 16-bit PCM data and the SNR of the digital microphone array system is limited by the MEMS microphone used and not by the signal processing along the way[2], the correct method for system design.
The noise introduced by the digital signal processing (as shown in Figure 30) is therefore below -85dB.

(Frequency) Images

Using down-sampling stages (compared to one single brick-wall filter) has the disadvantage of introducing frequency images, as indicated in Figure 30. Gaps (or spikes as shown with the arrows) are left in the frequency spectrum due to the non-perfect cut-off characteristic of e.g.

1. Using unsigned numbers 16 bits would allow 96dB SNR, while using signed 2's complement numbers 16 bits allow only 90dB SNR. This is just a mathematical effect. In the microphone array the output of the recording is 16-bit signed numbers.
2. See [75] for an introduction to practical considerations in fixed-point FIR filter implementations.

FIR1 running at 2 f_s and FIR2 running at $4f_s$. These gaps can be reduced to a minimum by careful design of the filter roll-off frequency thus ensuring that the overlapping areas (or the frequency responses) cross below the SNR that is necessary for the system.

DSP implementation

Several applications are needed to build the digital microphone array. These are:

- Matlab: tool to specify and generate signal processing components
- Mentor Modelsim ISE: Verilog HDL simulator to build the interfaces (e.g. AC'97) and assemble and simulate the digital microphone array DSP (pre- and post-synthesis and place and route)
- Xilinx ISE Foundation: Synthesis, translation, mapping and place and route of Verilog HDL
- Xilinx CORE Generator: DSP core generator

The modules necessary to build the digital microphone array are presented in Table 1 below.

Table 1: Digital Microphone Array components

Component[a]	Module name[b]	Description
Top	top.v	Top level module where everything is instantiated and wired together
AC97 interface	- ac97_if.v	AC'97 interface module, handling all AC'97 specific tasks such as reset, AC'97 clocking, slot 0 management, address and data control (slot 1 and 2) and audio data management (slots 3 to 10
DSP	- dsp.v	DSP top-level where 4 stereo DSPs and their FIFOs are instantiated and wired up
Clock manager	-- clk_mgr.v	DSP clock management
Synchronisation block	-- dsp_sync.v	DSP synchronisation, ensuring all stereo DSPs are working synchronously
DSP core	-- dsp_stereo_x.v	Individual DSP core, processing two channels simultaneously
CIC filter	--- xcic4th.v	Stereo 4th order CIC filter
FIR3 filter	--- xfir3_st	Stereo FIR3 filter
FIR2	--- xfir2_st	Stereo FIR2 filter
FIR1	--- xfir1_st	Stereo FIR1 filter

Table 1: Digital Microphone Array components

Component[a]	Module name[b]	Description
FIFO	-- fifo2.v[c]	FIFO memory, interfacing the DSPs and the AC'97 interface to ensure that eight parallel samples are sent via the AC'97 interface

a. Not every component is listed here - please refer to the project "work.mpf" generated in Modelsim and the documentation in the HDL code for a full summary

b. Hierarchy is represented by using '-' with the number of '-' implying the depth of the module

c. See [28] for details

After introducing and specifying the building blocks of the digital microphone array DSP, the next section describes how the individual components have to be correctly assembled and wired up. This is shown in Figure 35 below.

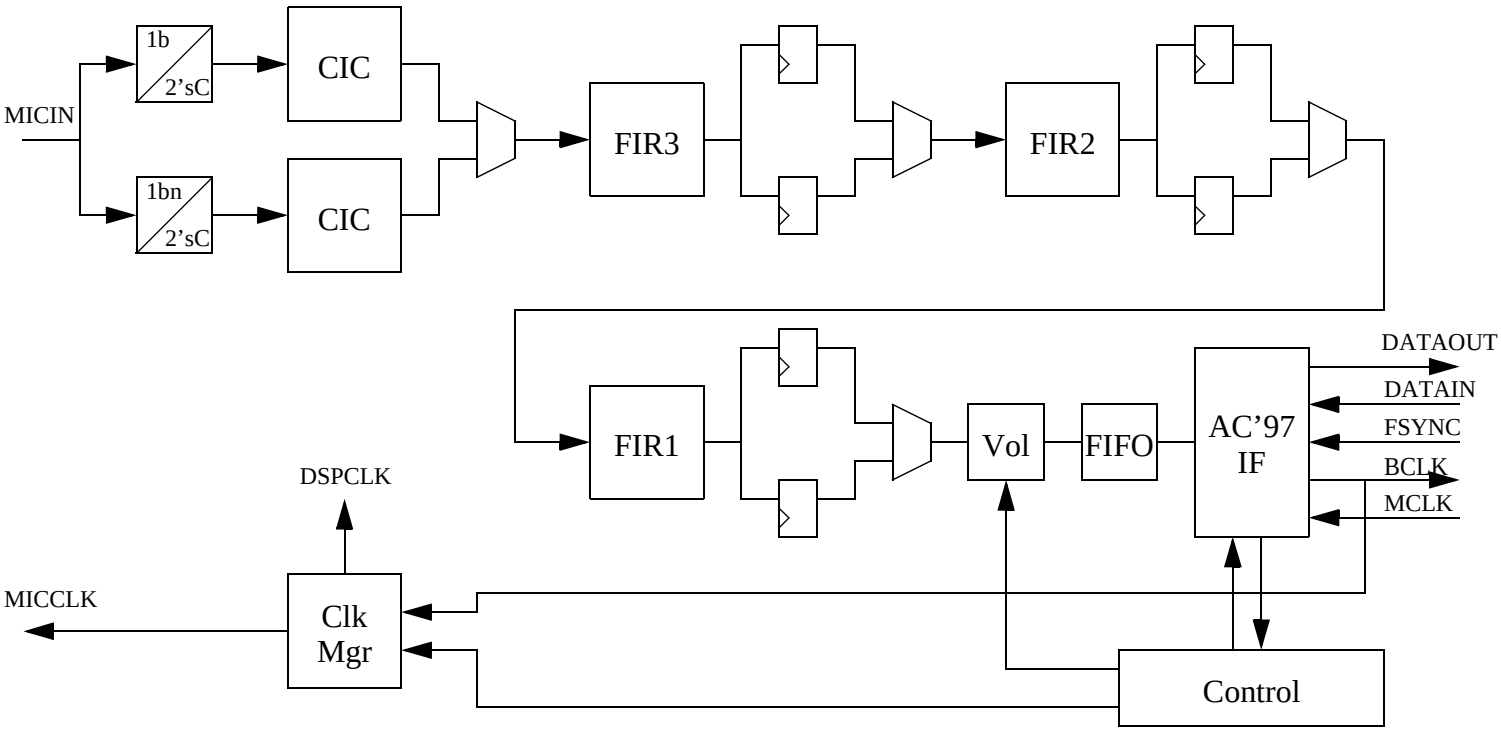


Figure 35 Digital Microphone Array DSP

The input to the digital microphone array from the MEMS microphone carries the signal of two microphones on one wire. First these signals need to be separated and converted from binary 0s and 1s to 2's complement numbers, i.e. $0 \rightarrow -1_d \rightarrow 11_b$ and $1 \rightarrow 1_d \rightarrow 01_b$. Then the two channels are processed separately with two CIC filters, i.e. downsampled from 64 f_s to 8 f_s. The FIR

filters were implemented as stereo filters due to only 20 multipliers being available on the Spartan FPGA. With eight microphone channels and three filter stages a full parallel implementation would require 24 multipliers. This is not feasible in this device. Multiplexers and demultiplexers stages have been implemented in between the FIR filters. These are simple depth-one memory elements necessary for re-timing and post-processing the FIR filter output. After the last filter stage, i.e. FIR1, a volume control block is added, followed by the FIFO memory connecting to the AC'97 interface. The AC'97 interface block implements the AC'97 protocol. Two additional components are the control block - which allows register read and write to control the volume and access test structures - and the clock controller - which generates the clocks for the DSP and the microphones.

The digital microphone array DSP is built as shown in Figure 36 below, using all the components and EDA tools as described above.

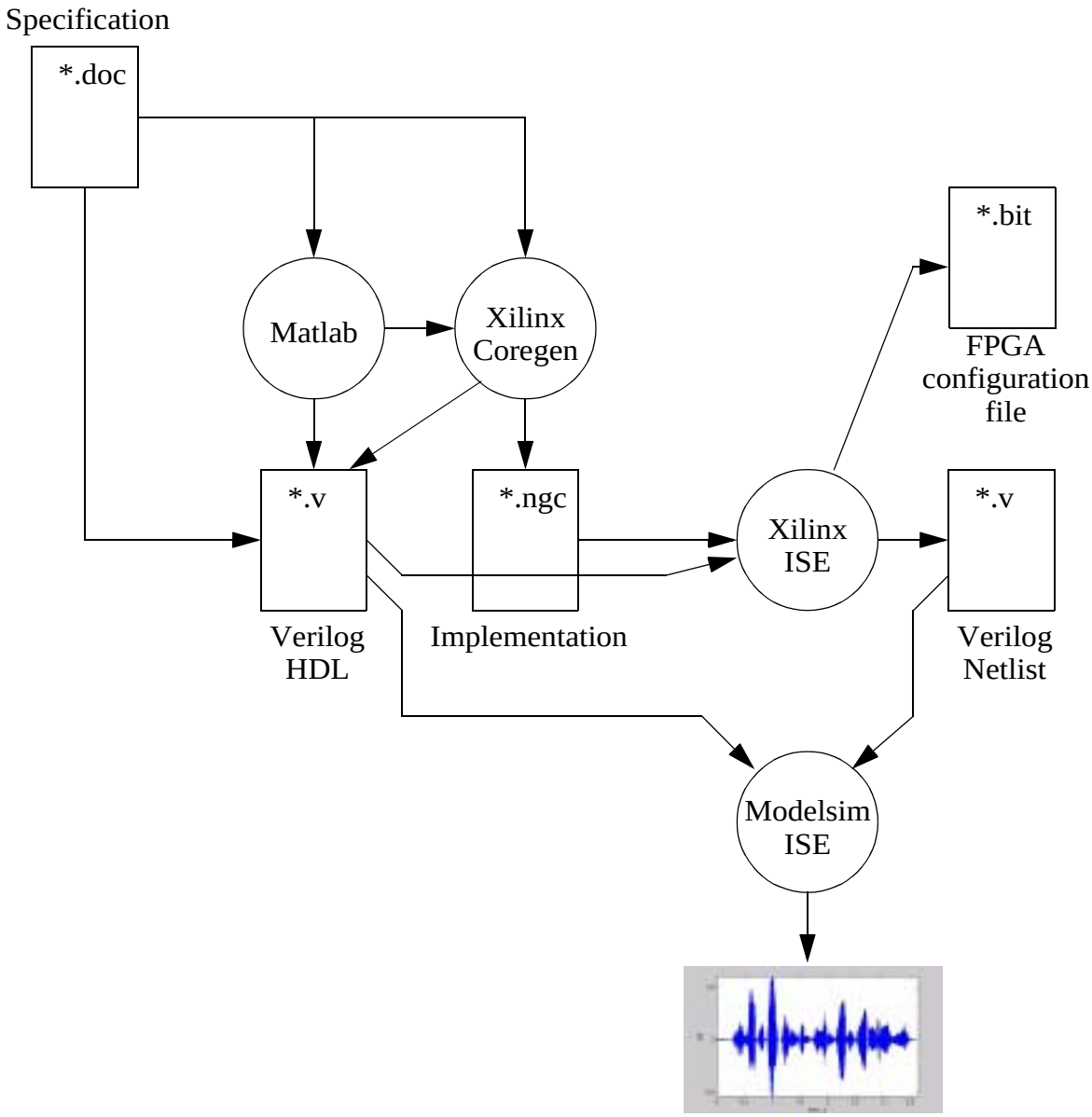


Figure 36 DSP implementation flow

From the specification of the DSP chip (and the AC'97 interface) the building blocks are generated using the Matlab Filter Design and Analysis tool fdatools and the Xilinx CORE Generator tool. The top-level DSP and its sub-components, clocking and interfaces (i.e. microphone to CIC filter, CIC to FIR3 filter, etc., FIFO and AC'97) are developed parallel to this. The functionality of the code is verified using Modelsim ISE. Once the system is working in simulation, the HDL code and the Xilinx CORE Generator implementation files are synthesised, translated, mapped and placed and routed using Xilinx ISE. The output of ISE is a binary programming file with which the Xilinx Spartan 3A (XC3S700A-FG484 - see [20]) chip can be configured and a post-route HDL netlist generated. Using this netlist the DSP can be re-

simulated using real timing information and compared, making sure that the reality matches the model.

(USB) Interface (IF)

Besides the DSP, the Interface (IF) is the second biggest component of the digital microphone array as shown in Figure 28. The TUSB3200A USB Streaming Controller (STC) has been selected for the Interface (IF) between the DSP which delivers the audio data via the AC'97 interface and the PC, i.e. the recording SW.
Figure 37 below shows the functional block diagram of the TUSB3200 USB STC.

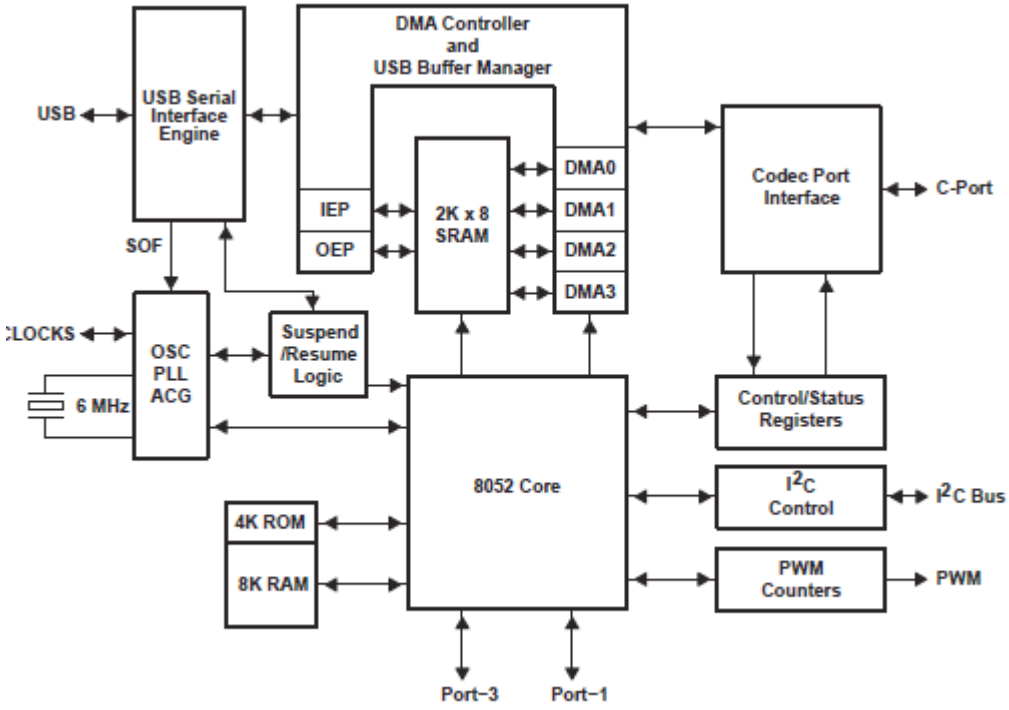


Figure 37 TUSB3200A Functional Block Diagram (from [16])

Three building blocks shown in Figure 37 are of interest for the digital microphone array. These are:

- Codec Port Interface
- DMA Controller and USB Buffer Manager
- 8052 core

The TUSB3200A USB STC is the only chip that could be found that allows the simultaneous streaming of eight channels of audio[1]. The 8052 core manages the USB protocol. The

1. This was confirmed by TI through personal communication during the feasibility study of the project.

TUSB3200A USB STC is provided with example FW (designed for stereo playback and mono record) allowing easy and fast expansion of the existing driver FW[1]. The DMA controller and USB Buffer Manager enable flexible loading of the audio from the AC'97 slots in the Codec Port Interface and send them upstream to the USB IF.

The amount of data that can be transferred using the USB v1.0 interface is also critical. The USB specification guarantees a minimal polling time of 1 ms, i.e. at least every 1ms a port is being checked and data up-loaded (see [1], [92] and [93]). This defines the maximum buffer size of the design to:

$$max(buffersize) = \frac{t(min)}{1/f_s} \cdot n \cdot w = \frac{1ms}{1/(48kHz)} \cdot 8 \cdot 2bytes = 768bytes \quad \text{Eq. (12)}$$

The full clock speed of the USB v1.0 interface allows a data rate of 12 MBits/s. Transferring 8 channels of audio data each with 16 bits width at 48 kHz requires 6.144MBits/s, therefore using about half of the available bandwidth. The USB specification guarantees safe transfer of data up to 90% of the available bandwidth. Transferring eight channels of audio data from the digital microphone array is therefore feasible.

Taking the existing FW of the TUSB3200A evaluation board, three types of modifications to the digital microphone array are necessary. These required changes are:

- USB descriptor (devdesc.c)
- DMA configuration (codec.c and mmap.h)
- Codec Port Interface (device.c)

The complete example FW provided by H. Nguyen is rather complex ([14]) compared to the requirements of the digital microphone array. Due to the limited time available to generate the new USB descriptor and setting up the TUSB3200A for the digital microphone array, the existing FW has been modified instead of writing clean and clear new FW.

The USB descriptor used for the digital microphone array is shown in the following Figure 38.

1. The TUSB3200A with the 8052 core is also compatible with the TI TAS1020B USB streaming controller, the predecessor of the TUSB3200A. The TUSB1020A is also the prime choice of Wolfson Microelectronics for their customer evaluation boards, therefore giving me access to their knowledge database and resources.

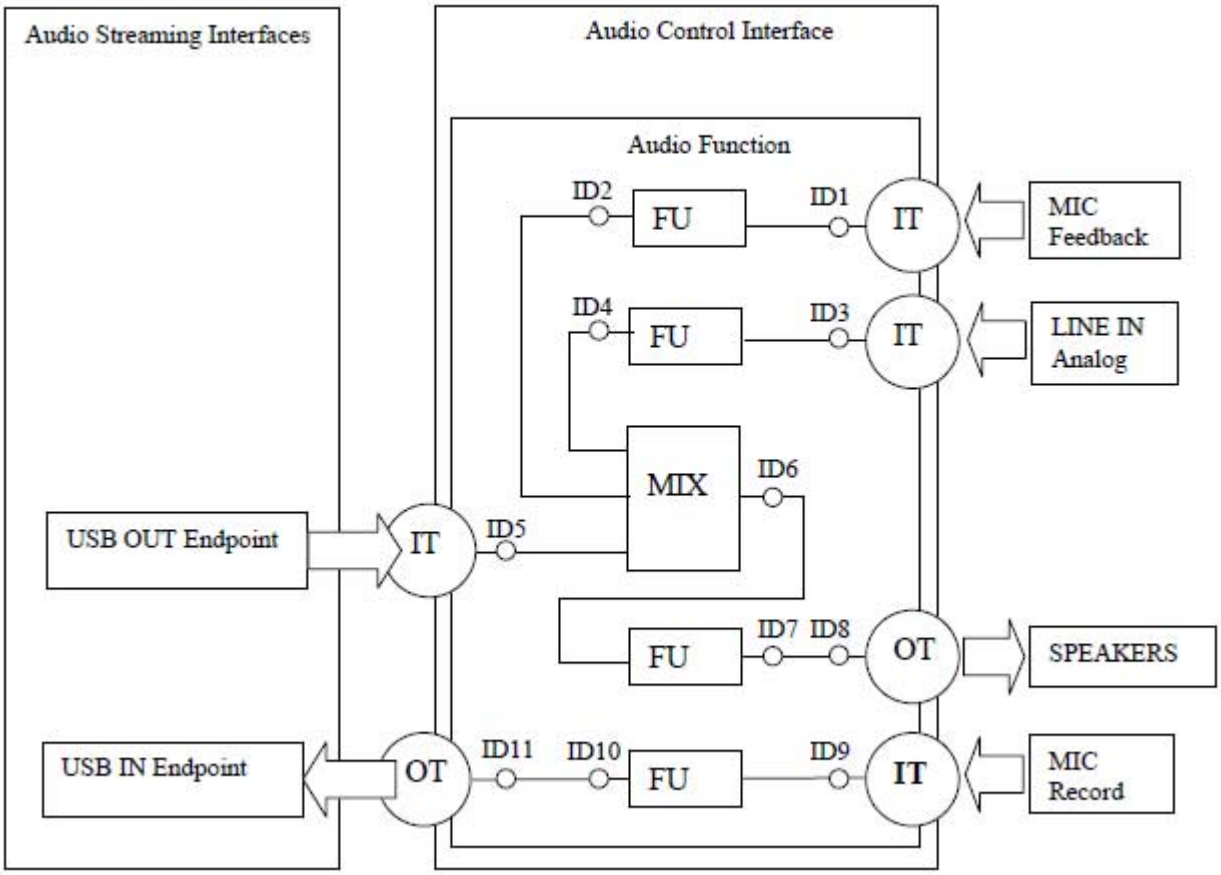


Figure 38 USB Audio Device Model (from [14])

The principle of a USB descriptor for an audio device can be understood by considering only a few points:

A microphone (audio input)[1] to the USB port on the PC is a USB In Endpoint requiring:

- an n-channel microphone Input Terminal IT
- a Feature Unit FU (describing features available for the channel, e.g. volume control or mute)
- a Mixer Unit MU describing how many channels are mapped in what way to the Output Terminal
- a USB endpoint descriptor, defining the setup and channels for the endpoint

The number of endpoints for a USB device is limited by the protocol and typically 15 ([93]). Each endpoint can contain a maximum of 254 channels ([92]). The existing FW contains a USB Out Endpoint terminal (for stereo DAC playback), a USB In Endpoint terminal (for the

1. For a speaker output this order is reversed.

microphone record path), an Endpoint for Interrupts and an Endpoint for the Human Device Interface (HDI) descriptor. The Interrupts and HDI endpoints are required to comply with the specification.

The example FW provided by TI as shown in Figure 38 contains an additional microphone feedback and line input path to the speaker output. These are irrelevant for the microphone array and are ignored[1].

Modifications to the USB descriptor were necessary in the upstream path with regards to:

- the number of channels (in small steps from mono to stereo to 4, 6, 7 and 8 channels)
- the maximum packet size (as calculated shown in Eq. (12))

In addition, the Codec Interface configuration of the TUSB3200A device needed to be modified in such a way that the AC'97 interface slots 3 (mono), 3 and 4 (stereo), etc. up to 3 to 10 could be accessed by the DMA controller.

Finally, the DMA Controller and the memory usage had to be adapted to the number of channels that were accessed and transferred. TI allows DMA access from addresses F800h to FF27h, a total address range of 1832 bytes as shown in Figure 39 below.

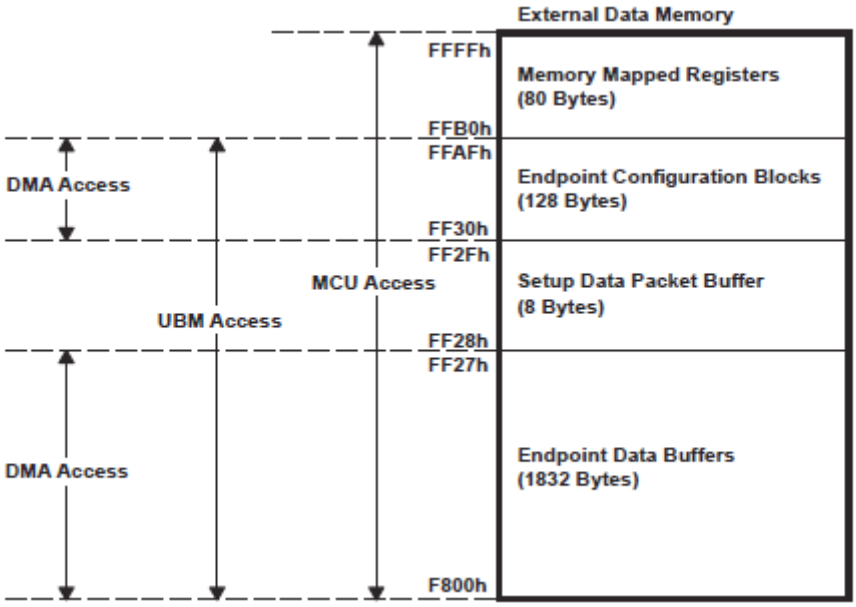


Figure 39 TUSB3200A Buffer Space Memory Map

Within the available address space about 50% is used for the upstream data, 15% for the downstream data and a few bytes for the interrupt control and HDI descriptor.

Note: Details of the example FW provided by TI can be found in [14].

1. Ideally they would be deleted but due to the complexity of the USB standard this was not successful.

USB Firmware Flow

As mentioned in the previous section, the modifications to the FW were executed in incremental steps. The process used for this is shown in Figure 40 below.

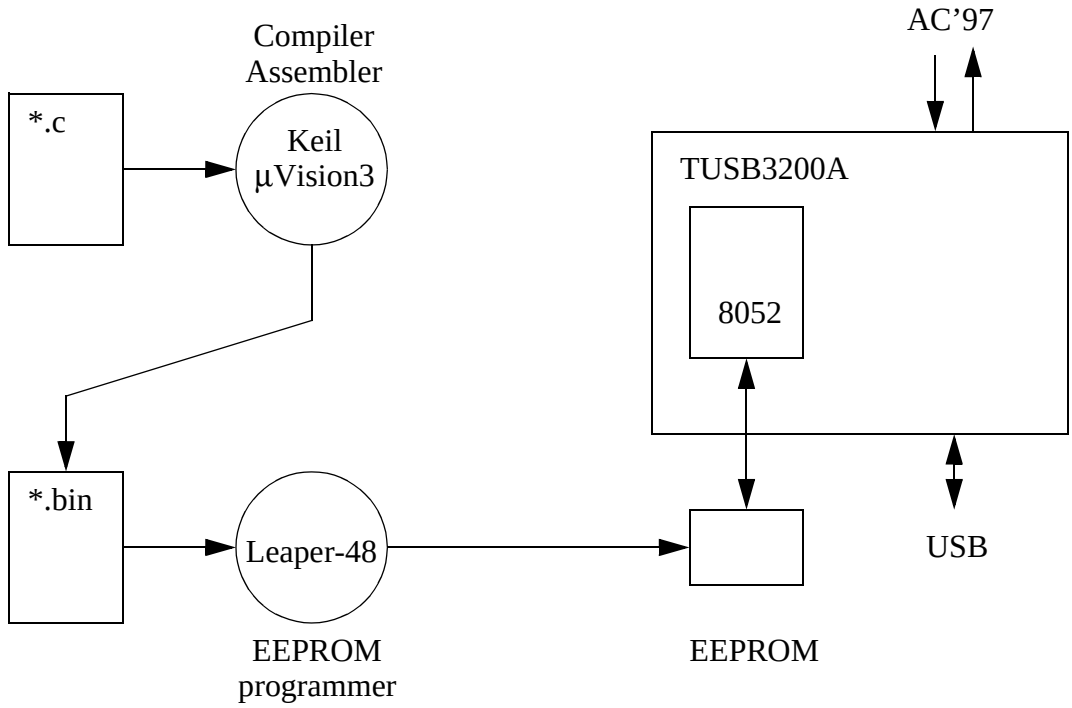


Figure 40 FW design flow[1]

In addition to the TUSB3200A evaluation board the Keil µVision3 8051[2] development kit was needed. The input to µVision3 is the sample code provided by TI. µVision3 generates a binary *.bin file which can be used to program an EEPROM on the TUSB3200A evaluation board [11]. A FW update is therefore performed by (re-)programming an EEPROM and replacing it on the evaluation board [13]. This has been done in the following sequence:

- regenerate existing FW, i.e. create backup EEPROM
- generate stereo ADC path
- generate 4-channel ADC path
- generate 8-channel ADC path

1. Both the Keil µVision3 assembler/compiler and the Leaper-48 EEPROM progammer were made available by Wolfson Microelectronics plc
2. The 8051 and 8052 cores are compatible, see [94]

FW and OS limitations

During the modification of the FW several limitation were found:

- Microsoft (MS) Operating Systems (OS) - both XP and Vista - do not support more than stereo channels[1]
- Using (Ubuntu) Linux seven channels of audio were successfully recorded

The limited time available led to a detour in the process of building the digital microphone array as no immediate support could be expected either from the Microsoft or Linux community. As an alternative plan an 8in4 channel scheme was devised and implemented. Although 48 kHz was chosen as the desired sample rate for the microphone array (see Figure 41), ASR systems typically work at a 16 kHz sample rate, enough to represent human speech. This has been made use of and a TDM-like scheme implemented, multiplexing twelve 16 kHz channels in four 48kHz slots, as shown in Figure 42 below.

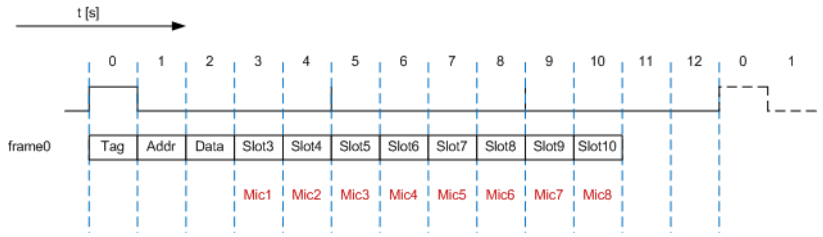


Figure 41 8in8 channel TDM

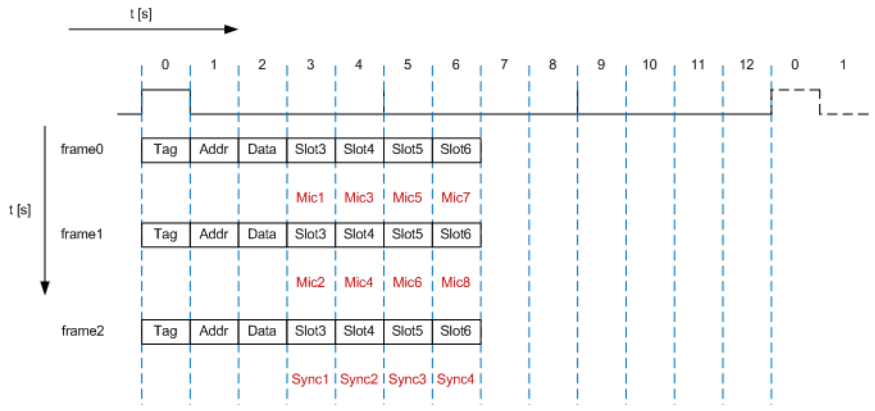


Figure 42 8in4 channel TDM

1. According to Microsoft, the XP and Vista Operating Systems support the universal audio architecture (UAA) and therefore multi-channel audio streaming [91], although being limited to 5.1 channels playback and stereo record. Vista should support four channels of microphone recording [117]. It is, however, not able to recognise the newly created microphone array if more than two channels are defined.

The alternative scheme of eight audio channels sampled at 16 kHz multiplexed into the 48kHz frame as available in the AC'97 specification requires post manipulation, using e.g Matlab. This is a major drawback to a realtime system (though does not render it unusable) but the only viable option available in the time frame of this project.

Further reading

It was only possible to give an overview of the signal and speech processing techniques used to design the digital microphone array in this section. Building the DSP and Interfaces and modifying the C code to update the USB firmware in the very short timeframe available were only possible due to my previous experience.
The list of recommended further reading provided here is useful for the interested reader if they need to deepen their understanding in a particular field.

- "Principles of Sigma-Delta Modulation for Analog-to-Digital Converters" by S. Park is a excellent introduction to SDM, ADCs and over-sampling [57]
- Most Electrical Engineering standard works will have a detailed introduction to signal processing and FIR filters. "Understanding Digital Signal Processing" by R.G Lyons [6] is highly recommended
- For the best low-power least area implementation of FIR filters, specifically half-band filters, refer to [79] (and [80] for a summary)
- A good introduction to USB is provided by Jan Axelson's "USB Complete: Everything You Need to Develop Custom USB Peripherals", 3rd edition, 2005 [1] [115]
- A very short introduction to C programming is given in T. Zhang's "Teach Yourself C in 24 Hours" by Sams Publishing [10]
- The best starting point on resources on the Intel 8052 core can be found at http://www.8052.com [94]
- Programmers from Zitek UK wrote an introduction to the C51 compiler, "C51 Primer - An Introduction To The Use Of The Keil C51 Compiler On The 8051 Family", available through their sales department [4]

ASR - Methodology

As previously mentioned, the CSTR instrumented meeting room (IRM) at the University of Edinburgh is equipped with an analogue microphone array. The performance of this analogue array is well known and documented (see e.g. [34] or [43]). This analogue microphone array (and all the other equipment and wiring) is fixed to the tables and floor of the IMR.
The following section analyses and compares the speech recognition performance (in terms of WER) of the newly built digital microphone array with the existing analogue microphone array. First the corpus used for the recordings is presented, followed by a description of the tools used for the beamforming and speech enhancement. Next an outline of the speech recognition software is given and last the specification of the test and evaluation method.
The methods used and the dataflow are illustrated in Figure 43 below.

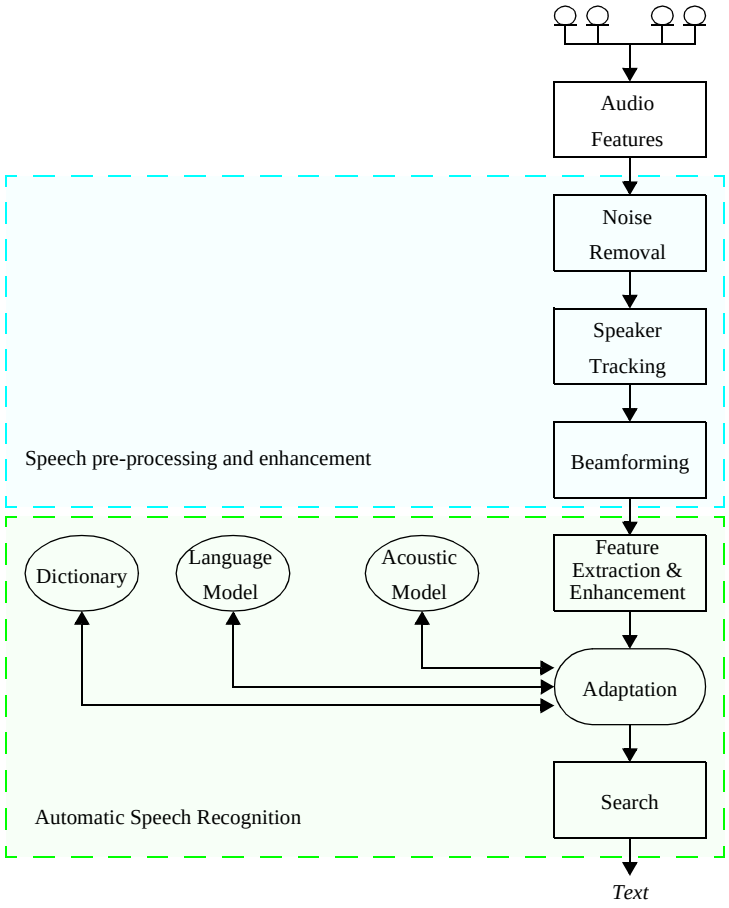


Figure 43 Architecture of a DSR System (with kind permission of [8])

In order to analyse and compare the digital and analogue microphone arrays, a body of recorded speech is required. Processing of the recordings is carried out first by running some speech pre-processing and enhancement methods followed by the automatic speech recognition.

Baseline system

The baseline system used for the analysis of the digital microphone array is identical to the one used for the multi-channel Wall Street Journal audio visual corpus (MC-WSJ-AV), as defined by Lincoln et al. [43]. The HMMs for this baseline system are taken from the U.K. English Cambridge version of the Wall Street Journal (WSJ) WSJCAM0 (see Fransen et al. [31]). The original Wall Street Journal (WSJ) corpus is described by Paul and Baker [56].
The speaker independent WSJCAM0 database is available from the Speech Separation Challenge [124]. Over 140 speakers[1] each speaking about 110 utterances were used to create the WSJCAM0 corpus of spoken British English (see Robinson et al. [61] 1995). These recordings were made with a head-mounted microphone in an acoustically isolated room. The recorded sentences were taken from the WSJ corpus. A British English pronunciation dictionary (BEEP) and trigram language models from the 5k and 20k WSJ corpus are also part of [124].

The specifications of the baseline recognition system used to evaluate the analogue and digital microphone arrays are:

- HMMs trained with HTK on the WSJCAM0 database
- 53 male and 39 female speakers with British English accents
- 11000 tied-state triphones
- three emitting states per triphone, 6 gaussian mixture components per state
- 52-element feature vectors (comprising 13 MFCCs and 0th cepstral coefficient) with 1^{st}, 2^{nd} and 3^{rd} order derivatives

Beamforming and speech enhancement

The output recordings of the microphone arrays are eight channels of raw audio data. Speech pre-processing and enhancement is necessary before any speech recognition can be run. The

1. 92 from the 140 speakers were used for the training set, 20 for the development test set, and two times 14 for two evaluation test sets.

mdm-tools [46], a set of software tools for processing multiple distant microphone signals, were used for:

- noise removal
- speaker tracking

and

- beamforming (and post-filtering).

In the first stage the stationary noise in the multi-channel audio file(s) is removed by Wiener filtering in two passes (see [46]). After this, blind source speaker tracking is performed. The source(s) of the signal are tracked by calculating the cross correlation [105] of the different channels. A peak in the cross correlation function of two channels indicates a signal source. Using the speaker tracking and delay files the beam can be formed. A standard superdirective beamformer is engaged to generate a one channel output waveform file which is then used for the speech recognition. The output of the beamformer is post-filtered using Wiener filters to further improve the SNR.

The algorithms used for speech enhancement and beamforming are:

- Cross-correlation to determine the signal (or speaker) source as defined in Knapp and Carter [42]
- Superdirective adaptive beamforming as defined in Cox et al. 1986 [27] and Cox et al. 1987 [26]
- Wiener filtering (see Simmer et al. [7] and the reference therein for an introduction)

Speech Recognition

Automatic Speech Recognition (ASR) is performed on the microphone array data using HTK [114]. HTK, the HMM Tool Kit , is designed to build speech recognition systems using HMMs (Hidden Markov Models). The basic flow for running speech recognition in the setup as defined by Lincoln et al. [43] is shown in Figure 44 below.

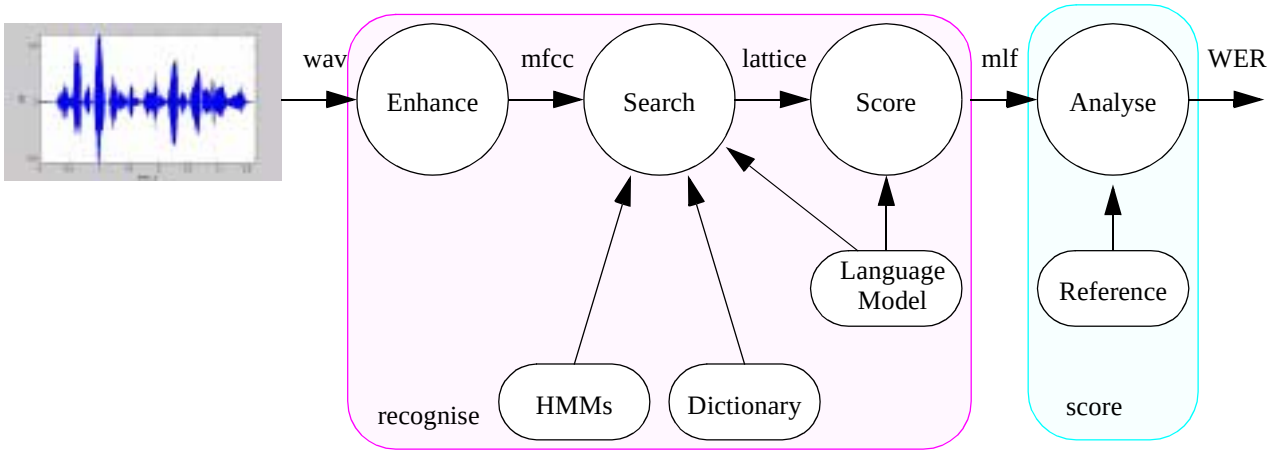


Figure 44 ASR flow

Two scripts have been designed to run the speech recognition, recognise.bsh and score.bsh. These scripts are self-documenting and it is not within the scope of this dissertation to present them in detail. Please note that in these bash scripts the actions are executed in sequence as shown in Figure 44. In addition to the basic steps, many checks are integrated which make sure that all the necessary information is available before HTK is run. Speech recognition is performed by carrying out the following steps:

- enhance, to generate MFCC coefficients from the wav files
- search, to run the Viterbi search algorithm and produce the lattice of best matching sentences given the input utterance
- score, to re-score the output lattice from HVite using a language model and to output the best sentence

Scoring of the recogniser output is done using

- analyse, to calculate the WER (word error rate) comparing the reference sentence with the recognised output

HMM adaption

The WSJCAM0 database ([61] from [124]) used for the analysis of the two microphone arrays is speaker independent (SI) and has been recorded using head-mounted microphones. In order to achieve improved results the HMMs can now be adapted to different scenarios. Such scenarios are:

- adaptation to microphone array (i.e. digital vs. analogue),
- adaptation to gender (i.e. female vs. male),

- adaptation to speaker (using speaker dependant adaptation sentences)

and

- any combination of the above.

HTK supports two kinds of adaptation (see [114] for details):

- MLLR (Maximum Likelihood Linear Regression)
- MAP (Maximum A Posteriori)

Note: Please read "Automatic Speech Recognition and Distant Speech Recognition" on page 34 for details on MAP and MLLR adaptation. An excellent introduction to speech recognition and speaker adaptation is also provided by Young (2009) [77] and the references therein.

In order to adapt the HMMs that were initially provided for the evaluation some modifications to the flow are necessary. This modified flow is presented in Figure 45 below.

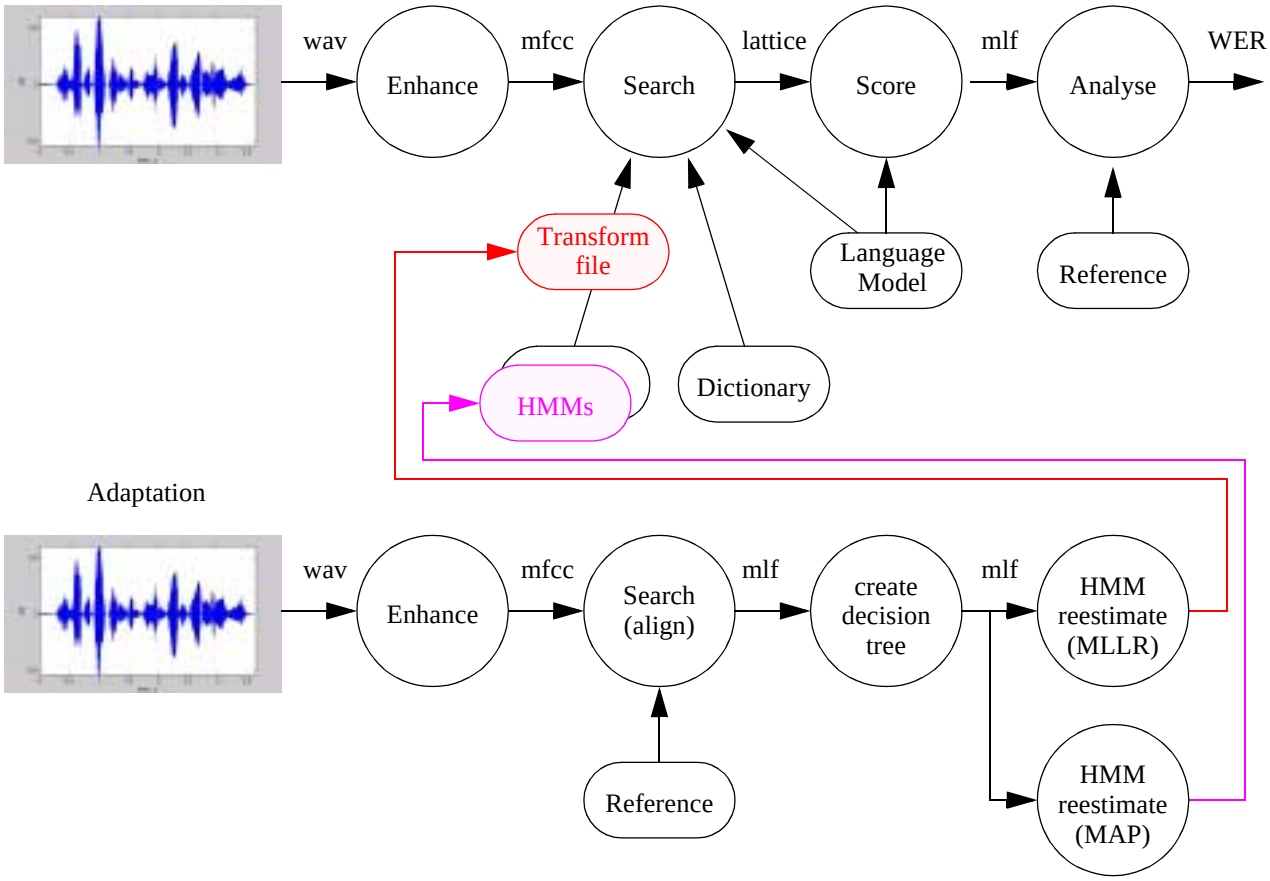


Figure 45 Adaptive ASR flow

As shown in Figure 45 above, in the adaptive flow a transform model file (TMF) is generated using MLLR. The TMF is located between the HMMs and the search algorithm, therefore modifying the HMM parameters when they are accessed by the Viterbi search.
For MAP adaptation, the means and variances of the individual HMMs are modified, i.e. new HMMs are generated. This requires significant amounts of adaptation data [77]. For the evaluation of the analogue and digital microphone array only a limited number of speakers could be recorded so insufficient amounts of data are available for performing MAP adaptation.

MLLR adaptation was therefore chosen to evaluate the microphone arrays. TMF files (and also the adapted HMMs for MAP) are created by the following steps:

- enhance, to create MFCC files from the adaptation sentences
- search (align mode), to align the adaptation sentences, i.e. to assign the adaptation specific tri-phones to the MFCCs
- create decision tree, to create a decision tree and to decide which models have enough data in order to modify/shift their HMM data
- HMM reestimation (run twice), to create TMFs. In the first pass global adaptation is performed and in the second pass this global adaptation forms the input transformations with which better frame and state alignments are produced. The transformation model files (TMF) act at specific points in the phone decision tree that is used by the search algorithm when accessing the HMMs.

Note: The flow described here follows the recommended adaptation flow as laid out by the HTK manual, chapter 3.6, “Adapting the HMMs” [114].

Test and evaluation of the results

The test setup and evaluation method for comparing the analogue and digital microphone arrays are presented next. The following procedure is applied for testing and evaluating the arrays:

1. Recognise individual speakers
2. Generate adaptation data for different microphones (analogue/digital)
3. Repeat step 1
4. Generate adaptation for gender and microphone type (female/male and analogue/digital)
5. Repeat step 1

In the first adaptation run, only the Gaussian means of the HMMs are modified. In the second run both means and variances are adapted.

ASR - Setup and Results

The following section gives a description of the setup used for recording the WSJ sentences followed by presentation of the results.

Equipment

As mentioned previously, this dissertation compares an analogue microphone array with the newly built digital MEMS microphone array. The analogue array recording setup contains:

- (8 x) Sennheisser MKE 2-P-C microphones [15]
- Motu 896mk3 firewire audio interface [120]
- Firewire interface on PC running Microsoft Vista [86]
- Bidule recording SW [103]

The digital microphone array setup contains:

- (8 x) Knowles digital MEMS microphones [12]
- Digital microphone array as specified in “DMA - Building”
- PC running Linux Xubuntu [133]
- Audacity recording SW Version 1.3.8 [100] (capturing 8 in 4 channels TDM)
- script to generate 8-channel wav file

The equipment as set up for the recording is shown in Figure 46 below.

Figure 46 Recording SW

On the right side is the PC running Xubuntu Linux recording the digital microphone array and on the left side is the PC running Microsoft Windows Vista recording the analogue microphone array.

Setup

The analogue microphone array is fixed in the middle of several tables in a meeting room at the Centre for Speech Technology Research (CSTR), Room 3.07, in the University of Edinburgh Informatics Forum. The digital microphone was therefore placed in an identical position and the prompter located between the two. This symmetrical setup, as illustrated in Figure 47 below, should put both microphone arrays into a similar environment.

All sources of noise were placed as far away from the array as possible, i.e. the recording PCs were located at the far end of the meeting room.

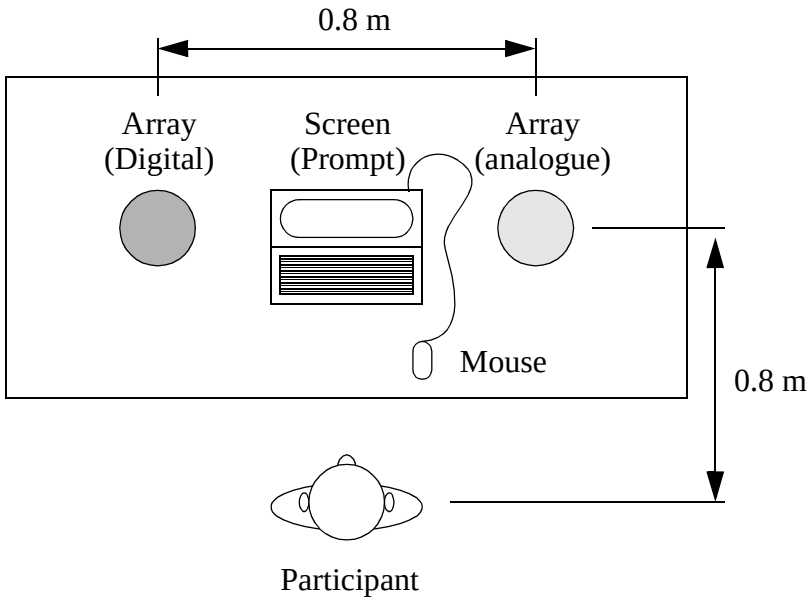


Figure 47 Recording setup

A photo of the setup as defined in Figure 47 is shown in Figure 48 below.

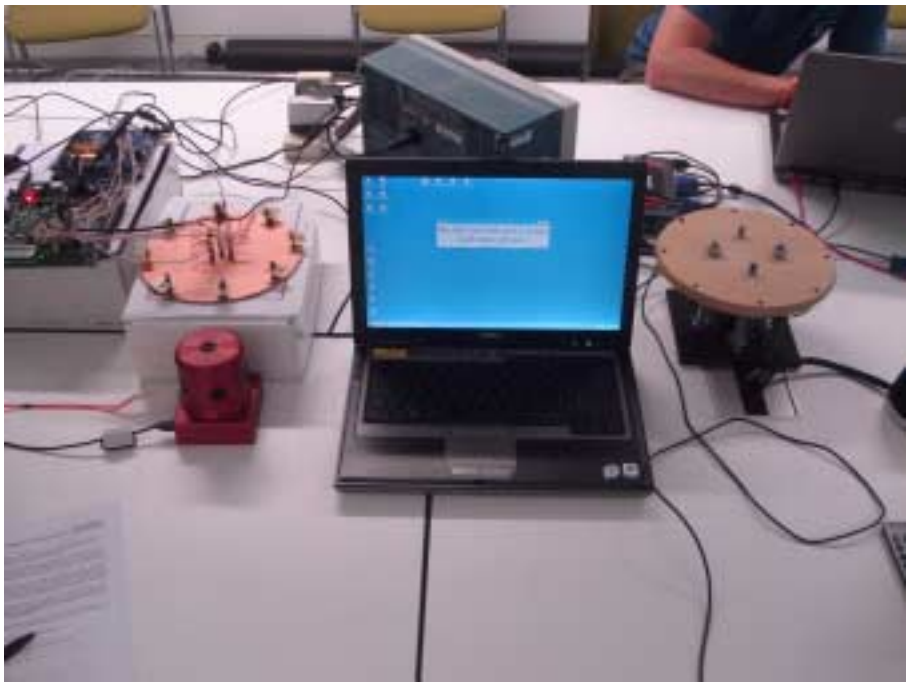


Figure 48 Photo of Recording Setup

The PC running the prompting SW was placed in between the two microphone arrays in such a way so as not to hinder the sound path while the prompts are still readable, i.e. not too far away.

Prompter

A prompter was developed for the recording of the MC-WSJ-AV corpus [43]. This same prompter was used for the evaluation of the microphone arrays and is shown in Figure 49 below.

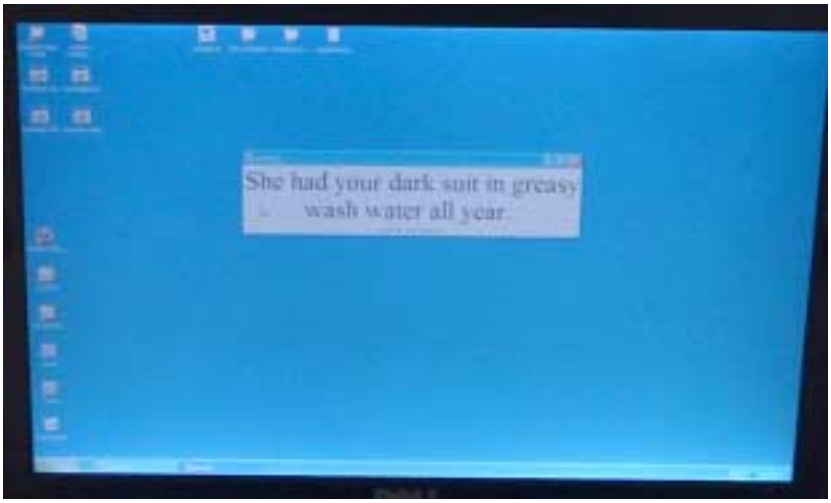


Figure 49 MC-WSJ_AV prompter (screen shot)

The participants read the sentences from the screen. Sentences are displayed in a semi-random order in six blocks. Each block starts with a few adaptation sentences followed by some sentences from the WSJ 5k and then 20k vocabulary. The sentences are more readable if a few of them are presented in order. Participants progress to the next sentence by clicking into the prompt window. Though instructed to wait at the end of the sentence, a few participants clicked while still talking or immediately after the end of the sentence. This is audible in the recordings. In addition, building works were being carried out on the recording days (20th and 21st July 2009), affecting participant T36 most.

Each participant read approximately 94 sentences and 1600 words which required about fifteen minutes.

Participants

Twelve participants were asked to lend their voices. Each participant received a short introduction to the purpose of the experiment and the following instructions:

1. Please sit in this chair placed in between the two arrays
2. Speak with a normal voice
3. Speak into the screen
4. Read the prompts from the screen

5. Progress by clicking the left mouse button
6. Please pause before progressing to the next sentence using the mouse
7. Ignore the “change seat” prompts[1]

In addition, each participant was asked to fill in a form confirming that their recordings can be used within the AMI/AMIDA project. This form is attached in “Appendix B” on page 118. Each participant was either paid £5 for their participation or £5 were given to a charitable organisation.

The participants’ details are summarised in Table 2.

Table 2: Participants[a]

Participant	Gender	Age[b]	Test Set	Recorded	Notes
1	Male	35	T7	ana: 48k /dig: 16k	ML
2	Male	-	T8	ana: 48k/dig: 16k	KE
3	Female	58	T9	ana: 48k /dig: 16k	BB
4	Male	36	T10	ana: 44k1 /dig: 16k	KR
5	Male	36	T21	ana: 48k /dig: 16k	RC
6	Female	40	T22	ana: 48k /dig: 16k	AI
7	Male	39	T23	ana: 48k /dig: 16k	SK
8	Male	44	T24	ana: 44k1/dig: 16k	SR
9	Female	-	T25	ana: 44k1/dig: 16k	LH
10	Female	55	T34	ana: 44k1/dig: 16k	BM
11	Female	52	T46[c]	ana: 44k1/dig: 16k	SB
12	Female	-	T37	ana: 44k1/dig: 16k	HH
13[d]	Male	39	T40	ana: 44k1/dig: 16k	EZ

a. All participants live in Edinburgh and speak neutral (RP) English
b. Declaration of age was optional
c. Participant 11 was recorded before participant 4 and T10 was used for her by mistake. The data from T10 was therefore copied into a newly created test set T46
d. Participant 13 is a non-native speaker of English and has been included for test purposes only

1. The prompter was designed for non-stationary speech recognition, i.e. sentences are to be read from six different pre-defined positions.

Data preparation

The recorded wav files from the participants need post-processing before they can be used for speech recognition. This process is different for the analogue and digital microphone arrays due to the 8in4 channel TDM scheme which needed to be used for recording the digital microphone array data over the USB interface. The steps required to get a WER from the wav recordings are:

1. Downsample wav files from analogue microphone array:
    - from 48 kHz to 16 kHz (using ch_wave [104])
    - from 44.1 kHz to 16 kHz (using Matlab[1])
2. Separate individual channels from digital microphone array recording (8in4 channels) (using Matlab script)
3. Apply noise reduction, beamforming, and post-filtering (using mdm_tools [46])
4. Transcript files as described below
5. Run automated speech recognition (ASR)

Two scripts needed to be developed for the post-processing of the wav files. The first script is used to downsample the wav files from 44.1 kHz to 16 kHz. This needs to be done on a per channel basis due to the large memory usage[2]. The second script is used to extract the eight audio channels, i.e. to find and discard the synchronisation channels and to put the eight channels in the correct order.[3]

Next, the utterances from the analogue and digital microphone array recordings are split into individual files. These files need to be named correctly to enable comparison of the recognised text with the reference text in order to calculate the WER.

The actual process of automatic speech recognition can now be run with the completed individual files for each utterance.

1. The recording was carried out over two days. All equipment was powered off during the night. The recordings of the second day were done at 44.1 kHz due to the Motu 896mk3 firewire audio interface defaulting to this sample rate.
2. Typical recordings last 15 minutes, eight channels of 16-bits (= 2 bytes) wavefiles at 48 kHz sample rate occupy 700 MBytes
3. The core of both scripts is matrices manipulation which utilises the strength of Matlab which was used to write the scripts. Both scripts are designed for fast performance so that the processing can be carried out in a real time system.

Results

After running the basic speech recognition tests four different adaptation scenarios were defined. The aim is to analyse the effects of channel (analogue vs. digital), gender (female vs. male) and individual (e.g. T7 vs. T8) on the WER. The four scenarios are defined in Figure 50 below.

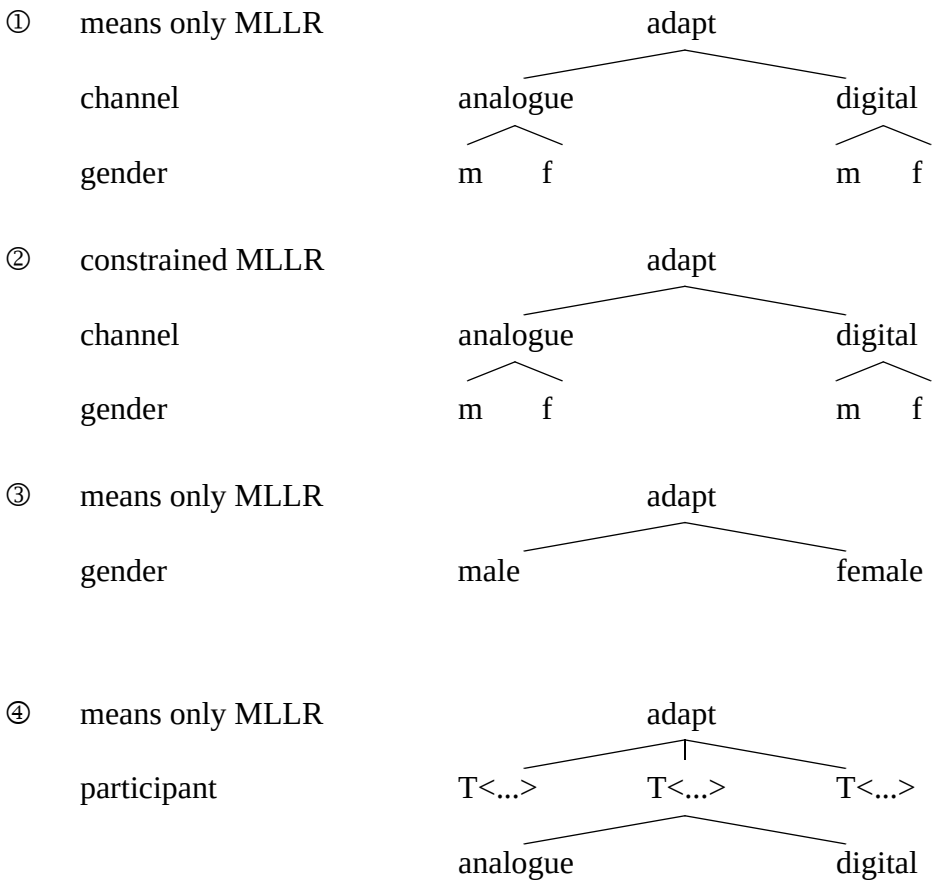


Figure 50 Adaption scenarios

First, combined effects of channel and gender using means-only MLLR adaptation is analysed. Second, these tests are repeated using constrained MLLR. Third, the effect of gender is looked at independent of channel, and last, adaptation for the individual participant and channel is performed.

WER

The final WER of the recogniser is determined by matching each of the recognised sentences with a reference sentence, i.e. by performing an optimal string match using dynamic programming. A typical output of such a process is:

```
====================== HTK Results Analysis =======================
  Date: Wed Aug  5 22:02:37 2009
  Ref : results/T7/T7.5k.digital.mlf
 Rec : /T7c0201-d.rec
     : /T7c0202-d.rec
------------------------ Overall Results --------------------------
SENT: %Correct=2.56 [H=1, S=38, N=39]
WORD: %Corr=70.14, Acc=64.66 [H=512, D=27, S=191, I=40, N=730]
====================================================================
```

The line starting with SENT: indicates that of the 39 test utterances, 1 (2.56%) was correctly recognised. The following line starting with WORD gives the word level statistics and indicates that of the 730 words in total, 512 (70.14%) were recognised correctly. There were 27 deletion errors (D), 191 substitution errors (S) and 40 insertion errors (I). The accuracy figure (Acc) of 64.6% is lower than the percentage correct (Corr) because it takes account of the insertion errors which Corr ignores.

For this dissertation WER (Word Error Rate) is calculated using Acc, i.e. WER = 100% - Acc.

Results with default settings

First, each participant's utterances were recognised. The calculated WER is shown in Table 5 below.

Table 3: Initial WER[a]

Participant	analogue		digital		Δ	
	5k	20k	5k	20k	5k	20k
T7	21.92	35.13	35.34	48.76	13.42	13.63
T8	40.51	58.85	48.16	63.56	7.65	4.71
T9	28.89	55.63	39.56	64.09	10.67	8.46
T10	15.62	35.39	21.16	45.62	5.54	10.23
T21	46.87	68.76	68.69	80.62	21.82	11.86
T22	40.09	61.68	61.14	80.41	21.05	18.73
T23	38.41	58.26	47.4	67.88	8.99	9.62
T24	17.86	44.84	23.14	49.57	5.28	4.73
T25	41.72	66.07	59.64	73.48	17.92	7.41
T34	52.56	81.01	79.97	91.33	27.41	10.32
T36	23.3	47.78	33.95	61.41	10.56	13.63
T37	34.93	65.26	56.18	79.64	21.25	14.38
T40	59.97	83.97	84.51	98.97	24.54	15
Average male	30.20	50.21	40.65	59.34	10.45	9.13
Average female	36.92	62.91	55.07	75.06	18.16	12.16
Average	33.56	56.56	47.86	67.2	14.3	10.64

a. T<*n*> are the participants as defined in Table 2, the average WERs of the male, female and all participants are also shown

Note: Lincoln et al. [43] achieved an average WER of 55.2% compared to 33.6% in this project. The digital MEMS microphone array achieves 47.9%. Note that in this project the speakers are stationary while in Lincoln et al. they read from six different positions.

Results with adapting the means

Alignment of the individual's adaptation sentences is necessary in order to run the ASR in adaptation mode. The alignment process itself is a very efficient method for checking whether the participants said what they were told to say. The process detected three kinds of errors, all of which are shown in Table 4 below.

Table 4: Report from alignment process[a]

Omitted	Corrupted	Unknown
T8a0102-a T8a0102-d	T8c0209-a T9a0104-a T23a10b-a T25a105-a	T9a0105-a T21a010g-a T23a010g-a T21a010g-d T23a010g-d T34a010d-d

a. T8a0102-a: Participant 8, adaptation sentence 0102, analogue microphone
T23a010g-d: participant 23, adaptation sentence 010g, digital microphone

In the case of participant 8, sentence 0102 was not recorded, as shown in "omitted". Corrupted sentences are sentences which the recording SW (i.e. the recording PC) distorted. In all these sentences several seconds of speech are missing. Sentences marked "unknown" did not show any problems when listened to, but nevertheless caused problems in the alignment process. Those problems were overcome by increasing the threshold of the alignment process.

The calculated WERs obtained by running the ASR using means-only MLLR adaptation to the channel are shown in Table 5 below.

Table 5: WER after adaptation to channel (means only)

Participant	analogue		digital		Δ	
	5k	20k[a]	5k	20k[a]	5k	20k[a]
T7	15.89	n.a.	22.6	35.5	6.71	n.a.
T8	31.02	52.12	30.74	48.38	0.28	3.74
T9	22.61	50.61	25.9	56.05	3.29	5.44
T10	14.06	36.68	16.9	39.5	2.84	2.82
T21	28.8	47.52	38.64	n.a.	9.84	n.a.
T22	21.2	46.2	35.02	n.a.	13.82	n.a.
T23	30.62	n.a.	35.99	n.a.	5.37	n.a.
T24	15.37	n.a.	19.25	n.a.	3.88	n.a.
T25	22.57	52.63	28.86	58.41	6.29	5.78
T34	29.33	62.77	49.84	73.59	20.51	10.82
T36	13.78	43.0	19.89	46.91	6.11	3.91
T37	23.72	49.74	33.77	61.19	10.05	11.45
T40	49.54	n.a.	67.72	n.a.	18.18	n.a.
Average male	22.63	n.a.	27.35	n.a.	4.73	n.a.
Average female	22.2	n.a.	32.21	n.a.	10.01	n.a.
Average	22.41	n.a.	29.78	n.a.	7.37	n.a.

a. Due to system instability during running of these tests only part of the 20k results are available.

Note: Lincoln et al. [43] achieved an average WER of 36.2% compared to 22.4% in this project. The digital MEMS microphone array achieves 29.8%. Note that in this project the speakers are stationary while in Lincoln et al. they read from six different positions.

The calculated WERs obtained by running the ASR using means-only MLLR adaptation to the channel and gender are shown in Table 6 below.

Table 6: WER after adaptation to microphone and gender (means only)[a]

Participant	analogue	digital	Δ
T7	15.34	22.33	6.99
T8	27.62	29.04	1.42
T9	19.78	24.18	4.4
T10	14.2	15.77	1.57
T21	29.34	37.03	7.69
T22	21.81	33.49	11.68
T23	28.03	34.78	6.75
T24	15.05	17.68	2.81
T25	21.48	31.87	10.39
T34	25.96	49.36	23.4
T36	14.2	19.32	5.12
T37	24.26	33.28	9.02
Average male	21.6	26.14	4.54
Average female	21.25	31.92	10.67
Average	21.42	29.03	7.6

a. The results presented in this table are for the 5k WSJ corpus only

Results with adapting the means and variances

For the second test scenario constrained MLLR is used to create the adaptation data. The calculated WERs obtained by running the ASR using constrained MLLR adaptation to the channel are shown in Table 7 below.

Table 7: WER after adaptation to microphone (means and variances)[a]

Participant	analogue	digital	Δ
T7	14.25	19.86	5.61
T8	30.03	33.99	3.96
T9	16.8	20.97	4.17
T10	12.5	16.76	4.26
T21	26.12	34.53	8.41
T22	24.27	39.94	15.67
T23	30.97	35.64	4.67
T24	14.13	16.93	2.8
T25	20.38	27.77	7.39
T34	27.88	39.58	11.7
T36	15.77	18.75	2.98
T37	19.35	31.14	11.79
Average male	21.33	26.29	4.95
Average female	20.74	29.69	8.95
Average	21.04	27.99	6.95

a. The results presented in this table are for the 5k WSJ corpus only

The calculated WERs obtained by running the ASR using constrained MLLR adaptation to the channel and gender are shown in Table 8 below.

Table 8: WER after adaptation to microphone and gender (means and variance)[a]

Participant	analogue	digital	Δ
T7	14.66	18.08	3.42
T8	25.51	30.31	4.8
T9	15.23	20.32	5.09
T10	12.22	15.06	2.84
T21	24.51	38.28	13.77
T22	24.58	38.86	14.28
T23	27.85	33.93	6.08
T24	15.37	17.86	2.49
T25	21.48	31.87	10.39
T34	24.84	42.15	17.31
T36	12.64	18.75	6.11
T37	19.6	30.97	11.37
Average male	20.02	25.59	5.57
Average female	20.05	30.19	10.15
Average	20.03	27.89	7.86

a. The results presented in this table are for the 5k WSJ corpus only

Results with adapting to gender

For the third scenario WERs are measured for the ASR system with adaptation data for gender only. The calculated WERs obtained by running the ASR using means-only MLLR adaptation to gender are shown in Table 9 below.

Table 9: WER after adaptation to gender (means only)[a]

Participant	analogue	digital	Δ
T7	15.75	22.74	6.99
T8	29.32	28.9	0.42
T9	19.15	22.45	3.3
T10	15.06	17.33	2.27
T21	28.62	35.06	6.44
T22	19.15	22.45	3.3
T23	28.55	34.26	5.71
T24	15.53	17.39	1.86
T25	23.39	30.1	6.71
T34	27.88	48.72	20.84
T36	15.2	16.9	1.7
T37	24.38	33.77	9.39
Average male	22.14	25.95	3.81
Average female	21.48	27.88	6.4
Average	21.81	26.91	5.1

a. The results presented in this table are for the 5k WSJ corpus only

Results after adapting to the individual participant

For the final scenario WERs are measured for the ASR system with adaptation data for the participants using their own adaptation data. First, this is done independently of the channel. In the second run the different channel characteristics (i.e. analogue vs. digital) are also considered. The calculated WERs obtained by running the ASR using means-only MLLR adaptation to the participant are shown in Table 10 below.

Table 10: WER after adaptation to speaker (means only)[a]

Participant	analogue	digital	Δ
T7	15.89	18.49	2.6
T8	22.8	22.66	0.14
T9	14.13	19.0	4.87
T10	11.08	12.22	1.14
T21	20.11	25.58	5.47
T22	22.27	30.57	8.3
T23	25.61	29.24	3.63
T24	15.06	16.46	1.4
T25	22.02	27.5	5.48
T34	23.08	34.17	11.09
T36	11.08	14.77	3.69
T37	23.23	28.5	5.27
Average male	18.43	20.78	2.35
Average female	19.3	25.75	6.45
Average	18.86	23.26	4.4

a. The results presented in this table are for the 5k WSJ corpus only

Note: Lincoln et al. [43] achieved an average WER of 31.6% compared to 18.9% in this project. The digital MEMS microphone array achieves 23.3%. Note that in this project the speakers are stationary while in Lincoln et al. they read from six different positions.

The calculated WERs obtained by running the ASR using means-only MLLR adaptation to the participant and channel are shown in Table 11 below.

Table 11: WER after adaptation to speaker and channel (means only)[a]

Participant	analogue	digital	Δ
T7	14.66	18.49	3.83
T8	23.23	23.8	0.57
T9	16.01	19.62	3.61
T10	10.65	13.36	2.71
T21	21.47	23.79	2.32
T22	19.97	32.26	12.29
T23	25.09	26.99	1.9
T24	13.98	17.55	3.57
T25	22.3	26.4	4.1
T34	23.4	34.13	10.73
T36	12.93	18.04	5.11
T37	21.58	25.04	3.46
Average male	18.18	20.66	2.48
Average female	19.37	25.92	6.55
Average	18.77	23.29	4.52

a. The results presented in this table are for the 5k WSJ corpus only

These results will be analysed and discussed in the next section.

Analysis and Discussion

This section analyses the results from the analogue and digital microphone arrays and discusses the findings.

Analysis

WERs were measured from the recordings of the two arrays using ASR techniques. These WERs are presented in “ASR - Setup and Results” on page 75. The different options that were analysed are:

- basic ASR vs. ASR using adaptation techniques
- channel, i.e. analogue vs. digital microphone arrays
- gender, i.e. male vs. female
- ASR adaptation technique, i.e. mean-only vs. constrained MLLR
- ASR adaptation of the individual and
- channel adaptation of the individual

Graphs showing the measured WERs are presented in the following Figure 51-60.

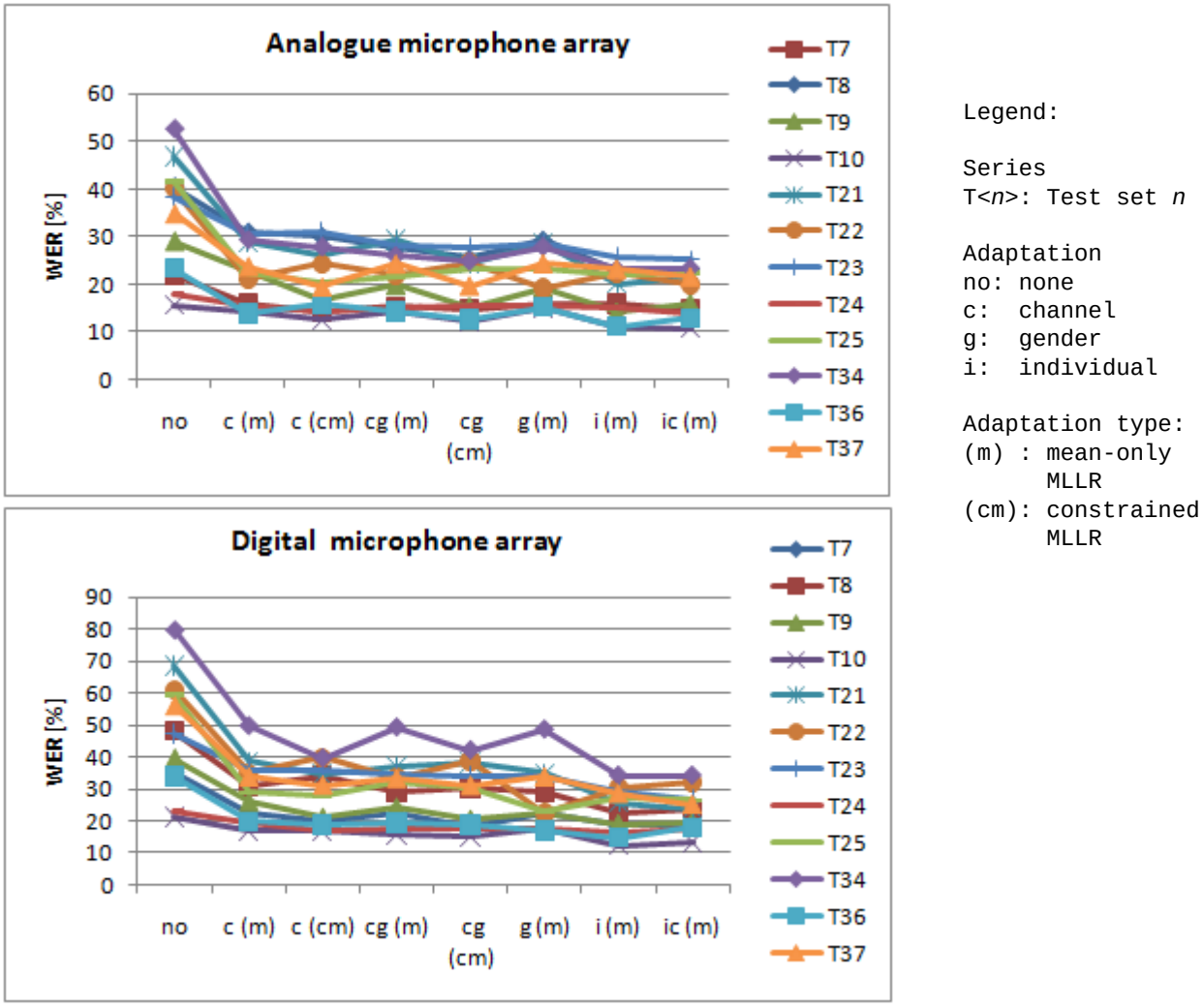
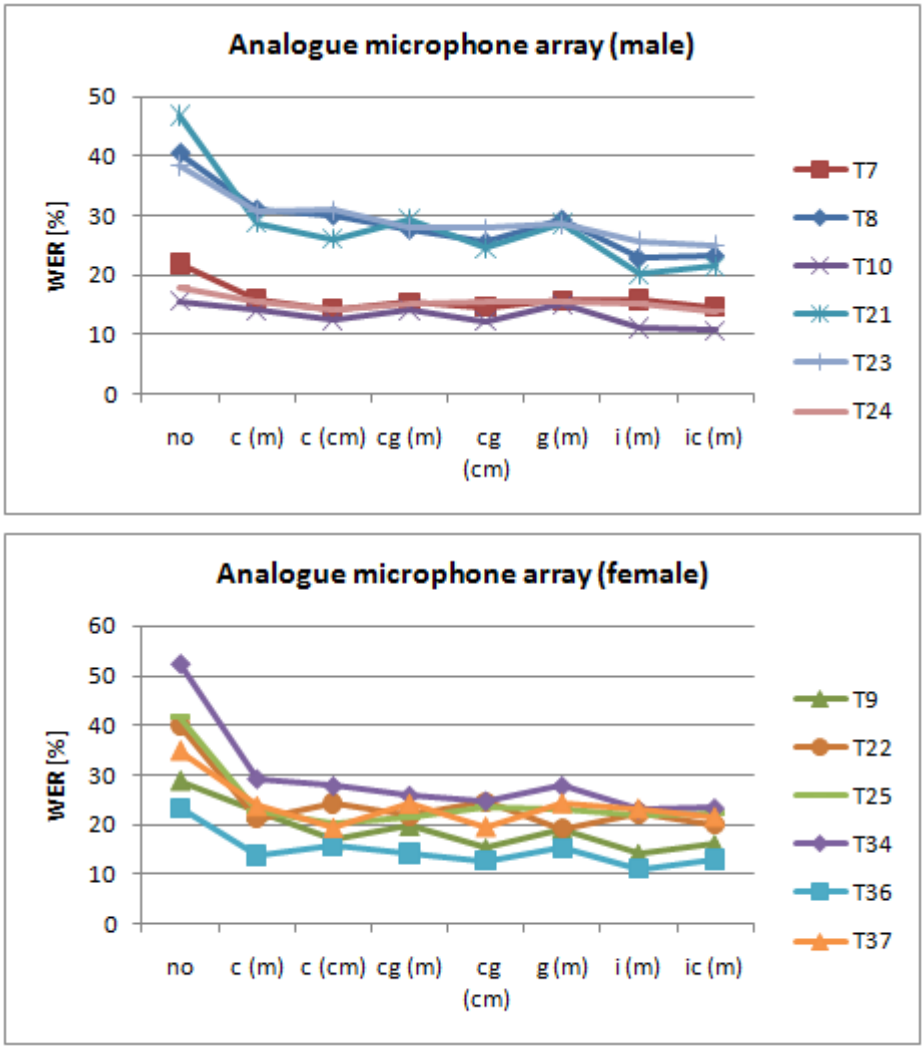


Figure 51 WER vs. adaptation and channel

The WERs of both the analogue and digital microphone arrays show the same tendencies with differing adaptation techniques, as presented in Figure 51 above. The best results can be obtained by adaptation to the individual speaker.

In the following Figure 52 the effects of the channel and gender are analysed in detail.




```
Legend:

Series
Tn: Test set n

Adaptation
no: none
c:  channel
g:  gender
i:  individual

Adaptation type:
(m) : mean-only
      MLLR
(cm): constrained
      MLLR
```

Figure 52 WER vs. adaptation and gender (analogue microphone array)

A special effect was observed early in the process of running the ASR. It appears that certain participants naturally achieve good WERs while others do not. This can be clearly seen in Figure 52 above. Half of the male participants achieved quite good WERs from the very beginning, while the other half achieved WERs which were only half as good. The effect of the different adaptation techniques is increased if the initial WER is not high. However, while WERs of less than 15% were achieved with participants T7, T10 and T24, the WERs of participants T8, T21 and T24 were always above 20%.

The distributions of the female participants, on the other hand, are similar and do not show two classes. Please also note that female speakers generally have poorer starting WERs which is believed to be due to the database being predominantly trained on male speakers.

The digital microphone array shows the same behaviour, as presented in Figure 53 below.

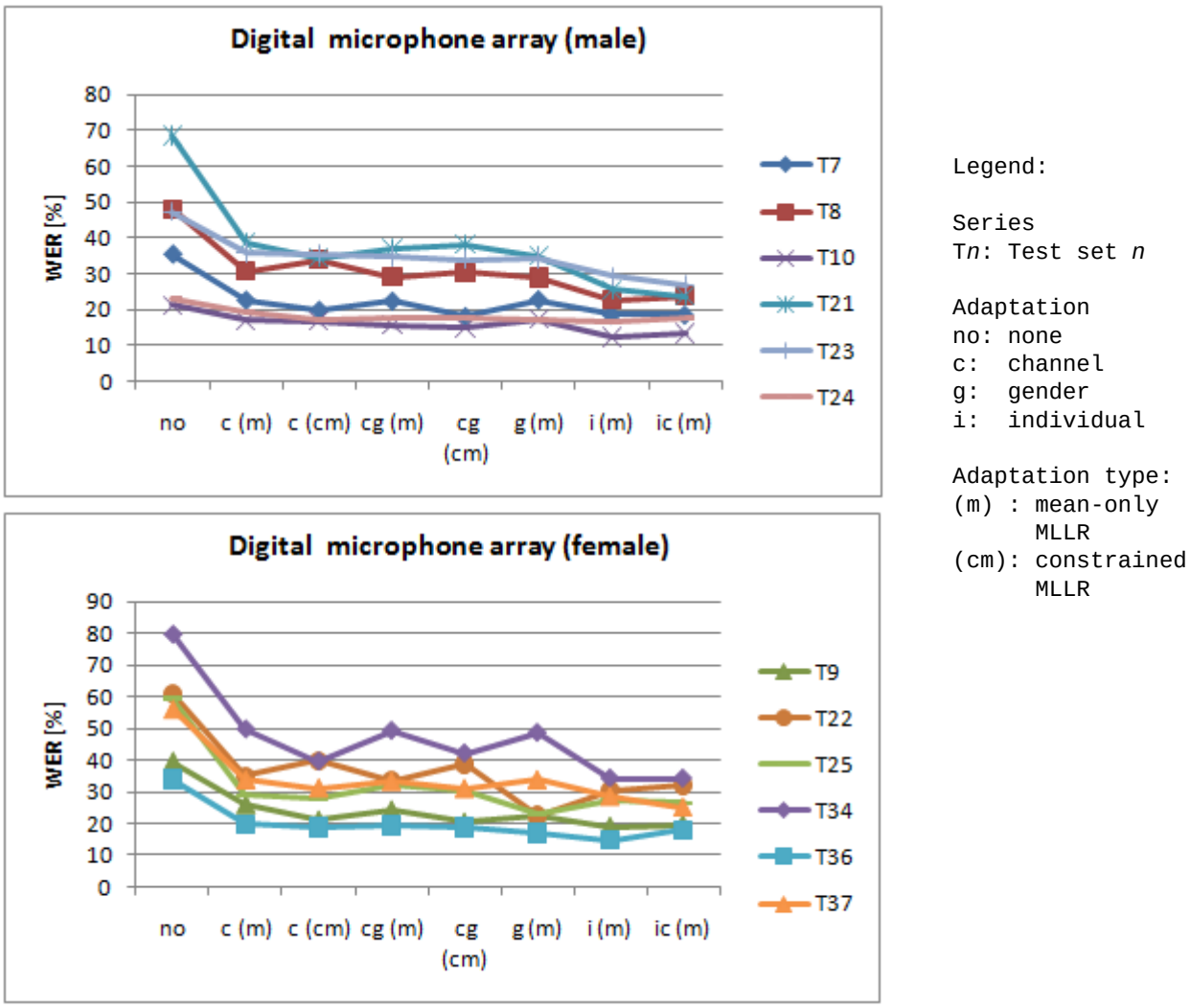
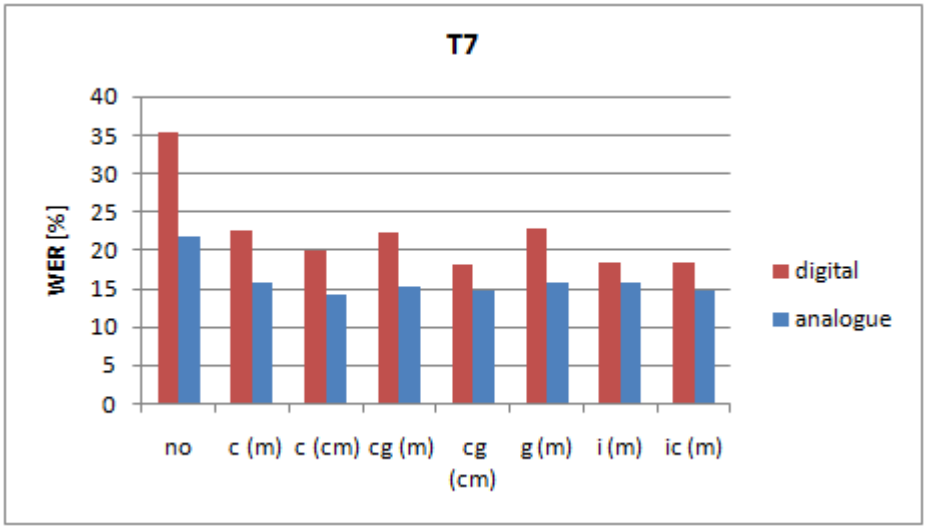


Figure 53 WER vs. adaptation and gender (digital microphone array)

The effects of the WERs vs. the channel type and gender of the individual, as discussed above, are also present with the digital microphone array, although more “blurred” (see Figure 53). WERs measured from utterances recorded by the digital microphone array are generally higher, but the gap between the analogue and digital array closes when the different adaptation techniques are applied.

The WERs for the different adaptation techniques are looked at in greater detail on two average individuals, one male and one female, in the following Figure 54 and Figure 55.




```
Legend:

Series
T<n>: Test set n

Adaptation
no: none
c:  channel
g:  gender
i:  individual

Adaptation type:
(m) : mean-only
      MLLR
(cm): constrained
      MLLR
```

Figure 54 T7: WER vs. adaptation (male)

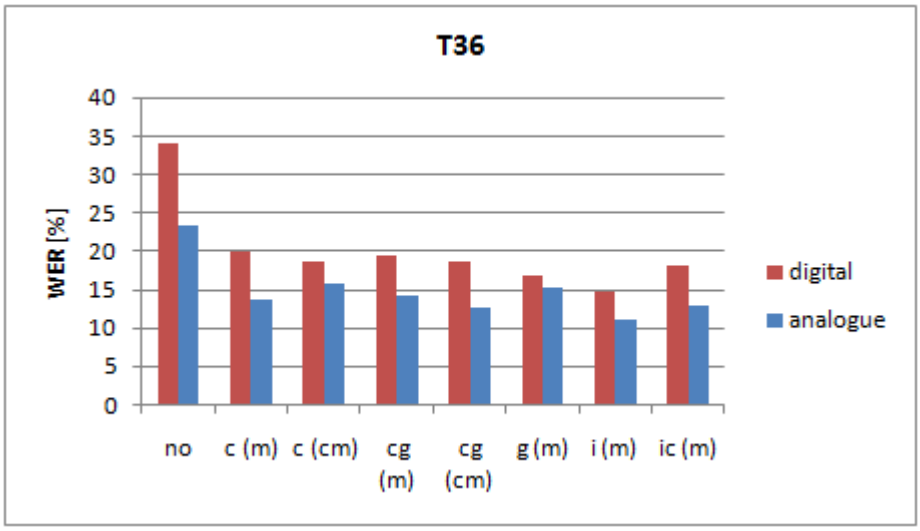
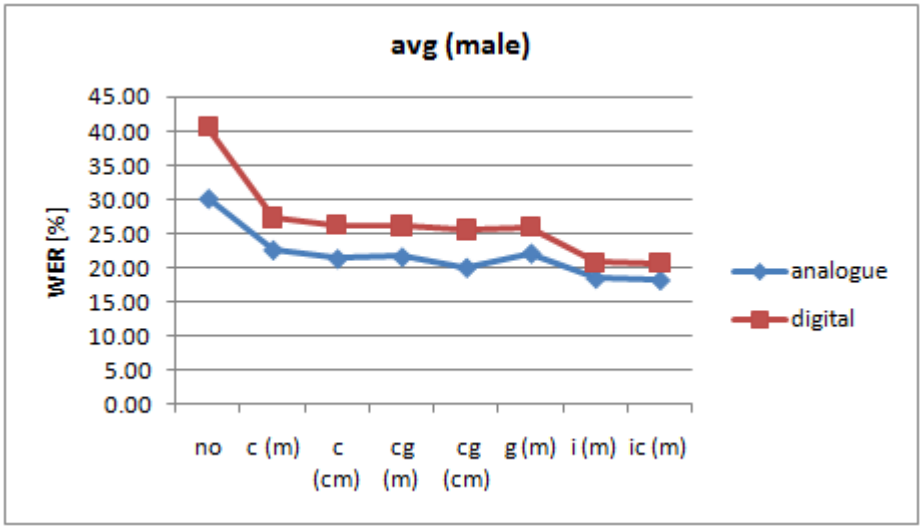


Figure 55 T36: WER vs. adaptation (female)

Both the analogue and digital microphone arrays display the same pattern with respect to adaptation technique. Applying constrained MLLR gives better results compared to means-only MLLR and adaptation to the individual speaker again gives better results compared to gender or channel (or both) adaptation.

Next, the average WERs are looked at in greater detail. The results are presented in Figure 56, Figure 57 and Figure 58 below.




```
Legend:

Series
T<n>: Test set n

Adaptation
no: none
c:  channel
g:  gender
i:  individual

Adaptation type:
(m) : mean-only
      MLLR
(cm): constrained
      MLLR
```

Figure 56 Average WER vs. adaptation (males only)

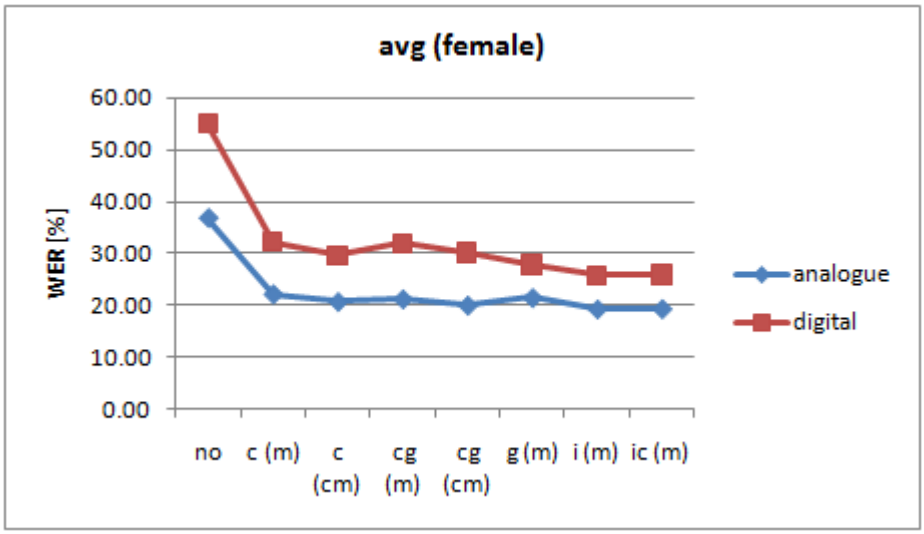
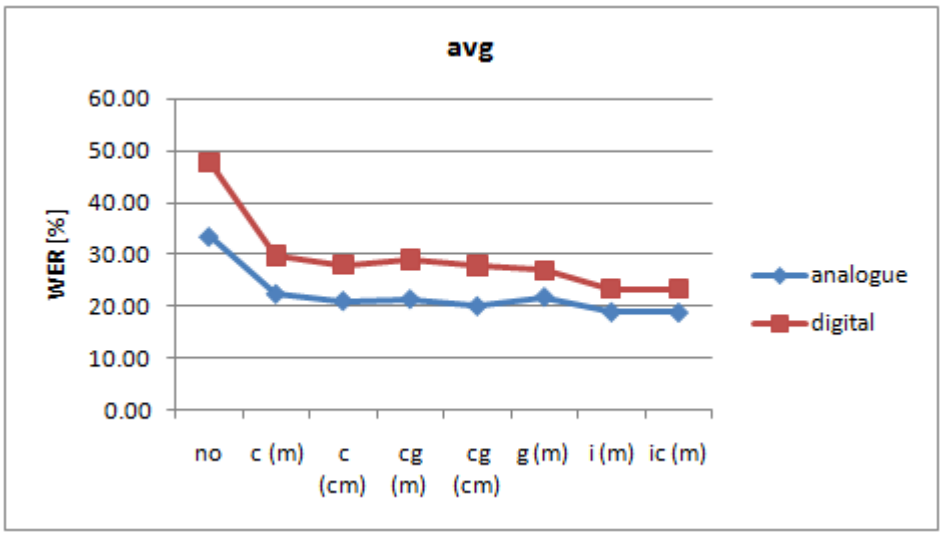


Figure 57 Average WER vs. adaptation (females only)




```
Legend:

Series
T<n>: Test set n

Adaptation
no: none
c:  channel
g:  gender
i:  individual

Adaptation type:
(m) : mean-only
      MLLR
(cm): constrained
      MLLR
```

Figure 58 Average WER vs. adaptation

Using averaged WERs for the two genders (Figure 56 and Figure 57) and all participants (Figure 58) clearly improved WERs are achieved when the adaptation type is more tailored towards the individual, i.e. the more closely the ASR system is adapted to the individual speaker and situation, the better the WER that can be achieved. Note that the effect of improved WERs using constrained MLLR compared to means-only MLLR (as observed in Figure 52 above) is cancelled out when considering the average values.

Discussion

This analysis of ASR system WERs in relation to the adaptation technique, channel, gender and individual participant shows that the newly designed digital microphone array compares well with the analogue one. Specific findings of the analysis are now looked at in greater detail. First, it was observed that certain participants achieve very low WERs without using adaptation, independent of gender. The WERs drop further after applying the different adaptation techniques. There is no obvious reason for this; there are no acoustical differences which can be distinguished nor had these individuals been used for initial training of the HMMs. The clear split in the WERs (see Figure 52) of T8, T21 and T23 (high WERs) vs. T7, T10 and T24 (low WERs) cannot, for example, be attributed to the sample rate as only T10 and T24 were recorded at 44.1 kHz. In addition, T36 (female) achieves excellent WERs though recorded at 44.1 kHz. It can be concluded that no effect from the sample rate could be observed and that there is no obvious reason why certain individuals perform better than others.

Looking at the distribution of the WERs vs. participant gender, no correlation can be found either. The WERs of male and female participants and analogue and digital recordings are spread evenly (Figure 51, Figure 52 and Figure 53).

As mentioned in section “Prompter” on page 78, building works were carried out during the recording of participant T36. These banging noises did not have a detrimental effect on the WERs measured, as shown in Figure 55. Most participants, as reported in “Prompter” on page 78, also clicked into their recordings. These mouse clicks were removed if possible. The click noises affected all participants equally and will have led to a general shift of all the data, insofar as the post-processing of the features did not remove these noises (see “Beamforming and speech enhancement” on page 69 for details).

During the alignment process it was found that recording data had been lost for the analogue microphone array (see Table 4 for details). Only the utterances from the adaptation set were analysed in detail. The test set sentences were listened to during data preparation (see “Data preparation” on page 80) but they were not checked for accuracy. While these utterances sounded generally correct, repeats, hesitations, stutters and other effects were observed. If the participant did not repeat an utterance with a mistake, then it was still used for the ASR process. For this dissertation it was considered more important to have as much data as possible to compare the analogue and digital microphone arrays than to remove any utterances which were not accurate. The two arrays are tested with the same data so that they are subject to the same effects. If this data is to be made public accuracy tests would need to be carried out.

Another feature of the individual participants is speed. Some people spoke quite fast, while others had a slower tempo. T25 and T34 were slow speakers. T36 and T8 were very fast speakers, while T37 and T7 spoke quite fast. Looking at Figure 52 and Figure 53, speed does not appear to affect the WERs. T34 in particular - a speech professional, separated the words as if to help the ASR. This habit, compared to the fast speakers T8 and T36, certainly did not produce any improvement on the WER.

Test scenarios ① and ② looked at the effect of constrained MLLR vs. means-only MLLR. It is expected that constrained MLLR leads to improved WERs, as the HMMs are better tailored to the individual. While the individual graphs of the WERs vs. adaptation techniques indeed indicate such a positive effect (Figure 51) the average WERs (Figure 58) show no significant effect.

No analysis of the 20k data has been carried out. Generating the 20k vocabulary results would take ten times longer than running the 5k test and was not feasible for this project. Initial WERs indicate the same behaviour as was observed for the 5k test, though significantly higher levels (see Table 5 and Table 6).

The performance of an ASR system improves with the type of adaptation so that the more a system is adapted to an individual speaker and situation the better are the WERs. This behaviour is as expected and rather undesirable because a system which as very tailored to any individual is inevitably inflexible. Designing a best possible ASR/DSR system will always be a trade-off of different factors.

As the digital microphone array system has a highly increased SNR but only marginally reduced WER performance compared to its analogue competitor, and that it can be made at a fraction of the cost, it is a viable option for distant speech recognition (DSR) in instrumented meeting rooms (IMR). Commercial products using digital MEMS microphones can be expected in the market within months.

Conclusion

This dissertation reports on the first successful implementation of a digital MEMS microphone array and its performance in comparison with an analogue microphone array. The two systems achieve similar word error rates in an automated speech recognition system.

An array comprising eight microphones was built with samples of Knowles digital MEMS microphones. The PDM output of these microphones was downsampled from 64 f_s to f_s using a DSP implemented on a Xilinx FPGA. This DSP was designed in Verilog HDL with the aid of a digital filter design tool and an FPGA core generator. The interface from the DSP to the recording SW was built using an off-the-shelf USB streaming controller that collects the audio data from the DSP via the AC'97 interface and sends it to the PC over the USB.
It was not possible to find a current operating system capable of capturing eight channels of USB audio data. A backup scheme was therefore designed and implemented using TDM to transfer the eight channels of audio sampled at 16 kHz in a 48 kHz frame. Efficient post-processing scripts were subsequently implemented to generate an eight-channel audio file.

The newly built digital MEMS microphone array was evaluated and compared with an existing analogue microphone array. An experiment was run in which twelve participants were asked to speak WSJ sentences which were then processed in an automatic speech recognition (ASR) system. The recordings were made in an instrumented meeting room. Distant speech recognition (DSR) was performed using a database made available by the CSTR. The database is built from the WSJCAM0 corpus using speech recorded with close-talking microphones. After initial WER measurements MLLR adaptation was also performed for evaluating the analogue and digital microphone arrays. Adaptation to channel, gender and individual speaker (and combinations of these) were performed.
Overall the digital microphone array compares well with the analogue one as shown in Table 12 below.

Table 12: Average WERs of analogue and digital microphone arrays

Adaptation	MLLR	WER [%] Microphone array		Δ
		Analogue	Digital	
None	n.a.	33.56	47.86	14.3
Channel	means-only	22.41	29.78	7.37
Channel	constrained	21.04	27.99	6.95
Channel and gender	means-only	21.42	29.03	7.6
Channel and gender	constrained	20.03	27.89	7.86
Gender	constrained	21.81	26.91	5.1
Individual	means-only	18.86	23.26	4.4
Individual and channel	means-only	18.77	23.29	4.52

The average WER of the digital microphone array is less then 5% above that of the analogue microphone array if adaptation to the individual speaker is used with the analogue array producing WERs of less than 19% and the digital one WERs of less than 23.5%.
Adaption to gender, channel or both gives WERs less than 30% for the digital array and less than 22.5% for the analogue array with the difference between the two being less then 8%.

The digital MEMS microphone array designed, tested and evaluated in this project is ideally suited to meeting the demands of audio recording for distant speech recognition in future instrumented meeting rooms (IMR) due to being easily portable and very cost efficient.

The future might well see the emergence of compact disc-sized microphone arrays which can easily be used for travelling. At a later date microphone array networks might well be found in every IMR which use the meeting participants' mobile phones and a host processor which collects the data.

Future Work

A project such as the one presented in this dissertation is as much a feasibility study as the development of a product. Indeed, it was only the recent emergence of MEMS microphones with direct digital PDM outputs that enabled the design of the digital MEMS microphone array (DMA). It can therefore be considered as a technology demonstrator. The following list of future work gives an idea of how the current system could be expanded and improved.

- DMA system: Due to the absence of a high-order PDM model for the digital MEMS microphone array no system simulations were carried out. For completeness, system simulations should be carried out to check the system's frequency response and DC (direct current) levels.
- DMA DSP: System simulations should also be undertaken to verify the correct dithering [36]
- DMA DSP: The current FPGA is utilised by about 60%. This gives the option of implementing advanced DSP features in HW, e.g. the beamformer, noise canceller or Wiener filters.
- DMA gain: The insertion of a gain block into the DSP revealed the presence of a DC offset. In theory, the DMA DSP should not have a DC offset. System simulations were not feasible due to the absence of a full simulation license. The Verilog HDL simulation ran at 0.1% of full capacity due to license restrictions.
- The current implementation of the DMA is not 100% AC'97 compliant due to the fact that the AC'97 specification has not been defined for this type of DMA and because certain aspects of the AC'97 specifications were not implemented so far.
- The AC'97 specification defines three resets: a reset pin, a power-on reset and a SW reset. Although this reset architecture was designed, this circuit was bypassed due to limitations in the FPGA.
- Many MEMS microphones support calibration [84]. Using the AC'97 interface (register control) and the flexibility of an FPGA this calibration could be utilised for different microphone brands.
- Although both MS and Linux OSs claim to support microphone arrays, in reality MS officially only supports up to four channels and it was only possible to achieve seven working channels using Linux. This is the most unfortunate limitation of the current digital MEMS microphone array, preventing it from being the perfect technology demonstrator.

- If the database of speech that was recorded is to be publicly available for future development then all utterances need to be checked for correctness and any inaccurate ones removed.
- ASR: Further investigate WER gap between analogue and digital microphones. Is this due to lack of adaptation data, or is it inherent due to the lower SNR of the digital MEMS microphones?
- DMA: Improve DMA SNR by optimising MEMS microphone clocking (e.g. signal termination, shielding, layout).
- DMA: Improve DMA SNR by optimising DSP performance, i.e. CIC and FIR filter specification
- ASR: Investigate and compare DMA performance with analogue array using spontaneous speech.

References

Books

[1] Axelson J., “USB Complete: Everything You Need to Develop Custom USB Peripherals”, 3rd edition, 2005, ISBN-13: 978-1931448024,

[2] Bitzer J. and Simmer K.U., “Superdirective Microphone Arrays”, in Brandstein M. and Ward D. (Eds.), “Microphone Arrays”, Springer Verlag, New York, 2001, ISBN: 3540419535

[3] Cianci E., Foglieatti V., Minotti A., Caroint A., Caliano G. and Pappalardo M., “Fabrication Techniques in Micromachined Capacitive Ultrasonic Transducers and their Application, chapter 2, in Leondes C.T. (Editor), “MEMS/NEMS: handbook techniques and applications”, Vol.2, Springer (September 6, 2006), ISBN-13: 978-0387245201

[4] Hitek UK. Ltd., “C51 Primer - An Introduction To The Use Of The Keil C51 Compiler On The 8051 Family”, Hitex (UK) Ltd., University of Warwick Science Park, Coventry, CV4 7EZ, Issue III, 2004

[5] Ladefoged P., “Elements of Acoustic Phonetics”, Chicago University Press; 2nd Revised edition (7 Dec 1995), ISBN-13: 978-0226467641

[6] Lyons R.G., “Finite Impulse Response Filter”, in chapter 5, “Understanding Digital Signal Processing”, Addison-Wesley Publishing Company, 1997, ISBN 0-201634678

[7] Simmer K.U., Bitzer J. and Marro C., “Post-Filtering Techniques”, in Brandstein M. and Ward D. (Eds.), “Microphone Arrays”, Springer Verlag, New York, 2001, ISBN: 3540419535

[8] Wölfel M. and McDonough J.A., “Distant Speech Recognition”, Wiley Blackwell, New edition (17 April 2009), ISBN-13: 978-0470517048

[9] Zha F.X., “Manufacturing Advisory Service System for Concurrent and Collaborative Design of MEMS devices”, chapter 1, in Leondes C.T. (Editor), “MEMS/NEMS: handbook techniques and applications”, Vol.1, Springer (September 6, 2006), ISBN-13: 978-0387245201

[10] Zhang T., “Teach Yourself C in 24 Hours”, Second Edition, 2000 by Sams Publishing, ISBN 0-672-31861-x

Datasheets

[11] Keil Software, “Getting Started with µVision2 and the C51 Microcontroller Development Tools, User’s Guide 02.2001

[12] Knowles Acoustics SPM0205HD4 Digital Microphone, datasheet received via personal confidential e-mail July 2009

[13] Leap Electronics Co., Ltd., “Leaper-48 Handy Universal Writer”, document retrieved July 2009, URL http://www.leap.com.tw/english/pdf/lp-48.pdf

[14] Nguyen H., “EVM AC97 FIRMWARE - MOD CODE”, Revision 1.0, Texas Instruments Incorporated

[15] Sennheisser MKE 2-P-C microphones, retrieved July 2009, URL http://www.sennheiser.co.uk/uk/home_en.nsf/root/professional_wired-microphones_lavalier-mics_004224

[16] Texas Instruments, TUSB3200A USB Streaming Controller (STC), Data Manual SLE”1008, 2001

[17] Xilinx Corp. DS512, March 2008, “Block Memory Generator v2.7”, document retrieved July 2009, URL http://www.xilinx.com

[18] Xilinx Corp. DS613, Oct. 2007, “CIC Compiler v1.0”, document retrieved July 2009, URL http://www.xilinx.com

[19] Xilinx Corp. DS534, Oct. 2007, “FIR Compiler v3.2”, document retrieved July 2009, URL http://www.xilinx.com

[20] Xilinx Corp. DS529, Mar. 2009, “Spartan-3A FPGA Family: Data Sheet”, document retrieved July 2009, URL http://www.xilinx.com

Papers

[21] AMI 2006, “The Future of Business Meetings - Applications for AMI Technologies”, retrieved Aug. 2009, URL http://www.amiproject.org/pdf/Applications-for-AMI-Technologies.pdf/view?searchterm=None

[22] Beis U., June 2008, An Introduction to Delta Sigma Converters, document retrieved July 2009, URL http://www.beis.de/Elektronik/DeltaSigma/DeltaSigma.html

[23] Brauer M., Dehé A., Bever T., Barzen S., Schmitt S., Füldner M. and Aigner R., “Silicon microphone based on surface and bulk micromachining”, J. Micromech. Microeng. 11 (2001) 319-322

[24] Callias F., Salchli F.H. and Girard, D., “A set of four ICs in CMOS technology for a programmable hearing aid”, IEEE Journal of Solid-State Circuits, April 1989, Vol. 24, Issue 2

[25] van Compernolle D., Ma W., Xie F. and van Diest M., “Speech Recognition in Noisy Environments with the Aid of Microphone Arrays”, Speech Communication 9 (1990) 433-442

[26] Cox H., Zeskind R.M. and Kooij T., “Practical Supergain”, IEEE Transactions on Acoustic, Speech, and Signal Processing, Vol. ASSP-34, No. 3, June 1986

[27] Cox H., Zeskind R.M. and Owen M.M., “Robust Adaptive Beamforming”, IEEE Transactions on Acoustic, Speech, and Signal Processing, Vol. ASSP-35, No. 10, October 1987

[28] Cummings C., “Simulation and Synthesis Techniques for Asynchronous FIFO Design with Asynchronous Pointer Comparisons, retrieved July 2009
URL http://www.sunburst-design.com/papers/CummingsSNUG2002SJ_FIFO2.pdf

[29] Elko G.W. and Meyer J., “Microphone Arrays”, in Benesto J., Sondhi M.M. and Huang Y.A. (Editors), “Springer Handbook of Speech Processing”, chapter 50, Springer 2008, ISBN-13: 978-3540491255

[30] Flanagan J.L., Johnson R., Zahn R. and Elko G.W., “Computer-steered microphone arrays for sound transduction in large rooms”, Journal of the Acoustical Society of America 78 (5), November 1985

[31] Fransen J., Pye D., Robinson T., Woodland P and Young S., “WSJCAM0 Corpus and Recording Description”, 2nd September 1994, document retrieved July 2009,
URL http://www.cstr.ed.ac.uk archives

[32] Fügen C., Wölfel M., McDonough J.W., Ikbal S., Kraft F., Laskowski M., Stücker S. and Kuatani K., “Advances in Lecture Recognition: The ISL RT-06S Evaluation System”, ICSLP Interspeech 2006

[33] Giuliani D, Omolongo M. and Svaizer P., “Experiments of Speech Recognition in a Noisy and Reverberant Environment using a Microphone Array and HMM Adaption”, ICSLP96

[34] Hain T., Burget L., Dines J., Garau G., Karafiat M., Lincoln M., McCowan I., Moore D., Wan V., Ordelman R and Renals S., “The AMI System for the Transcription of Speech in Meetings”, in Renals S. and Bengio S (Eds.), MLMI 2005, LNCS 3869, pp.450-462, 2006, Springer Verlag Berlin Heidelberg 2006

[35] Hain T., Burget L., Dines J., Garau G., Karafiat M., Lincoln M., Vepa J. and Wan V., "The AMI Meeting Transcription System: Progress and Performance", in Renals S. and Bengio S (Eds.), MLMI 2005, LNCS 3869, pp.419-431, 2006, Springer Verlag Berlin Heidelberg 2006

[36] Hicks C., "The Application of Dither and Noise-Shaping to Nyquist-Rage Digital Audio: an Introduction", 12th September 1995, document retrieved July 2009, URL http://www.digitalsignallabs.com/noiseb.ps

[37] Himawan I., McCowan I. and Lincoln M., "Microphone Array Beamforming Approach to Blind Speech Separation", MLMI 2007, LNCS 4892, pp. 295-305, 2007

[38] Hogenauer E.B., "An Economical Class of Digital Filters for Decimation and Interpolation", IEEE Transactions on Acoustic, Speech, and Signal Processing, Vol. ASSP-29, No. 2, April 1981

[39] Huang Y., Benesty J., Chen J., "Dereverberation", in Benesto J., Sondhi M.M. and Huang Y.A. (Editors), "Springer Handbook of Speech Processing", chapter 46, Springer 2008, ISBN-13: 978-3540491255

[40] Juang B.H., "Speech Recognition in adverse environment", Computer speech & language, Vol. 5, No3, pp. 275-294

[41] Kiyohara K., Kaneda Y., Takahashi S., Nomura H. and Kijima J., "A microphone array system for speech recognition", IEEE International Conference on Acoustics, Speech, and Signal Processing, 1997, ICASSP-97

[42] Knapp C.H. and Carter G.C., "The Generalised Correlation Method for Estimation of Teim Delay", IEEE Transaction on Acoustic, Speech and Signal Processing, Vol. ASSP-24, No.4, August 1976

[43] Lincoln M., McCowan I., Vepa J. and Maganti H.K., "The Multi-Channel Wall Street Journal Audio Visual Corpus (MS-WSJ-AV): Specification and Initial Experiments", IEEE Workshop on Automatic Speech Recognition and Understanding, 2005

[44] Malcolm G., "Dummy's Guide to HD Audio & UAA", Wolfson Microelectronics plc, Revision 0.4, 25th June 2007, Confidential

[45] McCowan I., "Microphone Arrays: A Tutorial", April 2001, document retrieved July 2009, URL http://www.idiap.ch/~mccowan/arrays/tutorial.pdf

[46] McCowan I., "mdm-tools: Multiple Distant Microphone Toolkit", 5th May 2005, document retrieved July 2009, URL http://www.cstr.ed.ac.uk archives

[47] McCowan I., Carletta J., Kraaij W., Ashby S., Bourban S., Flynn M., Guillemot M., Hain T., Kadlec J., Karaiskos V., Kronenthal M., Lathoud G., Lincoln M., Lisowska A., Post W., Reidsma D. and Wellner P., "The AMI Meeting Corpus", Proceedings of Measuring Behavior 2005, the 5th International Conference on Methods and Techniques in Behavioral Research

[48] McDonough J. and Wölfel M., "Distant Speech Recognition: Bridging the Gaps", Proceedings of HSCMA, Trento, Italy, 2008

[49] McDonough J. and Wölfel M., "Distant Speech Recognition: No Black Boxes Allowed", Proc. of ITG-Fachtagung, Aachen, Germany, 2008

[50] McDonough J., Kumatani K., Gehrig T., Stoimenov E., Mayer U., Schacht S., Wölfel M. and Klakov D., "To Separate Speech! A System for Recognising Simultaneous Speech", Interspeech 2006

[51] Moore D.C. and McCowan I., "Microphone Array Speech Recognition: Experiment on Overlapping Speech in Meetings", ICASSP 2003

[52] Morgan N., Baron D., Edwards J., Ellis D., Gelbart D., Janin A., Pfau T., Shriberg E. and Stolcke A., "The Meeting Project at ICSI", Proc. of the Human Language Technology Conference, 2001

[53] Neumann J.J. and Gabriel K.J., "A fully-integrated CMOS-MEMS audio microphone", Transducers '03

[54] Neumann J.J. and Gabriel K.J., "CMOS-MEMS membrane for audio-frequency acoustic actuation", Sensors and Actuators A 95 (2002) 175-182

[55] Ning Y.B., Mitchell A.W. and Tait R.N., "Fabrication of a silicon micromachined capacitive microphone using a dry-etch process", Sensors and Actuators A 53(1996) 237-242

[56] Paul D.B. and Baker J.M., "The Design for the Wall Street Journal-based CSR Corpus", Proceedings of the workshop on Speech and Natural Language, 1992, pp. 357-362 ISBN:1-55860-272-0

[57] Park S., Motorola Inc., "Principles of Sigma-Delta Modulation for Analog-to-Digital Converters", document retrieved July 2009,
URL http://digitalsignallabs.com/SigmaDelta.pdf

[58] Pederson M., Olthuis W. and Berveld P., "An integrated silicon capacitive microphone with frequency-modulated digital output", Sensor and Actuators A 69 (1998) 267-275

[59] Rabiner L.R. and Juang B.H., “An Introduction to Hidden Markov Models”, IEEE ASSP Magazine January 1986

[60] Renals S., Hain T. and Bourlard H., “Recognition and Understanding of Meetings in the AMI and AMIDA projects”, Automatic Speech Recognition and Understanding, 2007, retrieved July 2009, URL http://www.cstr.inf.ed.ac.uk

[61] Robinson T., Fransen J., Pye D., Foot J and Renals S., “WSJCAM0: A British English Speech Corpus for Large Vocabulary Continuous Speech Recognition”, in Proc. IEEE ICASSP, Detroit, 1995, pp. 81–84

[62] Royer M., Holmen J.O., Wurm M.A., Aadland OS. and Glenn M., “ZnO on Si integrated acoustic sensor”, Sensors and Actuators, 4 (1983) 357-362

[63] Scheeper P.R., van der Donk A.G.H., Olthuis W. and Bergveld P., “A review of silicon microphones”, Sensors and Actuators A 44 (1994) 1-11

[64] Schuller B., Wöllmer M., Moosmayr T. and Rigoll G., “Recognition of Noisy Speech: A Comparative Survey of Robust Model Architectures and Feature Enhancement”, Hindawi Publishing Corporation, EURASIP Journal on Audio, Speech, and Music Processing, Volume 2009, Article ID 942617, 17 pages

[65] Seltzer M.L., Raj B. and Stern R.M., “Speech recognizer-based microphone array processing for robust hands-free speech recognition”, IEEE International Conference on Acoustics, Speech, and Signal Processing, 2002. ICASSP02

[66] Seltzer M.J. and Ray B., “Speech-Recognizer-Based Filter Optimization for Microphone Array Processing”, IEEE Signal Processing Letters, Vol.10, No.3, March 2003

[67] Shriberg E., Stolcke A and Baron D., “Observations on Overlap: Finding and Implementation for Automatic Processing of Multi-Party Conversation”, ICSLP Eurospeech 2001

[68] Shriberg E., Dhillon R., Bhagat S., Ang J. and Carvey H., “The ICSI Meeting Recorder Dialog Act (MRDA) Corpus”, Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue at HLT-NAACL 2004

[69] Stern R.M., Liu F.-H., Ohshima Y., Sullivan T.M. and Acero A., “Multiple Approaches to Robust Speech Recognition”, Second International Conference on Spoken Language, (ICSLP'92) Banff, Alberta, Canada, October 13-16, 1992

[70] Stewart R.W. and Pfann E., “Oversampling and sigma-delta strategies for data conversion”, Conference Record of the Thirty-Second Asilomar Conference on Signals, Systems & Computers, 1998

[71] Weigold J.W., Brosnihan T.J., Bergeron J. and Zhang X., “A MEMS Condenser Microphone For Consumer Applications”, MEMS 2006

[72] Wölfel M., Fügen C., Ikbal S. and McDonough J.,”Multi-Source Far-Distance Microphone Selection and Combination for Automatic Transcription of Lectures”, ICSLP Interspeech 2006

[73] Woodland P.C., “Speaker Adaptation for Continuous Density HMMS, A Review”, ITRW on Adaptation Methods for Speech Recognition August 29-30, 2001, Sophia Antipolis, France

[74] Yamada T., Nakamura S. and Shikano K., “Robust speech recognition with speaker localization by a microphone array”, Fourth International Conference on Spoken Language, 1996, ICSLP 96, 3-6 Oct. 1996, Vol. 3, pp. 1317-1320

[75] Yates R., “Practical Considerations in Fixed-Point FIR Filter Implications”, 27th March 2007, document retrieved July 2009,
URL http://www.digitalsignallabs.com/fir.pdf

[76] Yates R., “Fixed-Point Arithmetic: An Introduction”, 7th July 2009, retrieved July 2009, URL http://www.digitalsignallabs.com/fp.pdf

[77] Young S., “HMMs and Related Speech Recognition Technologies”, in Benesto J., Sondhi M.M. and Huang Y.A. (Editors), “Springer Handbook of Speech Processing”, chapter 27, Springer 2008, ISBN-13: 978-3540491255

[78] Yurish S.Y., Kirianaki N.V. and Myshkin I.L., World Sensors and MEMS markets: Analysis and TRends”, Sensors & Transducers Magazine, Vol.62, Issue 12, Dec. 2005, ISSN 1726-5479 S

[79] Zwyssig E., “Low Power Digital Filter Design for Hearing Aid Applications”, Dissertation for the degree of a MSc by Research, October 2000, The University of Edinburgh

[80] Zwyssig E.P., Erdogan A.T., Arslan T., “Low power system on chip implementation scheme of digital filtering cores”, IEE Seminar on Low Power IC Design (Ref. No. 2001/042), January 2001, pp. 5/1 - 5/9

Patents

[81] Chen L.-T., Chu C.-H. and Cheng W.-H., “MEMS microphone module and manufacturing thereof”, U.S. Patent Application 20090129622, 18th March 2009

[82] Laming R.I. and Traynor A., “MEMS device”, U.S. Patent Application 20090152655, 18th June 2009

[83] Mian M., Drury R. and Hopper P.J., “MEMS microphone”, U.S. Patent US 7301212 B1, 27th Nov. 2007

[84] Poulsen J.K., Fallesen C., Stenberg L.J., Bosch J.J.G., “Calibrated Micromechanical Microphone”, U.S. Patent US 2008/0075306 A1, 27th March 2008

Specifications/Standards

[85] Intel, AC97 Audio Codec Specification, document retrieved July 2009, URL http://download.intel.com/support/motherboards/desktop/sb/ac97_r23.pdf

[86] Firewire, IEEE 1394-xxxx Standard for a High-Performance Serial Bus, retrieved July 2009, URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&isnumber=4659232&arnumber=4659233&punumber=4659231

[87] HDA, Intel High Definition Audio Specification, retrieved July 2009, URL http://www.intel.com/standards/hdaudio/

[88] I^2S, I2S, or Inter-IC Sound, or Integrated Interchip Sound, electrical serial bus interface standard, document retrieved July 2009, URL http://www.nxp.com/acrobat_download/various/I2SBUS.pdf

[89] SLIMbus, MIPI Alliance Specification for Serial Low-power Inter-chip Media Bus (SLIMbus), document not generally available, retrieved July 2009, URL http://www.mipi.org/specs/index.shtml

[90] S/PDIF, Sony/Philips Digital Interconnect Format (more commonly known as Sony Philips Digital InterFace), defined by IEC 60958 (often referred to as AES/EBU), known as IEC 60958 type II, URL http://www.iec.ch/cgi-bin/procgi.pl/www/iecwww.p?wwwlang=E&wwwprog=cat-det.p&progdb=db1&wartnum=036136

[91] UAA: Universal Audio Architecture, August 5, 2005, retrieved Aug. 2009, URL http://www.microsoft.com/whdc/device/audio/uaa.mspx

[92] Universal Serial Bus Specification Version 1.1, document retrieved July 2009, URL http://www.usb.org/developers/docs/

[93] Universal Serial Bus Device Class Specification for Audio Devices, Release 1.0, March 18, 1998, document retrieved July 2009, URL

Webpages

[94] 8052.com, retrieved July 2009, URL http://www.8052.com/

[95] Akustica Launches World's First CMOS MEMS Microphone as Single-Chip Device, Monterey, CA/Globalpress, February 27, 2006, retrieved Aug. 2009, URL http://www.embedded-computing.com/news/db/?2070

[96] Aliasing, retrieved July 2009, URL http://en.wikipedia.org/wiki/Aliasing

[97] The AMI/AMIDA project, retrieved August 2009, URL http://www.amiproject.org/

[98] Analog Devices, “iMEMS Microphone”, July 2009, retrieved July 2009, URL http://www.analog.com/en/audiovideo-products/imems-microphone/products/index.html

[99] Analog-to-Digital Converter, retrieved July 2009, URL http://en.wikipedia.org/wiki/Analog-to-digital_converter

[100] Audacity, retrieved July 2009, URL http://audacity.sourceforge.net/

[101] Autocorrelation, retrieved July 2009, URL http://en.wikipedia.org/wiki/Autocorrelation

[102] Akustica, “What is CMOS MEMS?”, 2007, retrieved July 2009, URL http://www.akustica.com/technology/whatis.asp

[103] Bidule, document July 2009, URL http://www.plogue.com/index.php?option=com_frontpage&Itemid=1

[104] ch_wave, retrieved Aug. 2009, URL http://festvox.org/docs/speech_tools-1.2.0/x444.htm

[105] Cross Correlation Function, retrieved July 2009, URL http://en.wikipedia.org/wiki/Cross-correlation_function

[106] Delta-sigma modulation, retrieved July 2009, URL http://en.wikipedia.org/wiki/Delta-sigma_modulation

[107] Dev/audio retrieved July 2009, URL http://www.dev-audio.com/dev-audio.html

[108] Donadio M.P., “CIC Filter Introduction”, 18th July 2000, document retrieved July 2009, URL http://users.snip.net/~donadio/cic.pdf

[109] Futurlec, “New High-Performance MEMS Microphones”, retrieved July 2009, URL http://www.futurlec.com/News/Analog/MEMS_Microphone.shtml

[110] EETimes, Johnson R.C., “TI embraces digital mics”, 11th Aug. 2007, retrieved July 2009, URL http://www.eetimes.com/showArticle.jhtml?articleID=202803858

[111] Eigenmike (em32) microphone array, retrieved Aug. 2009, URL http://www.mhacoustics.com/page/page/2949006.htm

[112] Electronicstalk, “MEMS microphones dominate new cellphone designs”, 11th March 2005, retrieved July 2009, URL http://www.electronicstalk.com/news/kow/kow108.html

[113] Electropages, “Wolfson - World's highest performing ultra-compact MEMS microphones”, 20th Oct. 2008, retrieved July 2009, URL http://www.electropages.com/viewArticle.aspx?intArticle=11752

[114] The HTK online manual, document retrieved July 2009, URL http://htk.eng.cam.ac.uk/

[115] LakeView Research, J. Axelson, “USB Complete”, retrieved July 2009, URL http://www.lvr.com/usbc.htm

[116] IEEE 1394 interface, retrieved July 2009, URL http://en.wikipedia.org/wiki/IEEE_1394_interface

[117] “Microphone Array Support in Windows Vista”, September 7, 2005, retrieved Aug. 2009, URL http://www.microsoft.com/whdc/device/audio/MicArrays.mspx

[118] “MEMS Microphones: A Global Technology, Industry and Market Analysis”, Innovative Research and Products (iRAP), Inc. 1st July 2007, Pub ID: IRAP1562165, retrieved Aug. 2009, URL http://www.marketresearch.com/product/display.asp?productid=1562165

[119] MOTU, 8pre, retrieved July 2009, URL http://www.motu.com/products/motuaudio/8pre/

[120] MOTU, 896mk3, retrieved July 2009, URL http://www.motu.com/products/motuaudio/896mk3/

[121] Pulse Density Modulation, retrieved July 2009, URL http://en.wikipedia.org/wiki/Pulse-density_modulation

[122] Python, retrieved Aug. 2009, URL: http://www.python.org/

[123] Rost A.-V., “1-bit A/D and D/A Converters”, 22nd June 2004, document retrieved July 2009, URL http://www.cs.tut.fi/sgn/arg/rosti/1-bit/

[124] Speech Separation Challenge Part II, retrieved July 2009, URL http://homepages.inf.ed.ac.uk/mlincol1/SSC2/evaluation.htm

[125] Technology Review, Green K., “Cheaper MEMS microphones”, 14th July 2006, retrieved July 2009, URL http://www.technologyreview.com/read_article.aspx?ch=specialsections&sc=personal&id=17170

[126] TI USB Streaming Controller, retrieved July 2009, URL http://focus.ti.com/docs/prod/folders/print/tusb3200a.html

[127] Tiny Akustica 1mm x 1mm MEMS Microphone Is Now Digital, Nov. 10, 2008, retrieved Aug. 2009, URL http://www.reuters.com/article/pressRelease/idUS155351+10-Nov-2008+MW20081110

[128] TUSB3200 Evaluation Module (EVM) retrieved July 2009, URL http://www.indesign-llc.com/ultracart/p-TUSB3200EVM_buy_now.html

[129] Wave PCM soundfile format, 20th January 2003, retrieved July 2009, URL http://ccrma.stanford.edu/courses/422/projects/WaveFormat/

[130] Wikipedia, “Terms of Use”, retrieved July 2009, URL http://wikimediafoundation.org/wiki/Terms_of_Use

[131] Xilinx Spartan 3A FPGA, retrieved July 2009, URL http://www.xilinx.com/products/spartan3a/

[132] Xilinx Spartan®-3A Starter Kit, HW-SPAR3A-SK-UNI-G, retrieved July 2009, URL http://www.xilinx.com/products/devkits/HW-SPAR3A-SK-UNI-G.htm

[133] Xubuntu, retrieved July 2009, URL: http://www.xubuntu.org/

Appendix A

Copyright and Trademark Information

The implementation, analysis and verification of the digital microphone array required the use of EDA (Electronic Design Automation) tools. The product or tool names and the company names are protected by copyright. General copyrights and trademarks are acknowledged in this section to simplify further reading of the thesis.

Copyrights (© or ®) used in this thesis are:

- Dolby
- Intel
- Matlab
- Mentor ModelSim
- Microsoft, Windows, Vista and XP
- SLIMbus
- Xilinx ISE
- µVision

Trademarks (™) used in this thesis are:

- Filter Design and Analysis Tool fdatool
- ModelSim ISE
- CORE Generator
- Keil
- Python

Company names (inc and plc) used in this thesis are:

- Knowles Electronics/Acoustics
- TI/Texas Instruments
- Wolfson Microelectronics
- Xilinx

Permissions and Copyright

All photos, figures and graphs used in this dissertation are reproduced with the permission of the author and, if applicable, publisher. For simplicity, this is not explicitly stated but indicated using (with kind permission of <author>) as shown in Figure 59 below.

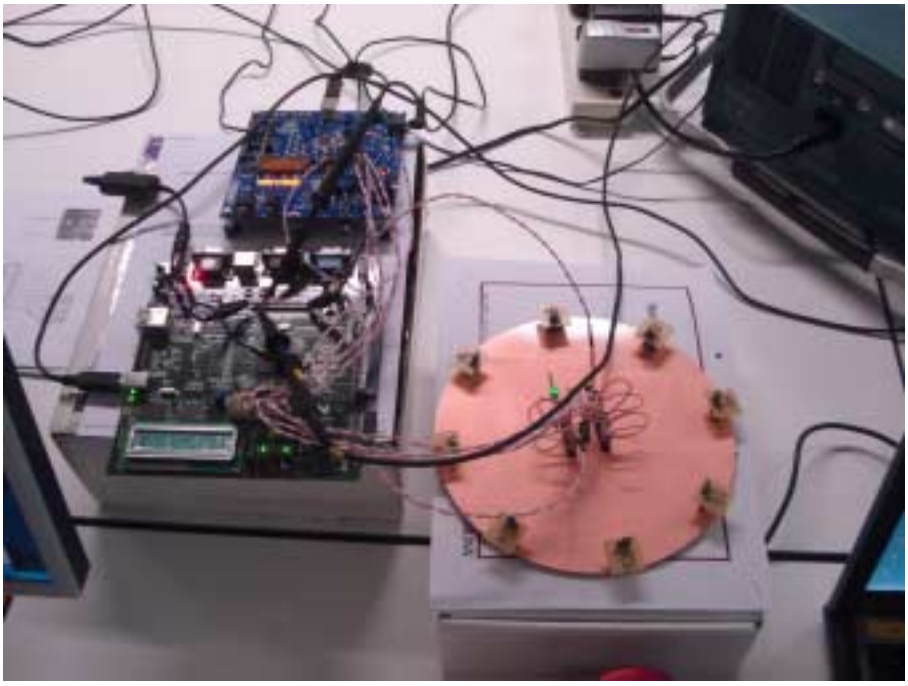


Figure 59 Digital Microphone Array (with kind permission of [...])

I would like to thank the following people for allowing me to use their pictures and figures.

- Institute of Physics (http://www.ipo.org) and Infineon (http://www.infineon.com), particularly Jill Membrey from the IOP and Roland Helm from Infineon
- Springer Science+Business Media (http://www.springer.com), particularly Barbara Schmidt-Löffler and the authors G. (Arden) Huang and Gary Elko
- Uwe Beis (http://www.beis.de/)
- Matthias Wölfel (http://www.distant-speech-recognition.org/)

Appendix B

AMI Meetings Corpus Consent Form

Version 1.1, October 8, 2004

Multi-modal research requires large amounts of acoustic recordings of spoken language, along with high quality video, and other multi-modal data recordings. Our goal is to compile such a corpus. This corpus will include a large number of Non-Native-English-Speakers, and will therefore be unique from those compiled at other institutions.

We are asking that when you participate in meetings in our specially equipped recording rooms, you allow us to record the meeting data. You may record multiple meetings, but will only need to complete this form once. Your participation is voluntary and you may stop at any point. The data will initially be used by the AMI Project Partners. It is possible however that at a later stage we will make some or all of the data available to the wider research community, in both transcribed and digitised formats.

No one other than the project staff will have access to any forms you provide to us. However, your name and general demographics may be mentioned in the course of your meeting(s), and you may be recognisable to some people. For this reason it is impossible to completely guarantee anonymity for things you may say. Some general demographics are also typically included in the scientific documentation of corpora and in published findings (e.g. age, dialect information) however, under no circumstances will your name and contact information be divulged as part of the published demographic information.

Please remember that comments you make about people or companies can defame them or invade their privacy, even if you/they are not specifically named but are still recognizable, so it is your responsibility to monitor your speech/behavior. **If you are concerned about any of your data, please advise us immediately and we can arrange for you to review the meeting(s) online**. **On your request we can remove a part of a meeting.**

By signing this form, you agree to allow us to record you and accept responsibility for your conduct in the meeting(s). It is your responsibility to monitor your own speech and actions during the meeting(s), and advise us if any data should be removed.

To indicate that you wish to participate as outlined above, please complete the following:

I, (please print name).....

have read this form, agree to its content and agree to take part in the research on these terms.

Signature: ..... Date: .....

Age: (optional)..... Sex: .....

Are you a native English speaker?

☐ **Yes**, please indicate your country and region.....

☐ **No**, please indicate your native language.....

How many months have you spent living in an English speaking country? .....

Which English speaking country have you lived in?.....

Please list any other language influences (other languages spoken, dialects, etc) .....

.....

Please provide your email address (or other contact information) so that we can contact you if necessary.

.....

Figure 60 AMI Meeting Corpus Consent Form