DEVELOPMENT AND EVALUATION OF POCKET-SIZE BLIND SOURCE SEPARATION MICROPHONE

[1]*T. Hiekata,* [1]*Y. Ikeda,* [1]*T. Yamashita,* [1]*T. Morita,* [2]*R. Zhang,* [3]*Y. Mori,* [3]*H. Saruwatari,* [3]*K. Shikano,*

[1]Kobe Steel,Ltd., 1-5-5 Takatsukadai, Nishi-ku, Kobe, Hyogo, 651-2271, Japan
[2]Feng Co.,Ltd., Hyogo, 670-0995, Japan
[3]Nara Institute of Science and Technology, Nara, 630-0101, Japan

ABSTRACT

We develop a new blind source separation (BSS) microphone named SSM-001 which can separate multiple sounds in real-time under noisy conditions. The BSS microphone is based on our previously proposed BSS algorithm which combines a Single-Input Multiple-Output (SIMO)-model based BSS and SIMO-model based binary masking. We modify this algorithm and implement it to DSP for the purpose of more effective and realistic real-time operation. In this paper, the issue of real-time implementation in the BSS microphone is illustrated in detail, and the experimental evaluations of the hardware reveal the proposed BSS microphone's efficacy.

***Index Terms*—** DSP, blind source separation, real-time implementation

1. INTRODUCTION

Real-time separation of target sound and noises is demanded for many applications, e.g., speech dialogue systems, cellular phones, and car navigation systems. Blind source separation (BSS) is beneficial to this purpose because BSS is a flexible approach to estimate original source signals using only the information of the mixed signals observed in each input channel. We have recently proposed a novel two-stage BSS algorithm [1] which combines (a) Single-Input Multiple-Output (SIMO)-model-based ICA (SIMO-ICA) [2] and (b) SIMO-model based binary masking [3] (SIMO-BM) applied to the SIMO-ICA's outputs. Also we continue to challenge the development of a BSS device.

In this paper, first, we mainly report an issue of real-time BSS implementation on hardware, which yields a new *pocket-size BSS microphone named SSM-001*. Several recent research studies [4] have dealt with real-time implementation of ICA, but still required high-speed personal computers. Consequently BSS implementation on a small-size LSI still receives much attention in industrial applications, so our microphone is equipped with a floating-point small-size Digital Signal Processor (DSP), and we implement our two-stage BSS algorithm to the DSP. Next we give extensive evaluations of SSM-001 from the viewpoint of sound quality, real-time separation performance, and polar pattern. From these results, we can show the efficacy of the developed BSS microphone.

2. REAL-TIME BSS MICROPHONE SSM-001

Figure 1 shows a picture of BSS Microphone SSM-001. Also the main specifications are listed in Table 1. The hardware block diagram is depicted in Fig. 2, and the picture of the internal board is shown in Fig. 3. As can be seen, SSM-001 is one of the world's smallest BSS microphone miniaturized into pocket-size hardware. We implemented two-stage BSS algorithm to the BSS microphone, which is based on SIMO-ICA and SIMO-BM [1]. The configuration of two-stage BSS is depicted in Fig. 4. SIMO-BM which follows SIMO-ICA can remove the residual leakage without adding huge computations. Due to this paper's main focus on a real-time implementation and space limitation, we would skip the detailed description and proof of the algorithm (see [1] for more information).

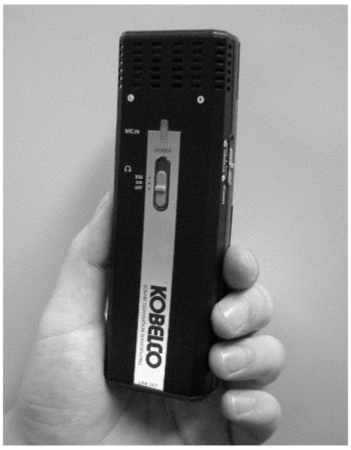


Fig. 1. BSS microphone (SSM-001).

Table 1. Specifications of BSS microphone (SSM-001)

processor	TI-DSP TMS320VC6727 (Clock 300 MHz)
input	internal stereo mic
	external stereo mic (plug-in power)
output	line out
	headphone out
terminal	external mic in
	lint out
	headphone out
sampling frequency	16 kHz or 8 kHz
battery	AA battery × 2
memory	Flash ROM: 8MByte (used about 330 KByte)
	SDRAM: 128MByte (used about 1.9 MByte)
size	136 mm (H) × 45 mm (W) × 27 mm (D)
weight	about 125 g (including battery)

Figure 5 shows a configuration of a *real-time implementation* for the two-stage BSS, and Fig. 6 shows DSP and SDRAM internal block diagram from the viewpoint of software. Signal processing in this implementation is performed in the following manner.

[Step1] Input signals are converted to time-frequency series $\boldsymbol{X}(f,t)$ by using a frame-by-frame fast Fourier transform (FFT), where $\boldsymbol{X}(f,t) = [A_{11}(f)S_1(f,t) + A_{12}(f)S_2(f,t), A_{21}(f)S_1(f,t) + A_{22}(f)S_2(f,t)]^{\mathrm{T}} + \boldsymbol{N}(f,t)$ ($A_{kl}(f)$ is the mixing matrix corresponding to room transfer function, and $\boldsymbol{N}(f,t)$ is the additive noise term).

This operation in DSP (Fig. 6) is as follows: Audio signals are

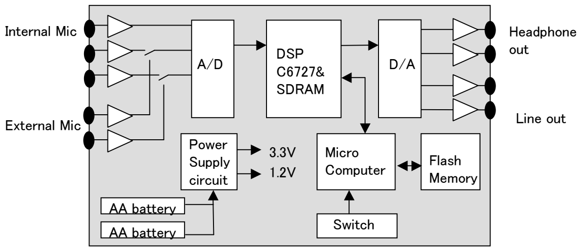


Fig. 2. Hardware block diagram of BSS microphone.

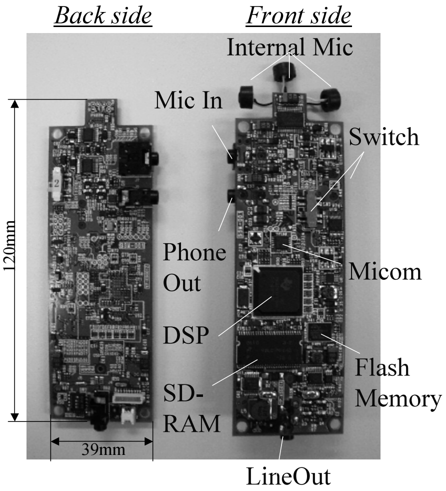


Fig. 3. Internal board of BSS microphone.

input via *McASP* (Multi-channel Audio Serial Port), and sent to an input buffer by *DMA* (Direct Memory Access) function. If the input buffer is filled, *DMA* generates an interrupt, and it wakes up *DMA_isr* routine. The *ICA_filter_task* is called in *DMA_isr* routine, then input signals in input buffer are converted to time-frequency series in the *ICA_filter_task*.

[Step2] SIMO-ICA is conducted using current data samples of a few seconds duration (T_s in Fig. 5) for estimating the separation matrix, which is applied to next *(not current)* duration samples as soon as the optimization of SIMO-ICA filter is finished. The iterative calculation in SIMO-ICA's ICA part is given as

$$\begin{aligned}
&\boldsymbol{W}^{[j+1]}_{(\mathrm{ICA1})}(f) \\
&= \boldsymbol{W}^{[j]}_{(\mathrm{ICA1})}(f) - \alpha\Bigg[\Bigg\{\text{off-diag}\left\langle \boldsymbol{\Phi}(\boldsymbol{Y}^{[j]}_{(\mathrm{ICA1})}(f,t)) \boldsymbol{Y}^{[j]}_{(\mathrm{ICA1})}(f,t)^{\mathrm{H}}\right\rangle_t\Bigg\}\cdot \boldsymbol{W}^{[j]}_{(\mathrm{ICA1})}(f) \\
&\quad - \Bigg\{\text{off-diag}\left\langle \boldsymbol{\Phi}(\boldsymbol{X}(f,t)-\boldsymbol{Y}^{[j]}_{(\mathrm{ICA1})}(f,t)) \cdot (\boldsymbol{X}(f,t)-\boldsymbol{Y}^{[j]}_{(\mathrm{ICA1})}(f,t))^{\mathrm{H}}\right\rangle_t\Bigg\} \\
&\quad \cdot (\boldsymbol{I}-\boldsymbol{W}^{[j]}_{(\mathrm{ICA1})}(f))\Bigg],
\end{aligned} \tag{1}$$

where α is the step-size parameter, and $\boldsymbol{\Phi}(\cdot)$ is the appropriate nonlinear vector function. We obtain the following solutions $\boldsymbol{Y}_{(\mathrm{ICA}l)}(f,t) = [Y_1^{(\mathrm{ICA}l)}(f,t),\ Y_2^{(\mathrm{ICA}l)}(f,t)]^{\mathrm{T}}$ ($l = 1,\ 2$) via SIMO-ICA without considering the permutation effect (see [2] for the proof).

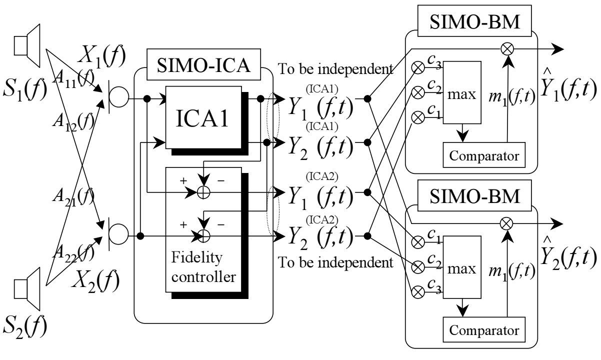


Fig. 4. Two-stage BSS algorithm.

$$\begin{aligned}
\boldsymbol{Y}_{(\mathrm{ICA1})}(f,t) &= \boldsymbol{W}_{(\mathrm{ICA1})}(f)\boldsymbol{X}(f,t) \\
&= [A_{11}(f)S_1(f,t),\ A_{22}(f)S_2(f,t)]^{\mathrm{T}} + \boldsymbol{E}_1, &(2) \\
\boldsymbol{Y}_{(\mathrm{ICA2})}(f,t) &= (\boldsymbol{I} - \boldsymbol{W}_{(\mathrm{ICA1})}(f))\boldsymbol{X}(f,t) \\
&= [A_{12}(f)S_2(f,t),\ A_{21}(f)S_1(f,t)]^{\mathrm{T}} + \boldsymbol{E}_2, &(3)
\end{aligned}$$

where $\boldsymbol{E}_i$ is the residual errors. Since each output of SIMO-ICA is again approximate to array signal at microphone positions, so we can concatenate an appropriate post-processing for eliminating the residual errors $\boldsymbol{E}_i$.

This operation in DSP (Fig. 6) is as follows: The filter update in SIMO-ICA is executed in an independent task named *OptimizeW_task*. The *OptimizeW_task*, which is called by *PRD* (Periodic Function Manager), checks the *fft buffer* periodically, to which the *ICA_filter_task* also sends time-frequency series. When fft buffer is filled, the *OptimizeW_task* begins to optimize the filter $W(f)$, and updates them in *W buffer*. The filter update in SIMO-ICA requires substantial computational complexities, but we realized to finish this process using 3-s-duration data at most 100 iterations within 0.5 seconds (corresponding to T_l in Fig. 5). Therefore, the filter update involves totally a latency of only within 3.5 seconds in case of using 3-s-duration.

[Step3] SIMO-BM is applied to the separated signals obtained by the previous SIMO-ICA. Unlike SIMO-ICA, binary masking can be conducted just in the current segment. The resultant output signal corresponding to the source 1 is determined in the proposed SIMO-BM as follows:

$$\hat{Y}_1(f,t) \quad = \quad m_1(f,t)Y_1^{(\mathrm{ICA1})}(f,t), \tag{4}$$

where $m_1(f,t)$ is the *SIMO-model-based* binary mask operation which is defined as $m_1(f,t) = 1$ if

$$\begin{aligned}
&Y_1^{(\mathrm{ICA1})}(f,t) \\
&\quad > \max[|c_1 Y_2^{(\mathrm{ICA2})}(f,t)|,\ |c_2 Y_1^{(\mathrm{ICA2})}(f,t)|,\ |c_3 Y_2^{(\mathrm{ICA1})}(f,t)|];
\end{aligned} \tag{5}$$

otherwise $m_1(f,t) = 0$. Here c_i is the parameter to balance the separation ability and the sound quality. Typically $c_1 = 1$, $c_2 = 0$, and $c_3 = 0 \sim 1$. The source 2 is obtained by the same manner.

[Step4] The output signals from SIMO-BM are converted into the resultant time-domain waveforms by using an inverse FFT. SIMO-BM and this process are executed in the *ICA_filter_task* in Fig. 6.

Although the separation filter update in the SIMO-ICA part is not real-time processing but includes a latency of a few seconds,

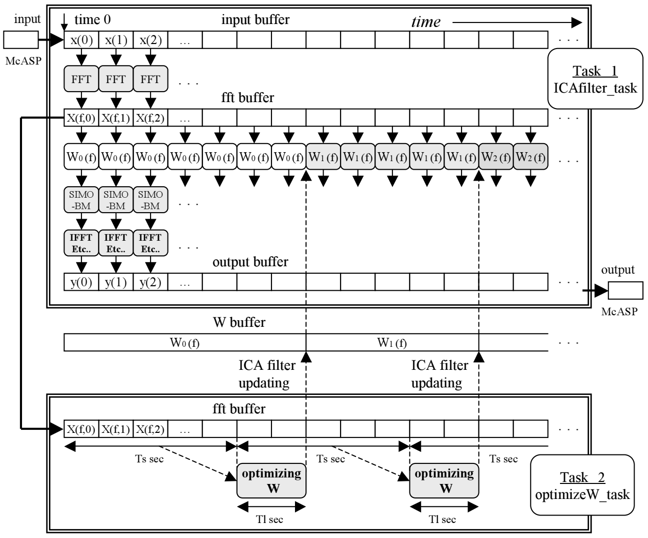


Fig. 5. Signal flow in real-time implementation.

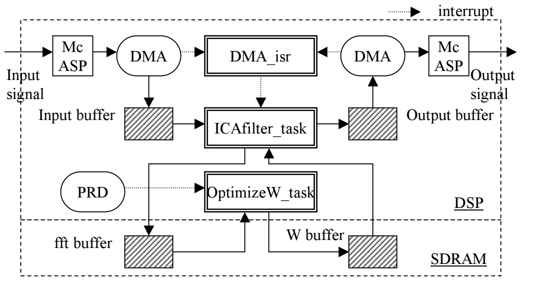


Fig. 6. DSP & SDRAM internal block diagram.

the entire two-stage system still seems to run in real-time because SIMO-BM can work in the current segment with no delay. Generally, the latency in conventional ICAs is problematic and reduces the applicability of such methods to real-time systems. In this method, however, the performance deterioration due to the latency in SIMO-ICA can be mitigated by introducing real-time binary masking.

3. SOUND QUALITY EVALUATION

To grasp the BSS microphone's basic behavior, we measure the following three scores. First, *noise reduction rate (NRR)* [5], defined as the output signal-to-noise ratio (SNR) in dB minus the input SNR in dB, is evaluated as the objective indication of degree of interference reduction (we don't take into account the sound distortion). Secondly we measure *cepstral distortion (CD)* which indicates the distance between the spectral envelope of the original source signal and the target component in the separated output (CD doesn't take into account the degree of interference reduction unlike NRR). Thirdly we score *PESQ MOS-LQO* (ITU-T Recommendation P.862.1) which is comparable with *Mean Opinion Score (MOS)*, and corresponds to the subjective indication of sound quality related to both NRR and CD.

In order to simulate the sound mixing, many impulse responses was measured in 200-ms reverberant room, and convolved with dry speech signals. Two speech signals are assumed to arrive from different directions, $(\theta_1, \theta_2) = (0°, ..., 90°, -90°, ..., 0°)$, and $(\theta_1, \theta_2) = (-90°, ..., 0°, 0°, ..., 90°)$ in intervals of 10°. Two types of sentences, spoken by two male and two female speakers selected from ASJ continuous speech corpus for research, are used as the original speech samples. Using these sentences, we obtain 12 combinations with respect to speaker pair for each (θ_1, θ_2). The sampling frequency is 8 kHz. This experiment is noise free, i.e, we do not consider an additional background noise. In ICA part, we use a null-beamformer-based initial value [5] which is steered to $(-60°, 60°)$.

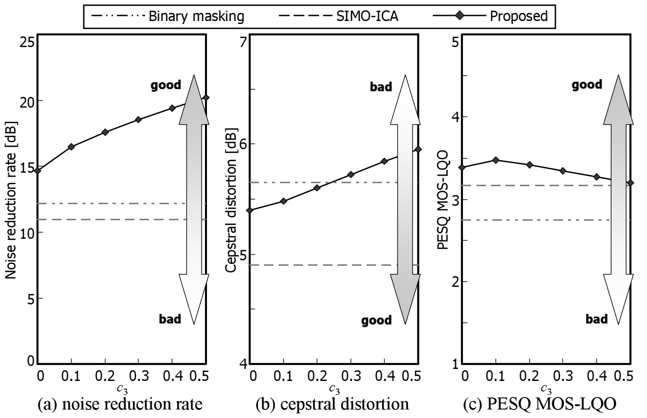


Fig. 7. Quality evaluation: (a) noise reduction rate, (b) cepstral distortion, and (c) PESQ MOS-LQO.

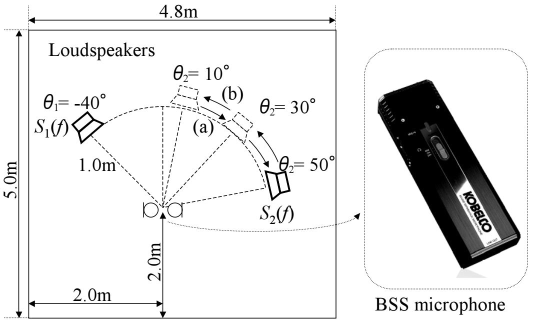


Fig. 8. Layout of reverberant room used in experiments.

Figure 7 shows the measurement results, where Fig. 7(a) shows NRR, Fig. 7(b) shows CD, and, Fig. 7(c) shows PESQ MOS-LQO. These results are average values of all the combinations of θ_1 and θ_2. We compare three methods as follows: (A) the conventional binary-masking, (B) the conventional SIMO-ICA, and (C) the proposed two-stage BSS method (ICA part in (B) or (C) uses 3-s-duration buffering for estimating the separation matrix). From these results, we can confirm that NRR can be improved as the c_3 parameter increases, but CD results in a larger value, i.e., the sound quality becomes worse. Based on the above-mentioned tradeoff, $c_3 = 0.1$ is the best parameter for PESQ MOS-LQO. This means that slight gain of c_3 provides the best sound quality for human hearing. Note that, from our preliminary experiments, c_3 should be increased to more than 0.1 especially under more noisy conditions.

4. EXPERIMENTS IN REAL-TIME PROCESSING

4.1. Separation performance for moving sound

In this section, a real-recording-based BSS experiment is performed using the BSS microphone in a real acoustic environment (see Fig. 8), where two loudspeakers and the BSS microphone are set. The reverberation time in this room is 200 ms. Two speech signals, whose length is limited to 32 seconds, arrive from different directions, θ_1 and θ_2, where we fix source 1 in $\theta_1 = -40°$, and move source 2 under two conditions in Table 2. Two types of sentences, spoken by two male and two female speakers selected from the

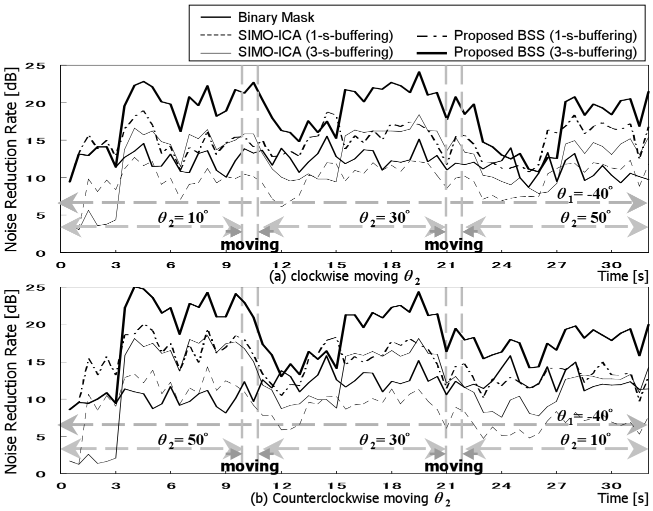


Fig. 9. Example of segmental noise reduction rate calculated along time axis at every 0.5 s period, using real recording data and real-time BSS.

Table 2. Moving conditions for sound source 2 in θ_2

θ_2	$0 \rightarrow 10\ sec$	$10 \rightarrow 11\ sec$	$11 \rightarrow 21\ sec$	$21 \rightarrow 22\ sec$	$22 \rightarrow 32\ sec$
(a)	10°	10° → 30°	30°	30° → 50°	50°
(b)	50°	50° → 30°	30°	30° → 10°	10°

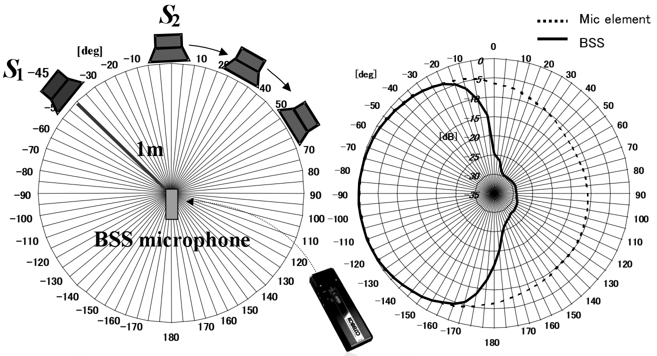
ASJ continuous speech corpus for research, are used as the original speech samples. Using these sentences, we obtain 12 combinations with respect to speakers and source directions. Average of SNR between each speech signal and background noise is 26 dB.

We compare five methods as follows: (A) the conventional binary masking, (B) the conventional SIMO-ICA with 1-s-duration buffering, (C) the conventional SIMO-ICA with 3-s-duration buffering, (D) the proposed BSS with 1-s-duration buffering, and (E) the proposed BSS with 3-s-duration buffering. In the proposed BSS method, we set $[c_1, c_2, c_3]$ = $[1, 0, 0.5]$, which gives the best performance (high NRR but low distortion) under this background noise condition.

Figure 9 shows the averaged segmental NRR for 12 speaker combinations, which was calculated along the time axis at every 0.5 s period. Figure 9(a) shows the clockwise-moving result of NRR, and Fig. 9(b) shows the counterclockwise-moving result of NRR. Both Figs. 9(a) and (b) show that the proposed BSS with 3-s-duration buffering outperforms binary masking and the conventional SIMO-ICA at almost all the time during 3–32 s. The difference between proposed BSSs with 1- and 3-s-duration bufferings is that 1-s-duration buffering can provide slightly rapid improvements in the early period and just after source moving, but the performance for the static sources is quite low compared with 3-s-duration buffering. This is due to that 1-s-duration data is too short to evaluate the sources' statistical independence. From these results, hereafter, we introduce 3-s-duration buffering for the two-stage BSS implemented in SSM-001.

4.2. Polar pattern of separation performance

In this section, we show the polar pattern of the BSS microphone's separation performance to visually indicate the proposed BSS's spatial-acoustical behavior. Figure $10(a)$ shows the measurement condition. We carried out real-time sound separation using source signals recorded in the real room where two loudspeakers and the real-time BSS microphone are set. Two speech signals with 30-s length arrive from different directions, θ_1 and θ_2, where we fix source 1 in $\theta_1 = -45°$, and arrange source 2 from $\theta_2 = 0°$ to $\theta_2 = 360°$ in intervals of $5°$. We set $[c_1, c_2, c_3]$ = $[1, 0, 0.5]$, and evaluate NRR considering that source 1 is the target and source 2 is noise.



(a) Measurement condition (b) Measurement results

Fig. 10. Polar pattern of separation performance: (a) measurement condition, and (b) measured result.

Figure $10(b)$ depicts the measured results. We can confirm that our BSS microphone has a great noise-reduction ability in the right-side area ranging $\theta_2 = -5 \sim -185°$. On the other hand, the BSS microphone is not proficient in reducing noise (source 2) in the left-side area. This is due to that (a) ICA part cannot separate the narrow-angle sources accurately, and (b) binary masking cannot work well because we do not employ the stereo microphone's directivity (i.e., power difference); both sources located in left-side area could not give an apparent signal power difference. This problem remains as our future work, and we have proposed an improved algorithm to handle the disadvantage [6].

5. CONCLUSION

We introduced and evaluated a new BSS microphone named SSM-001 which can separate a target sound in a noisy environment in real-time. We revealed that the BSS microphone has unprecedented performance. This motivates us to hope that the technologies of the BSS microphone will be adopted by many applications in the future.

6. REFERENCES

[1] Y. Mori, T. Takatani, H. Saruwatari, K. Shikano, T. Hiekata and T. Morita, "Blind source separation combining SIMO-ICA and SIMO-model-based binary masking", *Proc. ICASSP2006*, pp.V-81–84, 2006.

[2] T. Takatani, T. Nishikawa, H. Saruwatari and K. Shikano, "High-fidelity blind separation of acoustic signals using SIMO-model-based ICA with information-geometric learning," *Proc. IWAENC2003*, pp.251–254, 2003.

[3] R. Lyon, "A computational model of binaural localization and separation," *Proc. ICASSP83*, pp.1148–1151, 1983.

[4] H. Buchner, R. Aichner, and W. Kellermann, "A generalization of blind source separation algorithms for convolutive mixtures based on second order statistics," *IEEE Trans. Speech Audio Processing*, vol.13, no.1, pp.120–134, 2005.

[5] H. Saruwatari, S. Kurita, K. Takeda, F. Itakura, T. Nishikawa and K. Shikano, "Blind source separation combining independent component analysis and beamforming," *EURASIP Journal on Applied Sig. Process.*, vol.2003, pp.1135–1146, 2003.

[6] Y. Mori, H. Saruwatari, K. Shikano, T. Hiekata and T. Morita, "Directivity-dependency-reduced blind source separation integrating ICA, beamforming and binary masking", *Proc. IROS2007*, 2007.