IDIAP COMMUNICATION

Small Microphone Array: Algorithms and Hardware

Iain McCowan [a] Darren Moore [a]

IDIAP–Com 03-07

August 2003

Dalle Molle Institute for Perceptual Artificial Intelligence • P.O.Box 592 • Martigny • Valais • Switzerland

phone +41 − 27 − 721 77 11
fax +41 − 27 − 721 77 12
e-mail secretariat@idiap.ch
internet http://www.idiap.ch

[a] IDIAP

1 Overview

This report describes the processing algorithms and gives an overview of the hardware for the small microphone array unit in the IM2.RTMAP (Real-time Microphone Array Processing) project. The algorithms include techniques for speech enhancement, speaker localisation and speaker segmentation. The hardware consists of a DSP platform with 8 audio inputs and outputs, as well as a Fireware interface for communication with a PC or other modules.

2 Introduction

This report forms Deliverable D1 of the Real-time Microphone Array Processing white paper project within IM2 (IM2.RTMAP). It specifies the array processing algorithms and hardware for the small microphone array (SMA) module. The RTMAP project aims to develop a real-time microphone array system based on a modular architecture. The basic unit of the modular system is the SMA, which should be capable of stand-alone operation, performing enhancement, localisation and segmentation of a small number of speakers located in its vicinity. The particular algorithms that will be used for each of these tasks are described in this document, along with the hardware on which they will be implemented.

3 Algorithms

3.1 Speech Enhancement

The algorithm that will be implemented for speech enhancement will be a superdirective beamformer with post-filter based on a noise field coherence model $\hat{\mathbf{\Gamma}}_{nn}$, as described in [1, 2]. A summary of the technique is presented here.

Common practice in microphone array processing is to model the received multi-channel input as the desired signal filtered by the acoustic path to each microphone, plus an additive noise component on each channel. That is (omitting the frequency dependence for clarity),

$$\mathbf{x}=s\mathbf{d}+\mathbf{n} \tag{1}$$

where s is the desired signal, $\mathbf{d}$ is the propagation vector of the signal source

$$\mathbf{d}=\begin{bmatrix} d_1 & d_2 & \cdots & d_N \end{bmatrix}^T \tag{2}$$

and $\mathbf{n}$ is similarly the vector of additive noise signals

$$\mathbf{n}=\begin{bmatrix} n'_1 & n'_2 & \cdots & n'_N \end{bmatrix}^T \tag{3}$$

where N is the number of microphones in the array.

Using this model, Simmer *et al* [3] demonstrate how the optimal broadband Minimum Mean Square Error (MMSE) filter solution (that is, the multi-channel Wiener filter) can be expressed as a single-channel Wiener filter operating on the output of a classical Minimum Variance Distortionless Response (MVDR) beamformer, that is,

$$\mathbf{w}_{opt} = \left[\frac{\phi_{ss}}{\phi_{ss}+\phi_{nn}}\right]\frac{\Phi_{nn}^{-1}\mathbf{d}}{\mathbf{d}^H\Phi_{nn}^{-1}\mathbf{d}} \tag{4}$$

where $\mathbf{w}_{opt}$ is the vector of optimal filter coefficients, ϕ_{ss} and ϕ_{nn} are respectively the (single-channel) signal and noise auto-spectral density vectors, and Φ_{nn} is the (multi-channel) noise cross-spectral matrix. The bracketed factor in the above expression corresponds to a single-channel Wiener filter, while the remaining factor forms the well known solution for the filters of a Minimum Variance

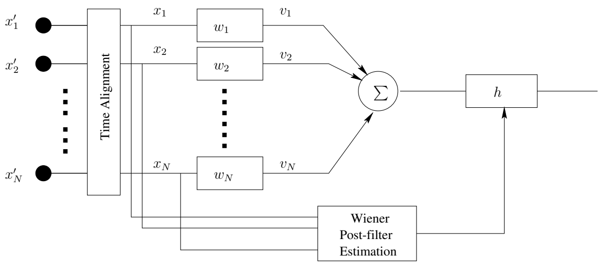

Figure 1: Filter-sum beamformer with post-filter

Distortionless Response (MVDR) beamformer [4]. The above equation suggests an optimal array processing structure like that shown in Figure 1, in which the transfer function of the single-channel Wiener post-filter is typically estimated from the aligned multi-channel input. The beamformer first maximises the directivity of the array response, and then the post-filter further enhances the output broadband Signal to Noise Ratio (SNR).

The beamformer term in Equation 4 will be implemented in the SMA using the well-known superdirective beamformer [4], which is the MVDR solution in a diffuse noise field (described below). The superdirective beamformer has been shown to lead to good performance in speech recognition applications [5].

To estimate the post-filter term in the above equation, that is

$$h = \frac{\phi_{ss}}{\phi_{ss} + \phi_{nn}}, \tag{5}$$

we require some estimate of the signal and noise spectral densities.

A useful measure to characterise noise fields is the *complex coherence function*. The coherence between two signals at points i and j is defined as

$$\Gamma_{ij} = \frac{\phi_{ij}}{\sqrt{\phi_{ii}\phi_{jj}}} \tag{6}$$

where ϕ_{ij} is the cross-spectral density between the signals at i and j. The coherence has the range $|\Gamma_{ij}| \leq 1$, and is essentially a normalised measure of the correlation that exists between the signals at two discrete points in a noise field.

A common technique for implementing a microphone array post-filter was proposed by Zelinski [6]. The Zelinski post-filter formulation, however, makes the assumption that the noise between sensors is uncorrelated, corresponding to a perfectly incoherent noise field ($\mathbf{\Gamma}_{nn} = \mathbf{I}$). Such a noise field will seldom occur in practice, although it can be a reasonable approximation if the spacing between sensors is sufficiently large.

While the Zelinski post-filter approximation has been shown to give reasonable performance in a variety of conditions [7, 5], the performance would be improved if a more accurate model of the noise field were used. In the following, the complex coherence function of the noise field is used to formulate a more appropriate estimation of the array post-filter transfer function.

With the assumptions of aligned signal on all sensors, zero correlation between the desired signal and the noise, if a model of the coherence ($\hat{\mathbf{\Gamma}}_{nn}$) is available, and the noise power spectrum is the same across sensors ($\phi_{n_in_i} = \phi_{nn}, \forall i$) (as is the case for isotropic noise fields), then we can write

$$\phi_{x_ix_i} = \phi_{ss} + \phi_{nn} \tag{7}$$

$$\phi_{x_jx_j} = \phi_{ss} + \phi_{nn} \tag{8}$$

$$\phi_{x_ix_j} = \phi_{ss} + \Gamma_{n_in_j}\phi_{nn} \tag{9}$$

The signal power spectral density can thus be estimated as

$$\hat{\phi}_{ss}^{(ij)} = \frac{\Re\left\{\hat{\phi}_{x_i x_j}\right\} - \frac{1}{2}\Re\left\{\hat{\Gamma}_{n_i n_j}\right\}\left(\hat{\phi}_{x_i x_i} + \hat{\phi}_{x_j x_j}\right)}{\left(1 - \Re\left\{\hat{\Gamma}_{n_i n_j}\right\}\right)} \tag{10}$$

where the average of $\phi_{x_i x_i}$ and $\phi_{x_j x_j}$ is taken to improve robustness. The post-filter denominator $(\phi_{ss} + \phi_{nn})$ can be estimated by $\hat{\phi}_{x_i x_i}$, as for the Zelinski technique.

The estimate is improved by averaging the solution over all unique sensor combinations, resulting in the post-filter

$$\hat{h}_p = \frac{\frac{2}{N(N-1)}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\hat{\phi}_{ss}^{(ij)}}{\frac{1}{N}\sum_{i=1}^{N}\hat{\phi}_{x_i x_i}} \tag{11}$$

This post-filter formulation will be implemented in the SMA, using a diffuse noise field as model for the noise coherence function. A diffuse, or spherically isotropic, noise field is a good model for a number of practical reverberant noise environments encountered in speech enhancement applications, such as offices and cars [8, 9, 10]. A diffuse noise field is characterised by uncorrelated noise signals of equal power propagating in all directions simultaneously. It can be shown that the coherence of a diffuse noise field is real-valued and is given by [11]

$$\Gamma_{ij} = \mathrm{sinc}\left(\frac{2\pi f d_{ij}}{c}\right). \tag{12}$$

The post-filter in the SMA will therefore be an implementation of Equation 11, using a diffuse noise model. This technique was evaluated for speech enhancement and speech recognition in [1].

3.2 Speaker Localisation

To locate sources, a simple single source localization technique based on Time Delay of Arrival (TDOA) is used. In particular, we use the SRP-PHAT technique described in [12], due to its low computational requirements and suitability for reverberant environments.

We define a vector of theoretical time-delays associated with a 3-D location $Z \in \mathbb{R}^3$ as

$$\boldsymbol{\tau}^Z \triangleq (\tau^{1,Z}, \ldots, \tau^{p,Z}, \ldots \tau^{P,Z}), \tag{13}$$

where P is the number of pairs and $\tau^{p,Z}$ is the delay (in samples) between the microphones in pair p:

$$\tau^{p,Z} = \frac{(||Z - M_1^p|| - ||Z - M_2^p||)\, f_s}{c}, \tag{14}$$

where $M_1^p, M_2^p \in \mathbb{R}^3$ are the locations of the microphones in pair p, $||.||$ is the Euclidean norm, f_s the sampling frequency, and c the speed of sound. Note that for a given time-delay τ_0 and a given pair p there exists a hyperboloid of locations Z satisfying $\tau^{p,Z} = \tau_0$.

From two signals $s_1^p(t)$ and $s_2^p(t)$ of a given microphone pair p, the frequency-domain GCC-PHAT [13] is defined as:

$$\hat{G}_{PHAT}^p(f) \quad \triangleq \quad \frac{S_1^p(f) \cdot [S_2^p(f)]^*}{\left|S_1^p(f) \cdot [S_2^p(f)]^*\right|}, \tag{15}$$

where $S_1^p(f)$ and $S_2^p(f)$ are Fourier transforms of the two signals and $[\cdot]^*$ denotes the complex conjugate. Typically the two Fourier transforms are estimated on Hamming-windowed segments of 20-30 ms. By performing an Inverse Fourier Transform, and summing the time-domain GCC-PHAT $\hat{R}_{PHAT}^p(\tau)$ across pairs, we obtain the SRP-PHAT measure,

$$P_{SRP-PHAT}(Z) \quad \triangleq \quad \sum_{p=1}^{P} \hat{R}_{PHAT}^p(\tau^{p,Z}), \tag{16}$$

From this, the source location is estimated as

$$\hat{Z} = \arg\max_{Z \in \mathbb{R}^3} \left[P_{SRP-PHAT}(Z) \right], \tag{17}$$

Based on geometrical considerations, at least 3 microphone pairs ($P \geq 3$) are required to obtain a unique peak.

The maximization is implemented using an exhaustive search over a fixed grid of points, $H \subset \mathbb{R}^3$ such that

$$\forall Z \in \mathbb{R}^3 \quad \exists Z_H \in H \quad \Gamma(Z, Z_H) \leq \gamma_0, \tag{18}$$

where $\Gamma(Z_1, Z_2)$ is the distance in time-delay space (in samples),

$$\Gamma(Z_1, Z_2) \quad \triangleq \quad \sqrt{\frac{1}{P} \sum_{p=1}^{P} \left(\tau^{p,Z_1} - \tau^{p,Z_2} \right)^2}, \tag{19}$$

and γ_0 is the desired precision in samples. Since we typically upsample the time-domain GCC-PHAT function $\hat{R}^p_{PHAT}(\tau)$ with a factor α_{up} (e.g. 20), the desired precision is set accordingly to $\gamma_0 = 1/\alpha_{up}$.

The grid H is built by picking points heuristically on a few concentric spheres centered on the microphone array. The spheres' radii were also determined by γ_0. Conceptually this approach relates to [14].

For each time frame, our implementation therefore approximates Eq. 17 with

$$\hat{Z} \approx \arg\max_{Z \in H} \left[P_{SRP-PHAT}(Z) \right]. \tag{20}$$

3.3 Speaker Segmentation

The segmentation approach is the location-based approach proposed in [15], which assumes a speaker k is confined to a physical region centred at location $\mathbf{x}_k \in \mathbb{R}^3$. The technique consists of two steps:

1. Classify each (speaker, frame) as speech or silence, *independently of other speakers and other frames*, thus obtaining K binary series

   $$ss^{(k)} = \left(ss_1^{(k)}, \ldots, ss_n^{(k)}, \ldots, ss_N^{(k)} \right)$$

   where k is the speaker index ($1 \leq k \leq K$), n the frame index ($1 \leq n \leq N$) and $ss_n^{(k)} \in \{0, 1\}$. "0" denotes a silent frame, "1" denotes a speech frame.

2. For each speaker k, apply a simple dilation/erosion process to smooth the binary sequence $ss^{(k)}$. This operation aims at connecting frames belonging to the same utterance, as well as eliminating spurious speech segments less than a specified minimum duration.

This technique was chosen as it implicitly handles the case of concurrent overlapping speakers (a common situation in real multi-party conversations), and lends to an online implementation.

3.3.1 Step One: Frame-Level Speech/Silence Classification

In contrast to the single stream of features used in the HMM approach in [16], we use here a separate stream of features for each speaker. Therefore, multiple speakers can be active within the same frame. For a given speaker k and a given frame n, we estimate the Steered Response Power (SRP) using a measure known as SRP-PHAT [17]. We sum the time domain version of the GCC-PHAT function defined in (15) at the theoretical time-delays associated with location $\mathbf{x}_k$:

$$P_{SRP}(k,n) \triangleq \frac{1}{P}\sum_{p=1}^{P} \hat{R}^{(p)}_{PHAT}\left(\mu_k^{(p)}\right) \tag{21}$$

where P is the number of microphone pairs and $\hat{R}^{(p)}_{PHAT}(\tau)$ is the time-domain GCC-PHAT. We have the property $P_{SRP}(k,n) \in [-1,+1]$. The higher the value of $P_{SRP}(k,n)$, the more likely it is for speaker k to be active at frame n.

For a given speaker k and a given frame n, speech/silence classification then amounts to:

$$ss_n^{(k)} = \begin{cases} 0 & \text{if} \quad P_{SRP}(k,n) < \theta_{SRP} \\ 1 & \text{if} \quad P_{SRP}(k,n) \geq \theta_{SRP} \end{cases} \tag{22}$$

where $\theta_{SRP} \in [-1,+1]$ is a threshold value that has to be tuned. In practice, most values $P_{SRP}(k,n)$ are positive and a typical threshold value is $\theta_{SRP} = 0.25$.

3.3.2 Step Two: Dilation/Erosion Process

Speech from one person mostly consists of short spurts (phonemes, words), interspersed with short silences. In obtaining a smooth speech/silence segmentation for each speaker, it is desirable to achieve two goals:

- **Goal 1**: to group spurts in order to form utterances. For a given speaker, two spurts that are separated by a small silence (e.g. less than 1 second) must be linked into the same segment.
- **Goal 2**: to remove any isolated spurt that lasts less than a minimum duration (e.g. 200 ms). We assume that such a spurt contains noise rather than speech.

Initially, we attempted to use single speaker HMMs to achieve the above goals. However, since a speech segment contains short alternating periods of speech and silence, it was found that a complex HMM topology was required, similar to that proposed for the overlaps in [16]. In addition, obtained results were significantly less than those of the previous work. In the current work, we instead achieve the above goals using an alternative approach based on simple binary dilation and erosion operators.

We apply a sequence of such operators on the binary series $ss^{(k)}$, thus achieving an effect similar to low-pass filtering in signal processing. The L-frame dilation operator for a binary sequence $u = \{u_n\}$ (with values in $\{0,1\}$) is defined as:

$$u = \{u_n\} \quad \rightarrow \quad v = f^L_{dil}(u)$$

$$\text{where} \quad \forall n \quad v_n = \max\left(u_{n-L}, \ldots, u_{n+L}\right)$$

The L-frame erosion operator for a binary sequence $u = \{u_n\}$ is defined as:

$$u = \{u_n\} \quad \rightarrow \quad v = f^L_{ero}(u)$$

$$\text{where} \quad \forall n \quad v_n = \min\left(u_{n-L}, \ldots, u_{n+L}\right)$$

In practice, the beginning and the end of u are mirrored to solve boundary problems.

For a given speaker k, the two goals mentioned above are achieved using a succession of dilations and erosions:

$$ss^{(k)} \quad \rightarrow \quad ss2^{(k)} = f^{L_2}_{dil}\left(f^{L_2+L_1}_{ero}\left(f^{L_1}_{dil}\left(ss^{(k)}\right)\right)\right)$$

where L_1 is the maximum "small silence" duration in frames (relates to goal 1.) and L_2 is the minimum speech duration in frames (relates to goal 2.). This operation can be implemented online with a buffer of $2 \times (L_1 + L_2)$ frames, incurring a delay of $L_1 + L_2$ frames.

4 Hardware and Low-level Software

4.1 High-level Requirements and Development Strategy

The high level requirements for the the SMA were :

- The SMA shall be a stand-alone device.
- The SMA shall have 8 microphone inputs with suitable ADC characteristics for microphone array processing.
- The SMA shall have sufficient computing power for real-time (floating-point) execution of SMA processing algorithms developed within the whitepaper.
- The SMA shall be able to transfer multi-channel digital audio data in real-time to/from a host PC or other SMA.
- The SMA shall have at least 2 loudspeaker outputs.
- Low-level software operations related to the acquisition, output and transfer of digital audio data shall be encapsulated within API's.

The general strategy taken in developing the SMA hardware to meet the above requirements was to use off-the-shelf evaluation boards and components wherever possible, with custom circuitry providing necessary interfaces between the off-the-shelf components and with external hardware.

All hardware and low-level software development was done by a team from the Infotronics Department of the Haute Ecole Valaisanne - Sion within a sub-project of the main white paper. A more detailed technical description of the SMA hardware and low-level software is provided in their final report [18], which is available upon request.

4.2 Hardware

The hardware solution for the SMA is based around relatively inexpensive off-the-shelf DSP and codec evaluation boards, with custom circuitry implemented to provide interfaces between the two evaluation boards, and between the SMA and external hardware (host-PC, microphones, loudspeakers). A high level overview of the hardware architecture is included in Figure 2.

4.2.1 DSP Evaluation Board

The evaluation board for the Analog Devices TigerSHARC DSP (ADSP-TS101S-EZLITE) was chosen to provide the necessary processing power for the SMA. The evaluation board features two TigerSHARC processors (each providing peak performance of 1.5GFLOPs) as well as 32Mb of RAM, high-speed link port interfaces and a JTAG debugging interface connected through a USB port.

A preliminary analysis of processing requirements revealed that evaluation boards for other varieties of DSP (e.g. Texas Instruments TMS320C6711) were inadequate for the tasks required to be implemented on the SMA. The TigerSHARC evaluation board was the only suitable candidate. More details are provided in [18].

4.2.2 Codec Evaluation Board

The evaluation board for the Texas Instruments TLV320AIC10 audio codec was chosen to provide the ADC/DAC functionality for the microphone inputs and loudspeaker outputs. Each TLV320AIC10 audio codec is capable of simultaneous ADC of one input channel and DAC of one output channel at rates of up to 22kHz with 16-bit resolution. The evaluation board ships with 2 codecs fitted, and has footprints for 6 additional codecs. The 2 pre-fitted codecs have microphone and loudspeaker

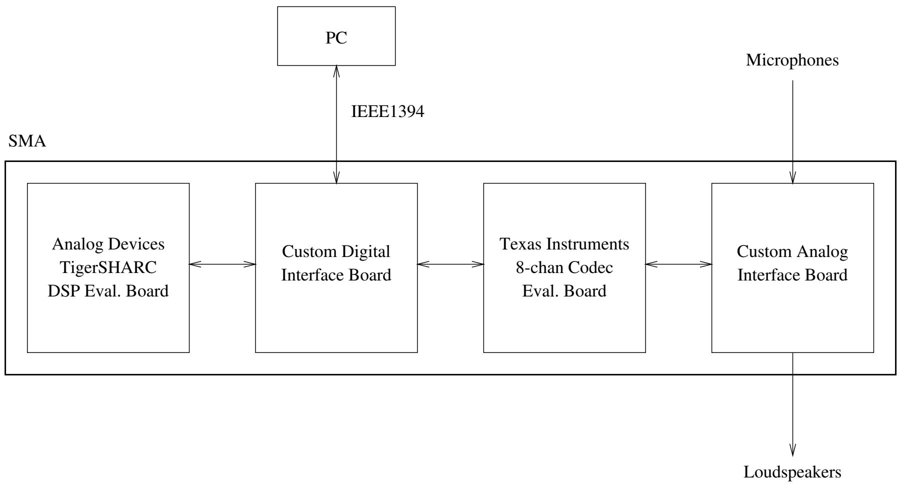

Figure 2: SMA hardware architecture

interface circuitry (ie. voltage bias for electret mics, input pre-amplifiers, output drivers, etc.), but the remaining 6 codec footprints do not have the accompanying analog conditioning circuitry.

The evaluation board was designed for use with an evaluation board for a Texas Instruments DSP, and therefore uses a SPI protocol for transfer of digital audio to and from the codecs. The Analog Devices TigerSHARC DSP does not contain an SPI port, so custom interface circuitry was required to enable communications between the two evaluation boards (see below).

4.2.3 Custom Analog Interface Board

Due to the lack of analog conditioning circuitry for 6 of the codecs on the codec eval. board, a separate custom analog interface board was required. Circuitry was implemented for all 8 microphone inputs and loudspeaker outputs, and the existing signal conditioning circuitry on the codec evaluation board was disabled to ensure uniform signal conditioning characteristics across all input and output channels.

4.2.4 Custom Digital Interface Board

A digital interface board, consisting of an FPGA with some peripheral IC's fulfils two roles :

1. provides the SPI-to-Link-Port interface for digital audio transfer between the codec and DSP eval. boards (as described previously).

2. provides the high-speed IEEE1394 (Firewire) interface between the TigerSHARC external port and an external device (host PC or another SMA).

Along with implementing the logic required for digital data transfers through the firewire and codec interfaces, the FPGA provides control and status registers that are mapped into the memory address space of the TigerSHARC DSPs.

An RS232 line driver is also connected electrically to the FPGA, but FPGA software support for RS232 remains to be implemented.

4.2.5 Hardware Summary

The main features of the implemented SMA hardware are :

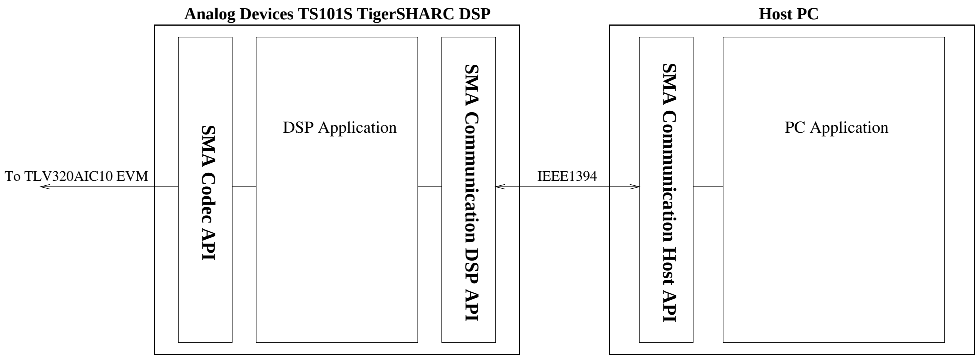

Figure 3: SMA High Level Software Architecture

- 2 Analog Devices TigerSHARC DSP's, providing 3GFLOPs computing power for each SMA.
- 8 microphone inputs with variable input gain and bias voltage for use with electret mics.
- 8 amplified loudspeaker outputs with variable output gain.
- 8 codecs for simultaneous ADC/DAC of all analog inputs and outputs at rates of up to 22kHz/channel.
- 2 High speed IEEE1394 (Firewire) interfaces for communications between multiple SMAs and/or a host PC.
- RS232 port circuitry implemented for possible future use with PTZ camera.

The resulting SMA hardware, consisting of a stack of 4 PCBs has been housed in a metal case. A photograph of the SMA hardware is included in Figure 4.

4.3 Low Level Software

The high level architecture of the SMA software is illustrated in Figure 3. SMA-related software executes on one or both of the TigerSHARC DSPs in the SMA hardware and also on a host PC. The software components developed in this phase of the project were the APIs that encapsulate the low-level software operations related to the configuration and control of the codec and firewire hardware interfaces, and the transfer of digital audio data over these interfaces.

4.3.1 SMA Codec API

The SMA Codec API provides functions to :

1. configure the signal conditioning and sampling characteristics of the codecs on the TLV320AIC10 evaluation board (sampling frequency, preamplifier gain, input/output channel usage).
2. configure input/output frame parameters (frame length, frame overlap) and setup DSP interrupt service routines to process frames.
3. start/stop codec operation in different modes (input-only, output-only, loopback mode, input and output).

Digital audio transfers on the DSP to or from the codecs utilise DMA, resulting in minimal overhead on the DSP.

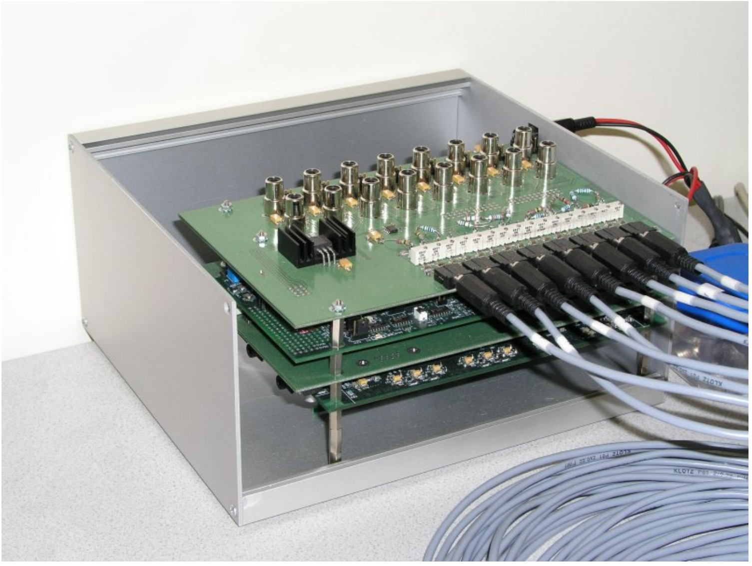

Figure 4: Small Microphone Array Hardware

4.3.2 SMA Communications API

The SMA Communications API consists of 2 APIs on the TigerSHARC DSP and the host PC providing data transfer capability over the IEEE1394 interface. The API on the DSP is based on previous work done by HEVs that implemented a firewire interface for a Texas Instruments DSP platform. The API on the host PC is based on a commercial firewire library (Unibrain).

The API implements 2 types of transfer, a *command* transfer or a *data* transfer. Command transfers are initiated by the host PC and are intended for configuring and controling data acquisition and processing on the SMA. An interrupt is generated on the DSP whenever a command is received from the host, which is serviced by a user defined callback. Data transfers are always initiated by the DSP, by either requesting a new buffer from the host during output, or by sending a newly acquired buffer to the host PC. The host API uses callback functions to service output buffer requests or input buffer arrivals.

5 Summary

This document has described the SMA hardware and low-level software as well as the algorithms for speech enhancement, speaker localisation and speaker segmentation that will be implemented in real-time. The SMA hardware and low-level software was developed by HEVs within a sub-project of the main white paper project. The implemented SMA hardware platform has sufficient computing power and I/O interfaces for the successful real-time implementation of the algorithms described.

References

[1] I. McCowan and H. Bourlard, "Microphone array post-filter based on noise field coherence," *Submitted to IEEE Transactions on Speech and Audio Processing*, 2001. Available as IDIAP RR 01-40.

[2] I. McCowan and H. Bourlard, "Microphone array post-filter for diffuse noise field," in *Proceedings of ICASSP-02*, vol. 1, pp. 905–908, 2002.

[3] K. U. Simmer, J. Bitzer, and C. Marro, “Post-filtering techniques,” in *Microphone Arrays* (M. Brandstein and D. Ward, eds.), ch. 3, pp. 36–60, Springer, 2001.

[4] H. Cox, R. Zeskind, and M. Owen, “Robust adaptive beamforming,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 35, pp. 1365–1376, October 1987.

[5] I. McCowan, C. Marro, and L. Mauuary, “Robust speech recognition using near-field superdirective beamforming with post-filtering,” in *Proceedings of ICASSP 2000*, vol. 3, pp. 1723–1726, 2000.

[6] R. Zelinski, “A microphone array with adaptive post-filtering for noise reduction in reverberant rooms,” in *Proceedings of ICASSP-88*, vol. 5, pp. 2578 –2581, 1988.

[7] C. Marro, Y. Mahieux, and K. Uwe Simmer, “Analysis of noise reduction and dereverberation techniques based on microphone arrays with postfiltering,” *IEEE Transactions on Speech and Audio Processing*, vol. 6, pp. 240–259, May 1998.

[8] J. Bitzer, K. U. Simmer, and K. Kammeyer, “Theoretical noise reduction limits of the generalized sidelobe canceller (gsc) for speech enhancement,” in *Proceedings of ICASSP 99*, vol. 5, pp. 2965–2968, 1999.

[9] J. Meyer and K. U. Simmer, “Multi-channel speech enhancment in a car environment using wiener filtering and spectral subtraction,” in *Proceedings of ICASSP 97*, vol. 2, pp. 1167–1170, 1997.

[10] G. W. Elko, “Superdirectional microphone arrays,” in *Acoustic Signal Processing for Telecommunication* (S. Gay and J. Benesty, eds.), ch. 10, pp. 181–237, Kluwer Academic Publishers, 2000.

[11] R. K. Cook, R. V. Waterhouse, R. D. Berendt, S. Edelman, and M. C. T. Jr, “Measurement of correlation coefficients in reverberant sound fields,” *Journal of the Acoustic Society of America*, vol. 27, pp. 1072–1077, 1955.

[12] J. DiBiase, H. Silverman, and M. Brandstein, “Robust localization in reverberant rooms,” in *Microphone Arrays* (M. Brandstein and D. Ward, eds.), ch. 8, pp. 157–180, Springer, 2001.

[13] C. Knapp and G. Carter, “The generalized correlation method for estimation of time delay,” *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. ASSP-24, pp. 320–327, August 1976.

[14] S. M. Griebel and M. S. Brandstein, “Microphone array source localization using realizable delay vectors,” in *Proceedings of the 2001 IEEE Workshop on Applications of Signal Proce ssing to Audio and Acoustics*, pp. 71–74, 2001.

[15] G. Lathoud, I. McCowan, and D. Moore, “Segmenting multiple concurrent speakers using microphone arrays,” in *Proceedings of Eurospeech 2003*, September 2003.

[16] G. Lathoud and I. McCowan, “Location based speaker segmentation,” in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, April 2003.

[17] J. DiBiase, *A High-Accuracy, Low-Latency Technique for Talker Localization in Reverberant Environments*. PhD thesis, Brown University, Providence RI, USA, 2000.

[18] J. Moerschell, G. Maitre, C. Praplan, and C. Briguet, “SMA low level development project final report,” tech. rep., Haute Ecole Valaisanne (HEVs), 2003.