A SPHERICAL MICROPHONE ARRAY FOR SYNTHESIZING VIRTUAL DIRECTIVE MICROPHONES IN LIVE BROADCASTING AND IN POST PRODUCTION

ANGELO FARINA[1], ANDREA CAPRA[1], LORENZO CHIESI[1] AND LEONARDO SCOPECE[2]

[1] Industrial Eng, Dept., University of Parma, Via G.P. Usberti 181/A, Parma, Italy
farina@unipr.it
[2] RAI CRIT Research and Technology Innovation Center, Corso Giambone 68 , Torino, Italy
l.scopece@rai.it

The paper describes the theory and the first operational results of a new multichannel recording system based on a 32-capsules spherical microphone array. Up to 7 virtual microphones can be synthesized in real-time, choosing dynamically the directivity pattern (from standard cardioid to 6th-order ultradirective) and the aiming. A graphical user's interface allows for moving the virtual microphones over a 360-degrees video image. The system employs a novel mathematical theory for computing the matrix of massive FIR filters, which are convolved in real time and with small latency thanks to a partitioned convolution processor.

INTRODUCTION

In recent years a lot of research has been produced about technologies for capturing and reproducing the spatial properties of the sound. Most of the proposed approaches employ massive arrays of microphone and loudspeakers, and process the signals by means of very complex mathematical theories, based on various modifications of the classical Huygens principle.

These methods include decomposition of the sound field as a superposition of plane waves, working in Cartesian coordinates (for example WFS, [1]), or as a superposition of spherical harmonics, working in spherical coordinates (High Order Ambisonics, [2]). Recently, even more complex methods have been proposed by Nelson and Fazi (decomposing the sound field in complex Hankel functions, [3]).

Whatever method is employed, at the end one can always think to the whole processing as the synthesis of a number of "virtual microphones", each of them feeding a loudspeaker in the playback system.

Having realized that, we decided to remove the constraints inherent with any of the previously-known techniques, and to generate directly the desired virtual microphones as a direct transformation of the "raw" recorded signals, without relying on an intermediate layer (or kernel) of "basic" waveforms.

Albeit this approach can work, in principle, with any geometrical layout of the microphone array, we decided to develop our system around an high-quality 32-capsules spherical microphone array, recently made available on the market [4].

The 32 signals are filtered employing a massive convolution processor, capable of real-time synthesis and steering of up to 7 virtual directive microphones, controlling their aiming and capture angle by means of a joystick, and employing a wide-angle panoramic video camera and a graphical "view&point" interface for an easy-to-operate user's interface.

This can be done in real time and with small latency during a live broadcasting event; alternatively, the raw signals from the 32 capsules can be recorded, together with the panoramic video, for subsequent synthesis of the virtual microphone signals in post-production.

The paper focuses on the innovative algorithms developed for processing the signals, making use of real-time synthesis of the filtering coefficients; the computation of these filters is based on inversion of a matrix of impulse responses measured inside an anechoic room.

This method is not based on any of the previously known theories, such as High Order Ambisonics, WFS and similar. There is no intermediate spatial kernel, such as a set of spherical harmonics or Hankel functions.

The virtual microphones being synthesized can be highly directive (with polar pattern constant with frequency, and with a beam width much sharper than a "shotgun" microphone), and are intrinsically coincident, so the signals can be mixed without concerns of comb filtering; it is possible to continuously move their aiming for following actors or singers on scene, or for giving instantaneous miking to people in the audience.

Surround recording of a concert is just one of the possible scenarios for employing this approach, which has also been successfully tested for dramas, sport events, and TV shows in which there is systematic

interaction of conductors and guests with in-studio audience.
A careful analysis of the performances of the new microphone system did show that frequency response, signal-to-noise ratio and rejection of off-beam sounds are better than those obtainable employing traditional processing algorithms applied to the same input signals, or dedicated top-grade ultra-directive microphones.

1 DESCRIPTION OF THE SYSTEM

1.1 Computation of the filters: the theory

Given an array of transducers, a set of digital filters can be employed for creating the output signals. In our case the M signals coming from the capsules need to be converted in V signals yielding the desired virtual directive microphones: so we need a bank of $M \times V$ filters. As always, we prefer FIR filters.

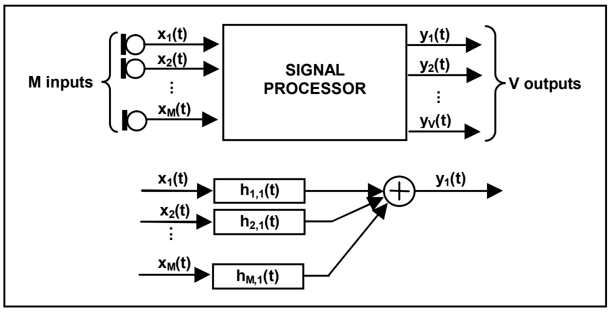


Figure 1: Scheme of the signal processing

Assuming $\boldsymbol{x_m}$ as the input signals of M microphones, $\boldsymbol{y_v}$ as the output signals of V virtual microphones and $\boldsymbol{h_{m,v}}$ the matrix of filters, the processed signals can be expressed as:

$$y_v(t) = \sum_{m=1}^{M} x_m(t) * h_{m,v}(t) \qquad (1)$$

Where * denotes convolution, and hence each virtual microphone signal is obtained summing the results of the convolutions of the M inputs with a set of M proper FIR filters.
One of the most used techniques for deriving filter banks for generating virtual microphones with arbitrary directivity is the Ambisonics method: first the M signals are processed deriving an equal or smaller number of spherical harmonics. Later these spherical harmonics signals are added together with proper gains, for synthesizing the desired virtual microphones. This has the advantage of allowing for the derivation of a large number of virtual microphones with a small additional effort, as most of the effort is represented by the computation of the spherical harmonic signals.
Furthermore, it is possible to change dynamically the aiming or the directivity of every virtual microphone simply changing the gains employed when adding the signals together.
In our approach, instead, every desired virtual microphone is derived directly from the original M signals, avoiding the Ambisonics encoding and decoding: the outputs of the processing system are directly the result of the digital filtering of the input signals, with a different set of filters for every virtual microphone.
In principle this allows for synthesizing virtual microphones having an arbitrary directivity pattern. In practice we decided, for now, to synthesize frequency-independent high-order cardioid virtual microphones.
The directivity factor Q of a virtual cardioid microphone of order n is described, in spherical coordinates θ, φ, by the expression:

$$Q_n(\vartheta,\varphi) = [Q_1(\vartheta,\varphi)]^n \qquad (2)$$

Where $Q_1(\vartheta,\varphi)$ is the directivity factor of a first order cardioid microphone:

$$Q_1(\vartheta,\varphi) = 0.5 + 0.5 \cdot cos(\vartheta) \cdot cos(\varphi) \qquad (3)$$

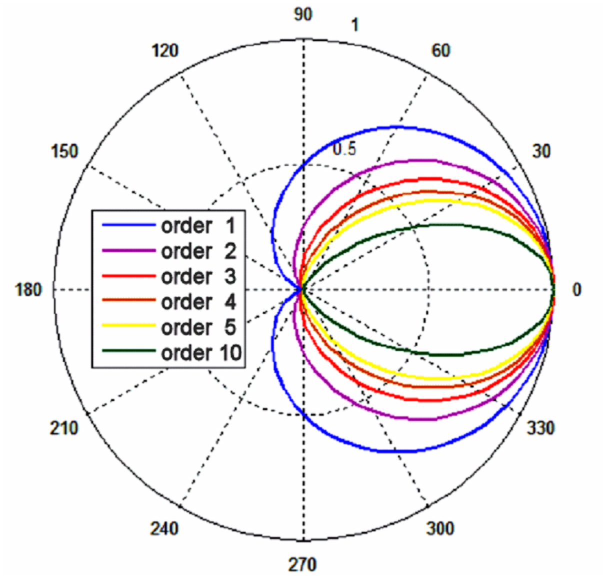
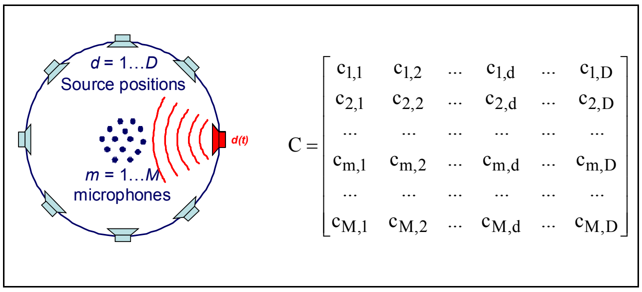


Figure 2: Polar plot of virtual cardioid microphones of various orders (target patterns $\boldsymbol{Q_n}$)

The processing filters h are usually computed following one of several complex mathematical theories, based on the solution of the wave equation, often under certain simplifications, assuming the microphones are ideal and identical. In some implementations the signal of each microphone is processed through a digital filter for compensating its deviation, with a heavier computational load.
In this novel approach no theory is assumed: the set of filters h are derived directly from a set of measurements,

made inside an anechoic room. A matrix of measured impulse response coefficients ***c*** is formed and the matrix has to be numerically inverted (usually employing some approximate techniques, such as Least Squares plus regularization); in this way the outputs of the microphone array are maximally close to the ideal responses prescribed. This method also inherently corrects for transducer deviations and acoustical artefacts (shielding, diffraction, reflection, etc.).
The characterization of the array is based on a matrix of measured anechoic impulse responses, obtained with the sound source placed at a large number D of positions all around the probe, as shown in Figure 3.



$$C = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,d} & \cdots & c_{1,D} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,d} & \cdots & c_{2,D} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ c_{m,1} & c_{m,2} & \cdots & c_{m,d} & \cdots & c_{m,D} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ c_{M,1} & c_{M,2} & \cdots & c_{M,d} & \cdots & c_{M,D} \end{bmatrix}$$

Figure 3: impulse response measurements from D source positions to the M microphones

The processing filters ***h*** should transform the measured impulse responses ***c*** into the prescribed theoretical impulse responses ***p***:

$$\sum_{m=1}^{M} c_{m,d} * h_m \Rightarrow p_d \qquad d = 1..D \tag{4}$$

Please notice that in practice the target impulse responses $\boldsymbol{p_d}$ are simply obtained applying a direction-dependent gain Q, given by eq. 2, to a delayed unit-amplitude Dirac's delta function δ: $p_d = Q_d \cdot \delta$.
Computation is easier in frequency domain (that is, computing the complex spectra, by applying the FFT algorithm to the N-points-long impulse responses ***c***, ***h*** and ***p***). Let's call ***C***, ***H*** and ***P*** the resulting complex spectra. This way, the convolution reduces to simple multiplication between the corresponding spectral lines, performed at every frequency index k:

$$\sum_{m=1}^{M} C_{m,d,k} \cdot H_{m,k} \Rightarrow P_d \qquad \begin{cases} d = 1..D \\ k = 0..N/2 \end{cases} \tag{5}$$

Now we pack the values of ***C***, ***H*** and ***P*** in proper matrixes, taking into account all the M input microphones, all the measured directions D and all the V outputs to create:

$$[H_k]_{MxV} = \frac{[P]_{DxV}}{[C_k]_{DxM}} \tag{6}$$

This over-determined system doesn't admit an exact solution, but it is possible to find an approximated solution with the Least Squares method, employing a regularization technique for avoiding instabilities and excessive signal boost [5,6]. The block diagram of the least-squares method is shown in Figure 4:

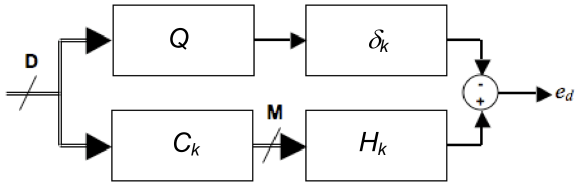


Figure 4: scheme of the Least Squared method with a delay in the upper branch

In this scheme we observe the delay block δ, required for producing causal filters, and the resulting total modelling error e, which is being minimized by the least-squares approach
In general, the frequency-domain representation of a Dirac's delta delayed by n_0 samples is given by:

$$\delta_k = e^{-j2\pi k \frac{n_0}{N}} \tag{7}$$

Albeit various theories have been proposed for defining the optimal value of the causalisation delay n_0, we did take the easy approach, setting n_0=N/2. Choosing N/2 samples is a safe choice, which creates inverse filters with their "main peak" close to their centre, and going smoothly to zero at both ends.
Furthermore, a regularization parameter is required in the denominator of the matrix computation formula, to avoid excessive emphasis at frequencies where the signal is very low.
So the solution formula, which was first proposed in Kirkeby et al. [6], becomes:

$$[H_k]_{MxV} = \frac{[C_k]^*_{MxD} \cdot [Q]_{DxV} \cdot e^{-j\pi k}}{[C_k]^*_{MxD} \cdot [C_k]_{DxM} + \beta_k \cdot [I]_{MxM}} \tag{8}$$

As shown in the image below, the regularization parameter β should depend on frequency. A common choice for the spectral shape of the regularization parameter is to specify it as a small, constant value inside the frequency range where the probe is designed to work optimally, and as much larger values at very low and very high frequencies, where conditioning problems are prone to cause numerical instability of the solution.

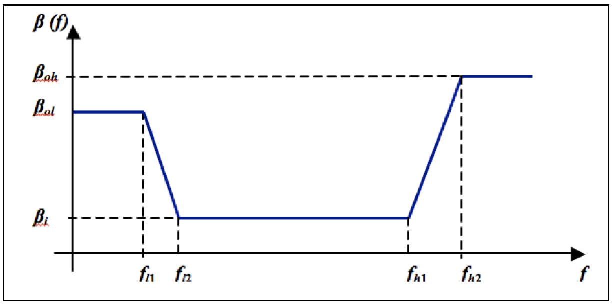


Figure 5: regularization parameter in dependence of the frequency

1.2 The microphone array

The experimentation described in this paper was realized using the Eigenmike™ microphone array produced by MH acoustics [4]. As shown in Figure 6 and Figure 7, the Eigenmike™ is a sphere of aluminium (the radius is 42 mm) with 32 high quality capsules placed on its surface; microphones, pre-amplifiers and A/D converters are packed inside the sphere and all the signals are delivered to the audio interface through a digital CAT-6 cable, employing the A-net protocol.
The audio interface is an EMIB Firewire interface; being based on the TCAT DICE II chip, it works with any OS (Windows, OSX and Linux through FFADO). It provides to the user two analogue headphones outputs, one ADAT output and the word clock ports for syncing with external hardware.
The preamplifier's gain control is operated through MIDI control; we developed a GUI (in Python) for making easy to control the gain in real-time with no latency and no glitches.
Actually RAI CRIT is the owner of two Eigenmikes™, shown in Figure 6 and Figure 7.

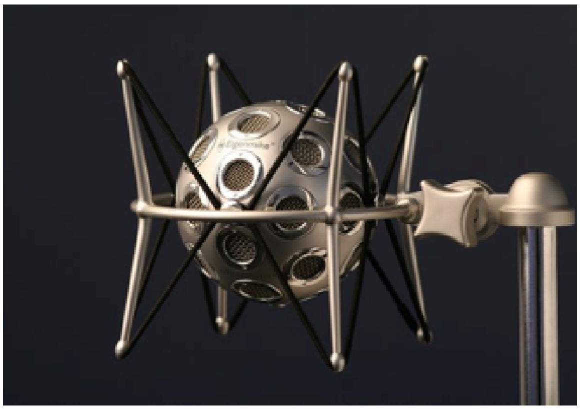


Figure 6: the first type Eigenmike™

Figure 7: the second type Eigenmike™

1.3 Experimental characterization of the array

Measurements of the microphone array were made employing the Exponential Sine Sweep (ESS) method, in order to obtain 32 Impulse Responses for each direction of arrival of the test signal.
The ESS method was chosen due to its capability of removing unwanted artefacts due to nonlinearities in the loudspeaker, and because it provides significantly better S/N than other methods based on periodic signals, such as MLS or linear sine sweep (TDS), as one of the author already discovered [7]. This made it possible to get a good S/N ratio employing short test signals, speeding up the measurement procedure, and consequently enhancing the time invariance of the system during the measurement period.
These measurements were made inside an anechoic room, to avoid undesired reflections and to maximize the signal/noise ratio.
The test signal was pre-filtered with a suitable FIR in order to flatten perfectly the frequency spectrum and to linearize the phase response of the full-range dual-concentric loudspeaker employed as sound source. The loudspeaker and the anechoic room were kindly made available by Eighteen Sound, Reggio Emilia, Italy.
The array was rotated along azimuth (36 steps) and elevation (18 steps), using a movable fixture for the azimuth rotation and a turntable for the elevation. In this way we obtained 36 x 18 x 32 impulse responses, each 2048 samples long (at 48 kHz).

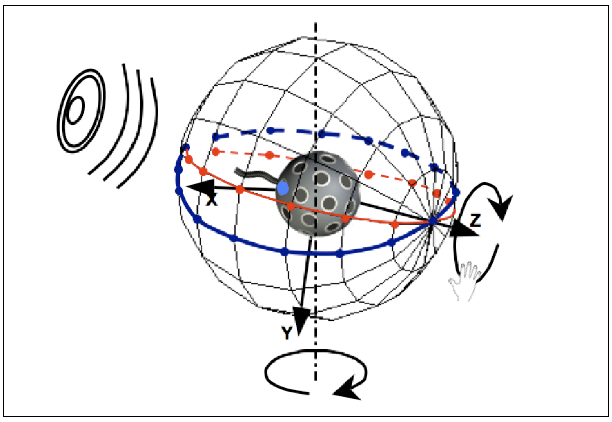


Figure 8: rotation of the microphone array

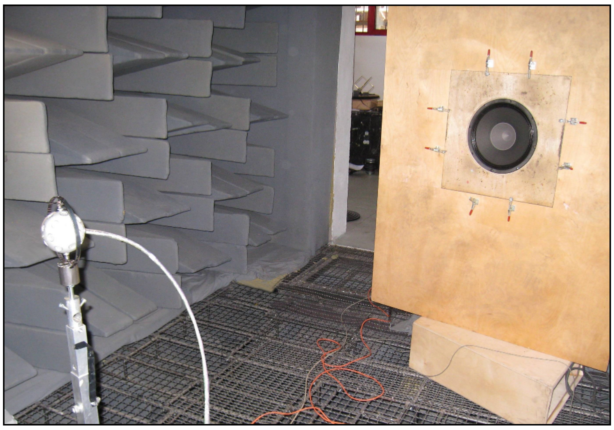


Figure 9: the microphone array in the anechoic room

Figure 10: movable fixture for azimuth rotation

The raw result of the measurement are the response of each capsule of the array to the sound arriving by every direction. The following picture shows the results for the capsule n.1 (all the others are very similar):

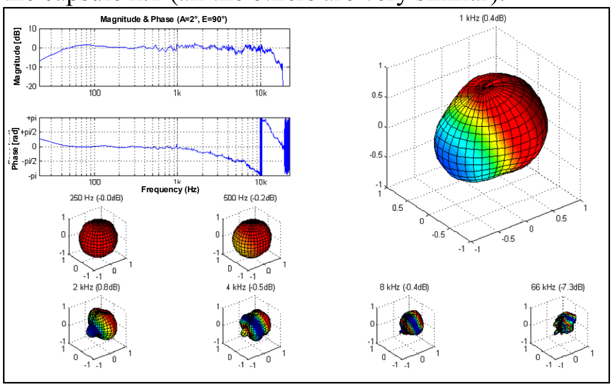


Figure 11: frequency response and polar patterns for capsule n.1

From the graphs shown in Figure 11 it is possible to verify that the directivity of the capsules, intrinsically omnidirectional at low frequency, is highly influenced by the microphone structure (the aluminium sphere): above 1 kHz the capsule becomes significantly directive. From the magnitude spectrum calculated in the direction in which the capsule shows the higher sensitivity, it is possible to verify that the transducers have a quite good frequency response from 30 Hz to 13 kHz, with a gentle roll off at higher frequencies.

A small phase error was introduced by this measurement technique in every impulse response, due to the variation of the distance between the microphone array and the sound source during the rotation. In particular the rotating table produced a sinusoidal variation of the distance caused by the little eccentricity of the support, which was accurately measured thanks to the absolute time-of-flight of the sound from the loudspeaker to the microphone, as shown in Figure 12:

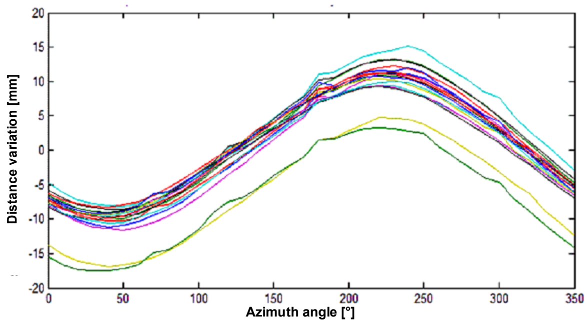


Figure 12: variation of the distance during rotation

With a mathematical calculation, based on the theoretical distance between capsule and source, implemented in a Matlab program, the impulse responses were carefully realigned in time, correcting for these positioning errors. Before this operation the worst case of distance error was 30 mm, after it was of 2 mm, a length that corresponds to the wavelength of a 170 kHz signal. For this reason we could assume that the phase coherence of impulse responses is reasonably good in the spectral region of interest. As shown in the Figure 13, in the case of a 3rd order virtual cardioid, the compensation improves the polar pattern at 4 and 8 kHz. In the band of 16 kHz the signal is compromised in both cases, and this is probably due to the spatial aliasing.

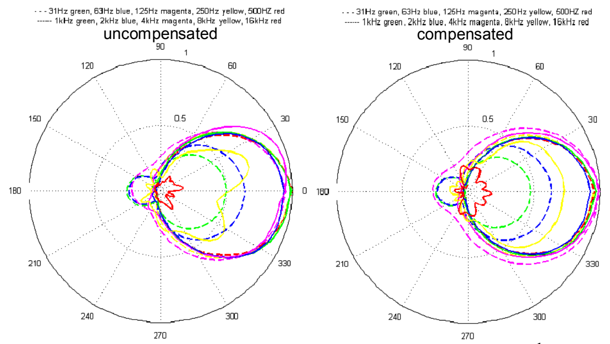


Figure 13: Improvement of polar patterns of a 3rd order virtual cardioid due to delay compensation

1.4 Synthesis and test of virtual microphones

In order to derive the matrix of filters, a Matlab script was produced. This script employs 2048 samples of each impulse response and it needs as inputs the number of virtual microphones to synthesize, their directivity, their azimuth and elevation. From these inputs, according with the theory and the procedure described in paragraph 1.1, is possible to invert the matrix of impulse responses obtaining the matrix of filters to apply to the capsule signals.

In our specific case the number of incoming sound directions D was 648, the number of microphones M was 32 and the number V of virtual microphones was the one desired.

The convolution of the FIRs matrix with the 32 signals coming from the capsules of the array should give as outputs the signals of virtual microphones with the desired characteristics. In the pictures below are shown some experimental results, showing some of the different directivity patterns obtained.

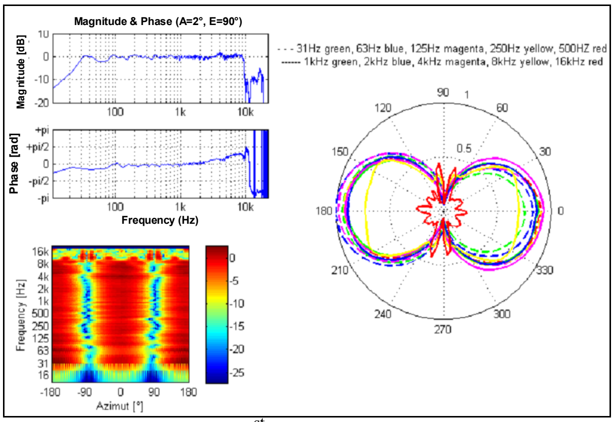


Figure 14: 1st-order figure of eight

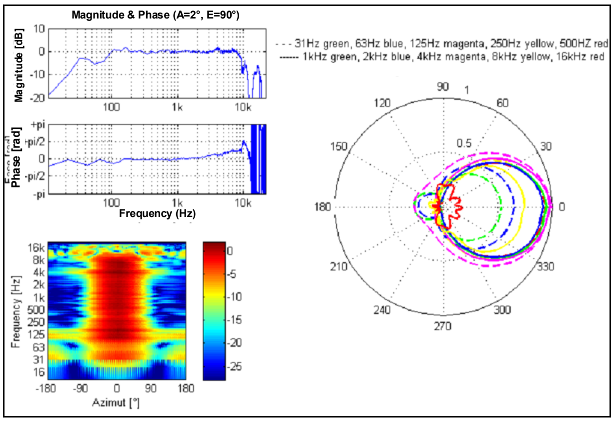


Figure 15: 3rd order cardioid

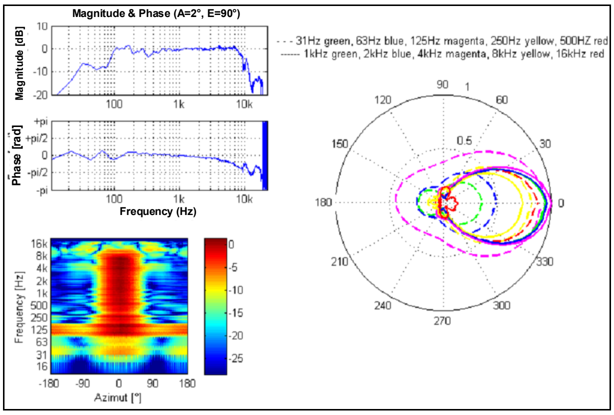


Figure 16: 6th order cardioid

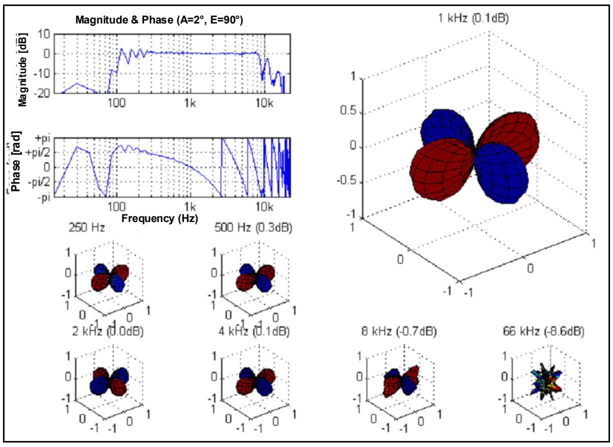


Figure 17: spherical harmonic "U" (2nd order)

1.4.1 Comparison between a "shotgun" microphone and a synthesized 6th order cardioid

One of the characteristics of this signal processing is the capability to synthesize virtual microphones with high directivity. A comparison between the most directive microphone used in broadcasting, a "shotgun" microphone, and a synthesized 6th order microphone could give an interesting evaluation of the performances of the new microphone system.

In the two pictures below the directivities of a Sennheiser MKH 70-1 (Figure 18) and 6th order cardioid (Figure 19) are compared. Please note that the scale is different (dB-scale for the Sennheiser, linear scale for our virtual microphone).

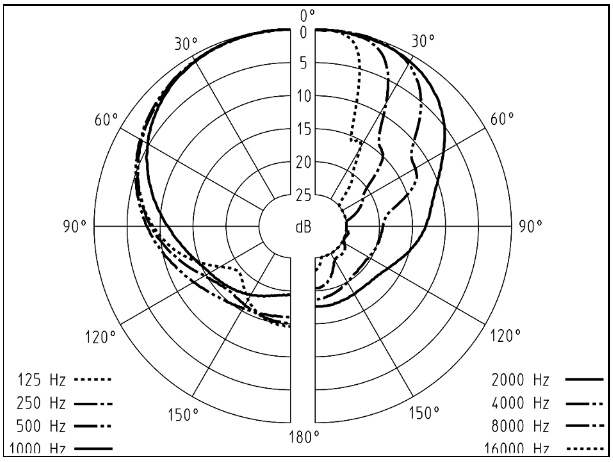


Figure 18: directivity of Sennheiser MKH 70-1

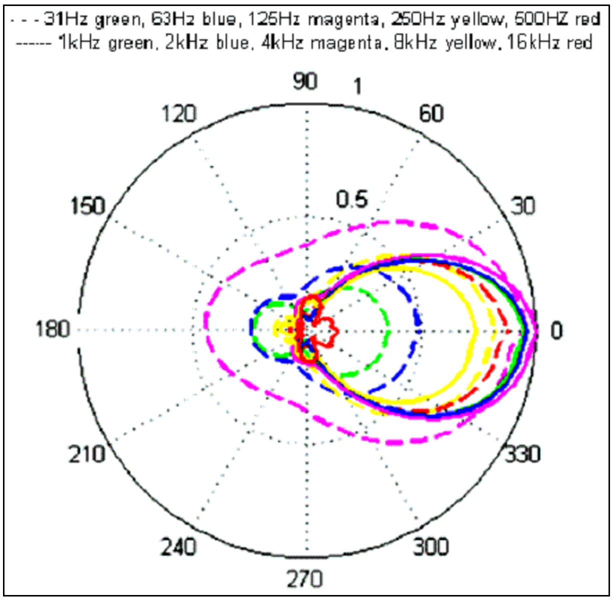


Figure 19: directivity of 6th order cardioid

The shotgun microphone's directivity strongly depends from the frequency. This means that the sources out of the main axis are affected by significant "colouring", in practice they sound as low-passed . The directivity of the virtual microphone is instead constant from the 250 Hz band upwards and this doesn't affect the colour of the sound. The beamwidth for the virtual cardioid corresponds to 60° at -6 dB, whilst the shotgun microphone ranges between 150 degrees (at very low frequency) up to 45 degrees at 16 kHz, but it is indeed more than 60 degrees up to 8 kHz. So our virtual cardioid microphone is much more directive and much less coloured than a state-of-the art shotgun mike.

2 THE REAL-TIME MICROPHONE ARRAY

When transforming the research project described in previous chapters to a microphone system usable in the real world for recording and broadcasting we had to face a number of issues:

- Provide high directivity in order to exclude the undesired noises that usually affect audio recordings in broadcast and film production.
- Provide the capability of changing orientation and directivity in real time by using a controller (for example a joystick).
- The capability of using this system in post production, varying the virtual microphones on recorded audio.
- The capability of seeing the subject on which the virtual microphone is focusing.
- Low latency.
- High audio quality.

In the next paragraphs all the steps performed to realize this kind of system will be shown.

2.1 Hardware

For using a system in broadcast production it should be very robust, without unnecessary complexities; on the other hand, we needed to provide massive computational power for performing all those FIR filters in real-time and with small latency, so we needed to add a dedicated "black-box" containing a very powerful mini-ITX motherboard with a Quad Core processor. All the signal processing is made by this computer. A laptop is used for visual control of the virtual microphone's characteristics over the IP network, a joystick is used for changing in real-time the directivities and orientations.

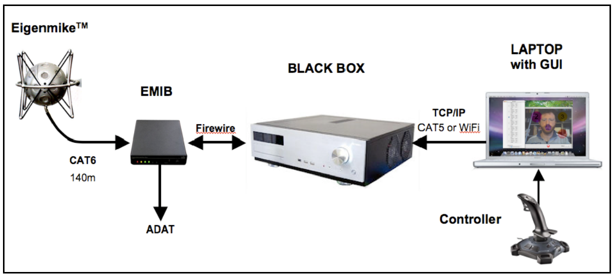


Figure 20: hardware scheme of the system

2.2 Processing software

The most important piece of software for this kind of applications is the convolver: BruteFIR [8] was chosen for its flexibility and for its capability of managing a large number of convolutions. BruteFIR is a Linux software and for this reason Linux (in particular Ubuntu Minimal distribution) was chosen as Operative System. The main problem of Linux when dealing with multichannel audio is the poor availability of drivers for Firewire audio interfaces. Luckily, the EMIB interface is fully supported by FFADO [9].
The audio connection between the different software tools is managed by Jack [10].
The scheme below represents the software architecture for the signal processing inside the "black box".

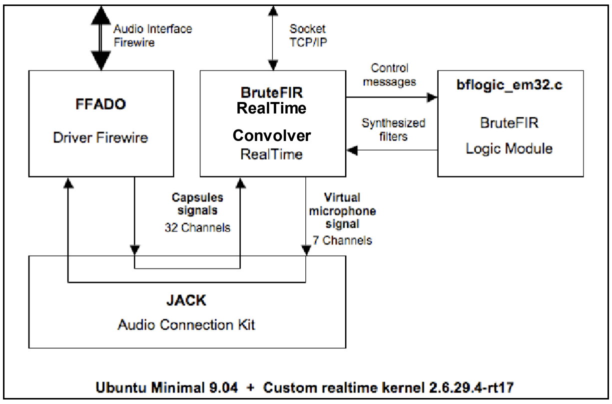


Figure 21: software architecture

2.2.1 *BruteFIR realtime convolver*

BruteFIR permits to implement up to 256 filters and considering that for one single microphone the system needs 32 filters (one for each input channel) we can get 8 virtual microphones. The choice of synthesizing only 7 virtual microphones is due to the need of having one "free slot" (32 filters) dedicated to the "double buffering", the dynamic switch from the old and the new virtual microphone every time the user needs to modify the characteristics of one of the virtual microphones (Figure 22). With BruteFIR, it is also possible to have a cross-fading during the change of the coefficients with the benefit of a glitch-free transition between two different positions of the virtual microphone.

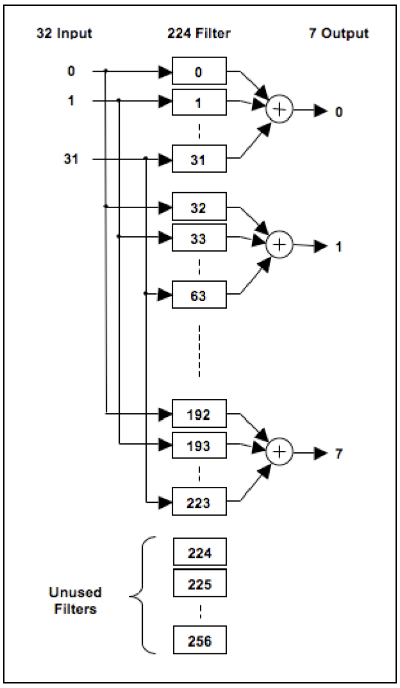


Figure 22: BruteFIR filtering structure.

2.2.2 *BruteFIR logic module*

This module implements the FIR's synthesis process described in the paragraph 1.4, and permits to control 7 independent virtual microphones.
The virtual microphone's parameters can be modified in real time via TCP/IP through a proper protocol. Every time the parameters are modified, the system automatically recalculates the FIR filters and substitutes them in the memory of the convolver through a "double buffering" mechanism.
The filter computation subroutine was rewritten in highly-optimized C++, making it possible to recalculate all the filters in a few milliseconds, so that the virtual microphones can be smoothly moved around without artefacts.

2.3 The graphical user's interface

The graphical user's interface has a lot of work to do:

- it should provide the connection to the black box via SSH protocol;
- it should start the software loaded in the black box unit;
- it should give to the user the vision of the scene in which he has to point the virtual microphones, thanks to a panoramic video camera;
- it should visualize the directivity of the microphones over the scene;
- it should read the movements imposed by the controller (a pointing device such as a joystick, a mouse, or a touchscreen);
- it should communicate to the black box every change of the virtual microphones in terms of direction, directivity and gain.

The GUIs were developed using Python with external libraries for interacting with the pointing device, for connecting via SSH and for the graphical layout.

2.3.1 *GUI for post-processing (playback)*

The first prototype of the graphical interface has the capability to load a 360° photo, made before the performance in the exact place of the microphone. The user has the capability to point different virtual microphones, seeing on the screen the direction of his focus and the directivity, thanks to visual aids projected over the panoramic photo (coloured circles, the larger is the circle, the wider is the directivity). The directivity can be changed continuously from omni to 6th order cardioid using a slider on the joystick; the user can control every single microphone of the 7 available pressing the buttons on the joystick.
This GUI is shown in Figure 23, and it is optimized for working during playback of the 32 channels previously recorded: it is ideal for post production, when the user needs to modify the surround setup, adjusting the directivities in order to find the best result in terms of spatialization and sound quality.

2.3.2 *GUI for real-time microphone steering*

A second possible application for this microphone system is in real-time: the user should be able to follow actors or moving sources, delivering live to the production the corresponding "dry" audio signals. For doing this, the GUI should permit to focus the microphones on a video stream coming from a panoramic surveillance camera, equipped with a 360° parabolic mirror, placed close to the microphone probe.
This GUI is shown in Figure 24. A slider labelled "Transparency" is visible: it is useful to adjust the transparency of the coloured pointers overlaid on the live video image.

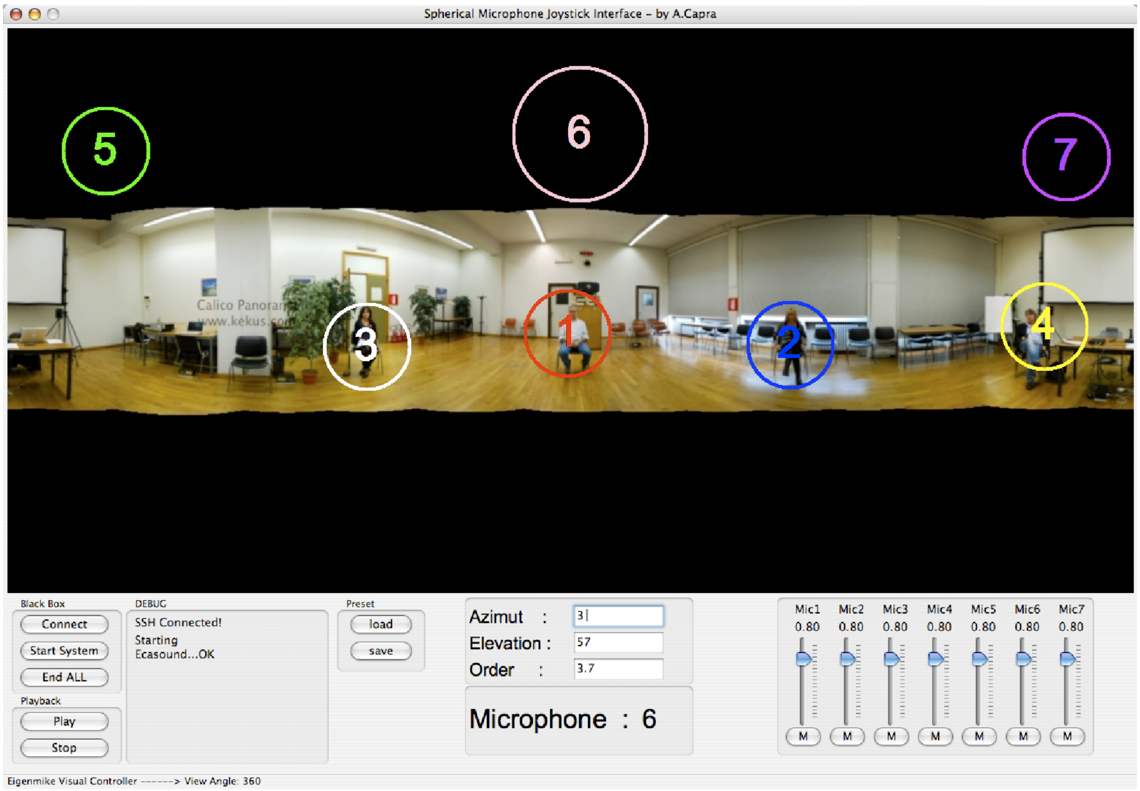


Figure 23: GUI with a panoramic photo

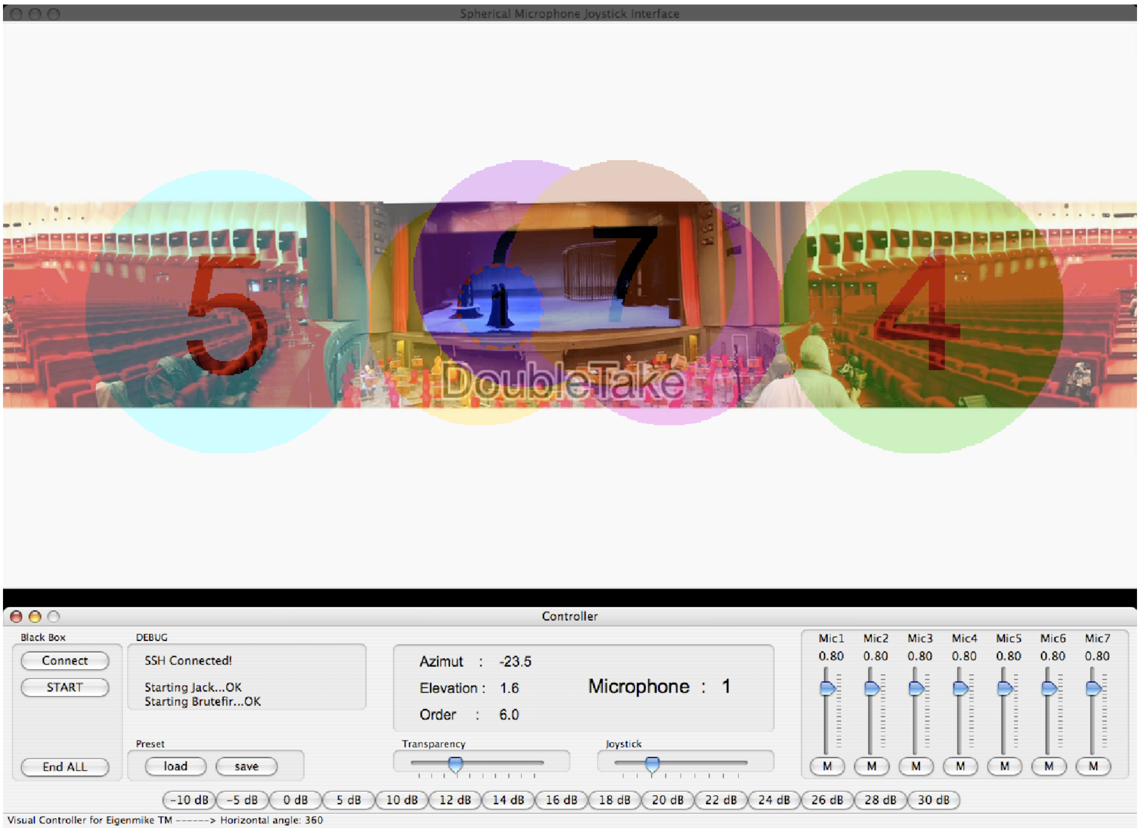


Figure 24: GUI for realtime processing overlaid on a live video stream

2.4 Performances of the system

The evaluation of the performances is done considering several factors:

- The total latency of the processed audio signal
- The rate of virtual microphone's updates
- The frequency response of the synthesized microphones
- The S/N ratio, particularly at low frequency.

2.4.1 The total latency

The total latency of the processing system is due to two different factors:

- The latency of the FIRs, with a delay typically around N/2 samples (for their causalization).
- The latency introduced by the convolution system and buffering

The system is stable using FIR filters of length 1024 or 2048, partitioned in blocks of 256 samples (and, consequently, operating FFTs of 512 samples for performing fast convolution).

Latency of the FIRs $N/2$	Latency due to the processing system (frame size × buffer number)	Total latency of the processed signal
1024 / 2	256 × 2	23 *ms*
1024 / 2	512 × 2	34 *ms*
2048 / 2	256 × 2	34 *ms*
2048 / 2	512 × 2	46 *ms*

Table 1: Performances in terms of latency.

This latency is a good result for a broadcasting application, because it is low enough for the sync between audio and video. It must be considered that in many cases the video is lagging behind the audio stream, and this amount of latency is approximately the “right one” for re-synching audio and video together.

2.4.2 The update rate

The update rate of the virtual microphones strictly depends on the time used by the logic module that synthesizes the FIR filters for the virtual microphones being changed by the user.

Filter's length	**Time for the synthesis of 1 virtual mike**	**Time for the synthesis of 7 virtual mike**
1024	70 *ms*	460 *ms*
2048	140 *ms*	920 *ms*

Table 2: Time required for virtual microphones updates.

The most critical applications, in terms of time for the update, are the ones in which the user tries to follow a sound source that is moving changing continuously the direction of the virtual microphone: in this case the frequency of update is 14 times for seconds, sufficient to guarantee a quick steering, even if the changes in direction are abrupt.

2.4.3 Frequency Response

The measured frequency responses for various types of virtual microphones synthesized with our array are shown in Figure 14 to Figure 17. As the regularization parameter β was set to a small value in the frequency range 100 Hz to 10 kHz, we find that the frequency response is almost perfectly flat within this range, even for highly-directive 6^{th}-order cardioids (Figure 16).

Of course one could extend the equalized frequency range, provided that also the length of the processing filters is augmented (this is particularly true if one wants to extend the range towards frequencies below 100 Hz).

On the other hand, as explained in the following, the attempt to extend too much the frequency range tends to boost the noise at low frequency for high-order directive microphones.

2.4.4 Noise

Synthesizing high-directivity virtual microphones starting from a small size array is prone to introduce significant noise at low frequency. This problem is particularly evident with High Order Ambisonics processing, where the high-order spherical harmonics signals tend to have an acceptable signal-to-noise ratio only in a very limited frequency range.

As we are not passing through spherical harmonics signals, this problem is partially alleviated. Also setting an optimal preamplifier gain on the preamplifiers improves significantly the S/N ratio.

Indeed, it was necessary to limit the frequency range where the regularization parameter is kept at a low value, for avoiding too loud low-frequency noise.

Many experiments were conducted for finding the optimal compromise between spatial directivity, flatness of the frequency response and S/N ratio.

Figure 25 shows the 1/3 octave spectrum of background noise of a virtual 6^{th}-order cardioid, measured with the preamplifiers set to a +20 dB gain, which corresponds to a digital clipping level of approximately 107 dB-SPL-peak, compared with the “intrinsic” background noise level of one of the 32 capsules (they all measure very similar, as the manufacturer closely matches the selected capsules).

The gain of the virtual microphone was calibrated for having the same sensitivity, along its axis, as a single capsule. The measurement was performed inside a large hemi-anechoic room (University of Ferrara), with very low acoustical background noise (17 dBA, compliant to NR-10 curve).

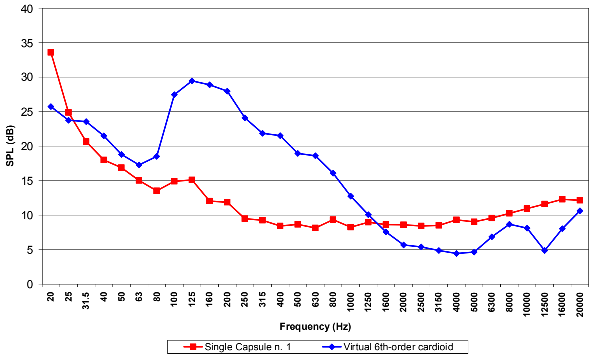


Figure 25: Background noise at +20 dB gain

As the regularization parameter β was set to a small value only above 100 Hz, the noise of the virtual microphone reaches the maximum at 125 Hz, then decreases quickly, resulting lower than the noise of a single capsule above 1250 Hz.
The overall A-weighted value of the virtual microphone is approximately 26 dB(A). These performances are exceptionally good for a 6^{th}-order cardioid, considering that the inherent background noise of a single capsule is approximately 21 dB(A).
Of course the low-frequency noise increases if the spatial filtering is forced to be accurate at frequencies much lower than 100 Hz: the choice of 100 Hz as the corner frequency, above which the directivity pattern is forced to be accurate, revealed to be an optimal choice in terms of usability of the virtual microphones and good signal-to-noise ratio.
It must be noticed that this does not means that the virtual microphone is deaf below 100 Hz: as shown in Figure 16, the virtual microphone still provides significant output also below 100 Hz, but the directivity pattern is not as sharp as at higher frequencies.

3 CONCLUSIONS

The goal of this research project was the realization of a new microphone system, capable of synthesizing a number of virtual microphones, dynamically changing their aiming and directivity.
The polar patterns of the virtual microphones can be varied continuously from standard types (i.e. omni or cardioid) up to very directive 6^{th} order cardioids, which resulted narrower than top-grade "shotgun" mikes.
These virtual microphones are obtained processing the signals coming from a 32-capsules spherical microphone array, employing digital filters derived from a set of impulse response measurements. This provides several benefits:

- Widest possible frequency range for a given size of the probe, with low noise;
- Inherent correction for the dissimilarities between the transducers;
- Partial correction also for acoustic artefacts, such as shielding effects, diffractions and resonances.

The finalized system provides a revolutionary approach to sound capture in broadcasting and film/music productions.

ACKNOWLEDGEMENTS

The authors are warmly thankful to Eighteen Sound for providing the anechoic room and the equipment employed for the experimental characterization of the probe.
MH Labs is acknowledged for their fundamental help, allowing for hacking the gain-control protocol and for interfacing their microphone to "unusual" hardware and software processing tools.

REFERENCES

[1] A.J. Berkhout, D. de Vries, and P. Vogel, *Acoustic control by wave field synthesis*, Journal of the Acoustic Society of America, 93(5):2764–2778, May 1993.

[2] S. Moreau, J. Daniel, S. Bertet, *3D sound field recording with High Order Ambisonics - objective measurements and validation of a 4th order spherical microphone,* 120th AES Convention, Paris, France - May 20-23, 2006.

[3] F. M. Fazi, P. A. Nelson, *The ill-conditioning problem in Sound Field Reconstruction*, 123rd AES Convention, New York, NY, USA - October 5/8, 2007.

[4] http://www.mhacoustics.com

[5] O. Kirkeby, P. A. Nelson, *Digital Filter Design for Inversion Problems in Sound Reproduction*, AES, vol. 47, no. 7/8 (1999 July/August).

[6] O. Kirkeby, P. Rubak, A. Farina, *Analysis of ill-conditioning of multi-channel deconvolution problems*, 106th AES Convention, Munich, Germany - May 8-11, 1999.

[7] A. Farina, *Simultaneous measurement of impulse response and distortion with a swept-sine technique*, 110th AES Convention, Paris 18-22 February 2000.

[8] http://www.ludd.luth.se/~torger/brutefir.html

[9] http://www.ffado.org/

[10] http://jackaudio.org

[11] A. Farina, A. Capra, L. Conti, P. Martignon, F. M. Fazi, *Measuring spatial impulse responses in concert halls and opera houses employing a spherical microphone array*, 19th International Congress on Acoustics (ICA), Madrid, 2007.