Proceedings of the 2006 IEEE/RSJ
International Conference on Intelligent Robots and Systems
October 9 - 15, 2006, Beijing, China

Real-Time Robot Audition System That Recognizes Simultaneous Speech in The Real World

Shun'ichi Yamamoto*, Kazuhiro Nakadai†, Mikio Nakano†, Hiroshi Tsujino†, Jean-Marc Valin‡,
Kazunori Komatani*, Tetsuya Ogata*, and Hiroshi G. Okuno*

* Graduate School of Informatics, Kyoto University, Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501, Japan
†Honda Research Institute Japan Co., Ltd., 8 - 1 Honcho, Wako-shi, Saitama 351-0114, Japan
‡CSIRO ICT Centre, Cnr Vimiera & Pembroke Rds, Marsfield NSW 2122, Australia

{shunichi, komatani, ogata, okuno}@kuis.kyoto-u.ac.jp,
{nakadai, nakano, tsujino}@jp.honda-ri.com, jean-marc.valin@usherbrooke.ca

Abstract— **This paper presents a robot audition system that recognizes simultaneous speech in the real world by using robot-embedded microphones. We have previously reported Missing Feature Theory (MFT) based integration of Sound Source Separation (SSS) and Automatic Speech Recognition (ASR) for building robust robot audition. We demonstrated that a MFT-based prototype system drastically improved the performance of speech recognition even when three speakers talked to a robot simultaneously. However, the prototype system had three problems; being offline, hand-tuning of system parameters, and failure in Voice Activity Detection (VAD). To attain online processing, we introduced FlowDesigner-based architecture to integrate sound source localization (SSL), SSS and ASR. This architecture brings fast processing and easy implementation because it provides a simple framework of shared-object-based integration. To optimize the parameters, we developed Genetic Algorithm (GA) based parameter optimization, because it is difficult to build an analytical optimization model for mutually dependent system parameters. To improve VAD, we integrated new VAD based on a power spectrum and location of a sound source into the system, since conventional VAD relying only on power often fails due to low signal-to-noise ratio of simultaneous speech. We, then, constructed a robot audition system for Honda ASIMO. As a result, we showed that the system worked online and fast, and had a better performance in robustness and accuracy through experiments on recognition of simultaneous speech in a noisy and echoic environment.**

Index Terms— **missing feature theory, robot audition, voice activity detection, real-time processing, parameter optimization, genetic algorithm**

I. Introduction

Speech recognition is essential in communication, and people with normal hearing capabilities can listen to many kinds of sounds under various conditions. For symbiosis between a robot and people in the real world, the robot should have hearing capability equivalent to people's. When several people talk with a robot in a daily scene, they can speak from any position. Therefore, the robot that has microphones embedded in the head and/or body, should cope with a mixture of speech signals originating from various directions and distances.

To deal with such situation, *Sound Source Localization (SSL)*, *Sound Source Separation (SSS)*, and *Automatic Speech Recognition (ASR)* of separated sounds are essential functions for robot audition. Robot audition has been studied actively for recent years, as typified by organized sessions on robot audition at IROS 2004 and IROS 2005. However, they mainly focused on SSL and SSS. Only a few researchers and research groups reported recognition of separated speech. Hara *et al.* reported *HRP-2* which used a microphone array to localize and separate a mixture of sounds, and recognized speech commands in a noisy environment [1]. They assumed only a single speech signal. Nakadai *et al.* reported *SIG* which used a pair of microphones to separate multiple speech signals by *Active Direction-Pass Filter (ADPF)* and recognized each separated speech by ASR [2]. They demonstrated that even when three speakers utter words at the same time, the robot recognized what each speaker said. However, since their system used 51 acoustic models trained under different conditions at the same time, the system requires a high computational cost and deteriorated under an environment with unexpected and/or dynamically changing noises. To deal with simultaneous speech under such an environment, we introduced an interfacing scheme between SSS and ASR based on *Missing Feature Theory (MFT)*. It improves recognition performance by using *missing feature masks (MFM)* which cover unreliable acoustic features used in ASR [3].

MFT is a popular approach for noise-robust ASR. Most reports assumed a single speech signal obtained from a single channel in a noisy environment [4], [5]. However, it was difficult to estimate MFM by a single channel approach, since a general model of noise cannot be assumed in the real world. If SSS by multi-channel input is used, the system can utilize information on interference from other sources to estimate MFM. We developed an automatic MFM generator by using SSS based on *Geometric Source Separation (GSS)* and a multi-channel post-filter with an 8ch microphone array. The automatic MFM generator used the information provided by SSS.

However, the prototype system had three problems;

1) offline,
2) hand-tuning of system parameters, and
3) failure in *Voice Activity Detection (VAD)*.

1-4244-0259-X/06/$20.00 ©2006 IEEE

To attain online processing, we introduced FlowDesigner-based architecture to integrate SSL, SSS, and ASR. This architecture brings fast processing and easy implementation because it provides a simple framework of shared object based module integration. To optimize the parameters, we developed parameter optimization with a *Genetic Algorithm (GA)* [6], because the system parameters are mutually dependent on each other, and it is difficult to build an analytical model to optimize such parameters. To improve VAD, we integrated new VAD based on a power spectrum and location of a sound source into the system, since VAD relying only on power often fails due to a low signal-to-noise ratio (SNR) of simultaneous speech.

The rest of the paper is organized as follows: Section II presents issues and approaches in our robot audition system. Section III explains the implementation of the robot audition system. Section IV describes evaluation of the system, and the last section concludes the paper.

II. Recognition System of Simultaneous Speech Signals

This section shows issues in our robot audition system and our approaches. Our robot audition system had three issues; First, it worked under an off-line environment. Second, eleven parameters required by the system were not optimized properly. Finally, because of the leakage of energy from other channels and remaining noises, ASR often failed to detect voice activity.

A. Implementation for online processing

Our robot audition system did not work online since all components were not integrated into one system, that is, it was an offline system. A large amount of data is communicated in the system. The processing speed would be faster when modules in the system are linked as one object, because data communication is achieved only by communicating a pointer on a shared memory. However, it is better that the modules are as independent as possible in terms of easy implementation. To maintain fast processing time and re-usability of modules, we introduced FlowDesigner architecture [7].

FlowDesigner is a free (GPL/LGPL) data flow oriented development environment. Six modules in the system – Sound Source Localization (SSL), Sound Source Separation (SSS), Parameter Selection, Voice Activity Detection (VAD), Acoustic Feature Extraction, Automatic Missing Feature Mask Generation shown in Fig. 3 – are implemented as module blocks on FlowDesigner. When two blocks have matching interfaces, they are able to be connected regardless of their internal processes. One-to-many and many-to-many connections are also possible. Thus, complex applications can be built simply by combining small reusable blocks. A block is coded in programming language C++ and implemented as an inherited class of the fundamental block. It is compiled as a shared object on Linux. Since data communication is done by using a pointer, it is much faster than socket-communication-based middlewares such as OpenRTM [8]. Therefore, FlowDesigner maintains a well-balanced tradeoff between independence and processing speed. As mentioned above, because a large amount of data is communicated in our system, FlowDesigner is suitable for our system. Concretely, SSS in Fig. 3 has the heaviest traffic, and a large bandwidth of 12.8 Mbps for input and 8 Mbps for output are necessary. The following sections describe each module in detail.

TABLE I
Parameters for system to recognize simultaneous speech signals

Genes	Hand-	Alleles			
(Parameters in recognition system)	tuned	Min	Max	Step	#Elem
Leak-estimate factor	0.25	0.05	0.5	0.05	10
Canceling rate for leak-estimate factor	1	0.2	1.8	0.2	9
Compensation for background noise	1.2	0.1	1.5	0.1	15
Coefficient for smoothed spectrum	0.5	0.1	0.8	0.1	8
Weight for reducing background noise	0	0	1	0.1	11
Coefficient for estimating a priori SNR	0.8	0.1	1	0.1	10
Amplification rate for leak-estimate-fact	1.5	1	2	0.1	11
Coefficient for estimating BG-noise	0.98	0.8	1	0.02	11
Reverberation decay	0.5	0	0.5	0.05	11
Instantaneous reverb. attenuation level	0.2	0	0.5	0.05	11
AMG threshold	0.25	0	0.65	0.05	13

B. Parameter Optimization by Genetic Algorithm

To improve recognition performance, we modified our robot audition system so that suitable parameters could be selected depending on circumstances. We optimized the parameters for combinations of speakers' locations. Our system has eleven parameters as shown in Table I. We made a gene correspond to a parameter, and made alleles correspond to parameter values. These parameters are dependent mutually, and thus it is difficult to optimize them manually. To solve this problem, we applied GA to parameter optimization.

We prepared three speech datasets for the parameter optimization. Dataset 1 contained mixtures of two simultaneous speech signals. The mixtures were composed by convolving speech signals of 216 Advanced Telecommunications Research Institute International (ATR) phonemically-balanced words and measured impulse responses. The impulse responses were measured at twelve positions in a room, where a robot was located at the center. The distance between the robot and a sound source was selected from one of 100, 150, and 200 cm. The azimuth of the sound source in the robot's coordinates was selected from 0°, 30°, 60°, and 90°. The height was 136 cm. In our configuration to generate two simultaneous speech signals, one speaker was located in front of a robot, that is, 0°, and the other speaker was located in other directions at the same distance. As a result, two simultaneous speech signals for the nine combinations of positions were composed. Dataset 2 contained clean speech data of continuous speech corpus called the Acoustical Society of Japan Japanese Newspaper Article Sentences (ASJ-JNAS). It includes 306 utterance sets (153 male and female for each). Each utterance set consists of 150 sentences excerpted from ASJ-JNAS. So, Dataset 2 contains about 45,000 sentences in total. To create Dataset 3, we first composed speech signals by convolving 216 ATR phonemically-balanced words and impulse responses which were measured at 100, 150, and

200 cm from 0°. Dataset 3 is, then, generated as a set of separated speech obtained by extracting a sound source in front of the robot (0°) from the composed speech signals.

As an acoustic model, the system used a triphone model consisting of HMM with 3 states and 4 mixtures. The triphone model is obtained as follows: first, it was trained on Dataset 2, and then it was adapted to Dataset 3 by using *Maximum Likelihood Linear Regression (MLLR)*. A grammar language model was used to recognize isolated words. The size of vocabulary in a word dictionary for ASR was 200 words.

The procedure of parameter optimization with GA is as follows:

1) Initialization
An initial population which includes N individuals is generated. All genes of each individual are decided at random.
2) Crossover
This process makes children from parents M pairs of individuals are selected at random, and the pairs make $2M$ children. M is called the number of crossovers. In our GA, children are generated by using a uniform crossover method. Since generated children are added to population, the size of the population becomes $N + 2M$.
3) Mutation
Each individual in the population mutates with a mutation rate p_m. In our GA, each gene is replaced with an allele selected at random.
4) Calculation of fitness
Fitness is defined as a word correct rate of our system, since our goal is to optimize parameters to improve a word correct rate. The word correct rates of our system are calculated by separating and recognizing Dataset 1.
5) Selection
The constant size of population N should be maintained. As a method of selection, we adopted a combination of *elite selection* and *roulette selection*. The procedure of selection is as follows:
    a) Elite selection
    L individuals which have the L-best fitness in the population ($N + 2M$ individuals) are selected.
    b) Roulette selection
    $N - L$ individuals are selected from the remaining population. An individual which has high fitness is selected with a high probability.
6) Evaluation of the population
The system compares the fitness average of all individuals in a current generation with that in the previous generation. When the difference is less than T_{GA}, the population is considered to have converged.

In optimizing the parameters, let population size $N = 100$, number of crossover $M = 40$, number of elite selection $L = 10$, and mutation rate $p_m = 0.01$. We optimized the parameters for a central speaker in three combinations of speaker directions. Two loudspeakers were located in the center (0°) and the left (30°, 60°, or 90°). In each combination of speaker directions, the loudspeakers were located at a distance of 100 cm, 150 cm, or 200 cm from the robot. Transitions of fitness are shown in Fig. 1.

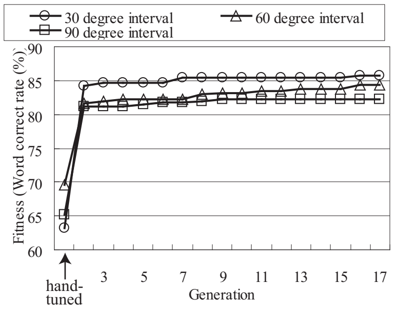

Fig. 1. Transitions of fitness in three combinations of speaker directions

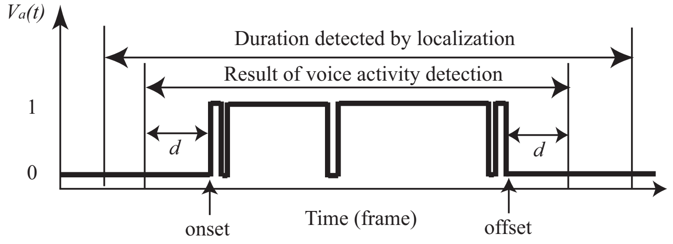
Va(t)
Duration detected by localization
Result of voice activity detection
1
0
d
d
onset
Time (frame)
offset

Fig. 2. An example of detected voice activity duration

C. Voice activity detection

VAD relying only on power is commonly used, however, it cannot detect speech with a low SNR such as simultaneous speech correctly. We propose a new VAD method based on a power spectrum and location of a separated sound. We assume that voice is active while localization results are provided. The detected voice activity duration includes unnecessary noises and silent durations, which deteriorate the performance of ASR. To solve the problem, we applied the proposed VAD method to speech separated by sound source separation.

The algorithm for our proposed VAD is as follows: first, voice activity is estimated per time frame. Let the spectrum of the separated sound source m at time frame t be $\hat{s}_m(k,t)$. $FBC(t)$ is defined as the number of frequency bands which satisfy $\hat{s}_m(k,t) \geq T_{sil}$ at time frame t. Since a speech signal has harmonics, $FBC(t)$ tends to be large. If $FBC(t) > T_{FBC}$, voice activity $V_a(t)$ at time frame t is estimated as 1 (active), otherwise 0 (silent). Voice activity duration is then detected by considering a temporal sequence of the estimated $V_a(t)$. Fig. 2 illustrates an example of detected voice activity duration, where a horizontal axis is time and a vertical axis is $V_a(t)$. When $V_a(t)$ are 0 for more than d time frames continuously, an onset or an offset of voice activity duration is detected. As a result, noises and useless silent durations are removed.

III. Implementation of Real-Time Recognition System

In this section, we will explain the implementation of the robot audition system that recognizes simultaneous speech signals. Fig. 3 shows the architecture of the system. Our

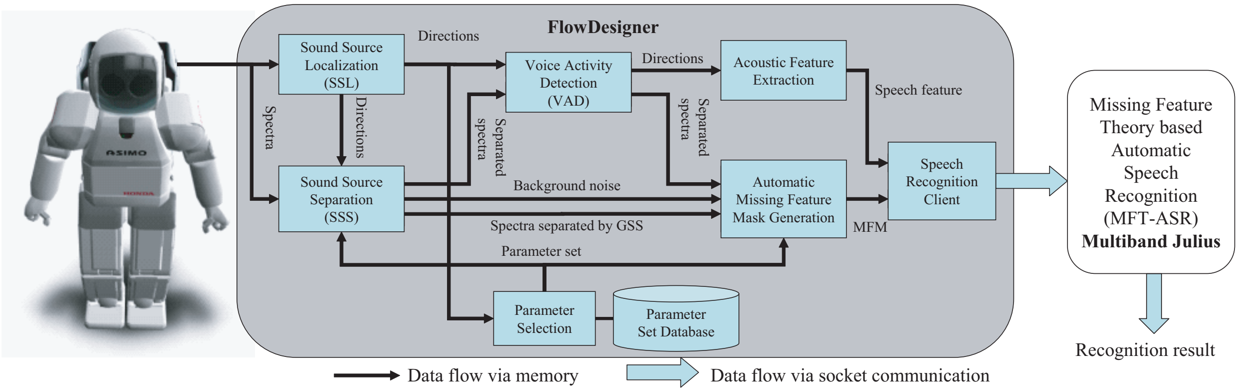
FlowDesigner
Sound Source Localization (SSL)
Directions
Voice Activity Detection (VAD)
Directions
Acoustic Feature Extraction
Speech feature
Spectra
Directions
Separated spectra
Separated spectra
Sound Source Separation (SSS)
Background noise
Spectra separated by GSS
Automatic Missing Feature Mask Generation
MFM
Speech Recognition Client
Parameter set
Parameter Selection
Parameter Set Database
Missing Feature Theory based Automatic Speech Recognition (MFT-ASR) Multiband Julius
Recognition result
Data flow via memory
Data flow via socket communication
ASIMO

Fig. 3. Real-time robot audition system that recognizes simultaneous speech

system consists of seven modules: Sound Source Localization (SSL), Sound Source Separation (SSS), Parameter Selection, Voice Activity Detection (VAD), Acoustic Feature Extraction, Automatic Missing Feature Mask Generation, and Missing Feature Theory based Automatic Speech Recognition (MFT-ASR). The six modules except for MFT-ASR are implemented as blocks for FlowDesigner described in Section II. For MFT-ASR, we use Multiband Julius. A CPU occupancy of MFT-ASR is high in recognizing speech. Since data communication between MFT-ASR and the other modules has acoustic features and MFM rather than raw signal data, data traffic communication is light. Therefore, we did not implement MFT-ASR as a module for FlowDesigner. FlowDesigner and Multiband Julius can run separately on different CPUs since they communicate with each other via a network.

Our robot audition system run on a personal computer with Pentium 4/2.4 GHz. The OS is Linux with kernel 2.4. The blocks for FlowDesigner are compiled with optimization option (-O3 -march=pentium4 -fomit-frame-pointer -msse -funroll-loops -ffast-math) by g++ (GCC) 3.2.2.

A. Sound Source Localization

We adopted steered beamformer [9] as a localization algorithm of our system. The basic idea behind the steered beamformer approach to source localization is to direct a beamformer in all possible directions and look for maximal output. For this task, we tried to maximize the output power of a simple delay-and-sum beamformer. It was reported that the SSL system was able to reliably detect speech sources within 5 meters, and the root mean square error of localization results was about 1.4°. The SSL block successfully tracked two sources when their angle interval was more than 20°.

B. Sound Source Separation

The SSS [10] consists of Geometric Source Separation (GSS) and a multi-channel post-filter. We modified the GSS proposed by Parra *et al.* [11] so as to provide faster adaptation using stochastic gradient and shorter time frame estimation. Initial separation using GSS was followed by the multi-channel post-filter that is based on a generalization of beamformer post-filtering [10], [12] for multiple sources. This post-filter used adaptive spectral estimation of background noises and interfering sources to enhance the signal produced. The main idea resides in the fact that, for each source of interest, the noise estimate is decomposed into stationary and transient components assumed to be due to a leakage between the output channels of the initial separation stage. It was reported that the SSS improved 10.3 dB in SNR on average for separation of three simultaneous speech signals.

C. Voice Activity Detection

VAD extracts the speech duration required for ASR by using spectra of speech signals separated by SSS. The method of VAD is detailed in Section II-C.

D. Acoustic Feature Extraction

This block calculates acoustic features for MFT-ASR from the spectrum of separated speech. Mel-Frequency Cepstrum Coefficient (MFCC) is a common acoustic feature for ASR. However, MFCC is not appropriate for recognition of speech separated in frequency domain, since noise in each frequency band spreads to all coefficients in cepstral domain. We used the Mel Scale Log Spectrum (MSLS) obtained by applying Inverse Discrete Cosine Transformation to MFCCs. The calculation of MSLS is described in [13]. The acoustic feature vector composes 48 spectral-related acoustic features consisting of 24 spectral features and 24 differential features.

E. Automatic Missing Feature Mask Generation

Since a multi-channel post-filter provides information about the amount of noise included in a frequency band, automatic MFM generation uses the information to estimate MFM that indicates reliability of each spectral feature. Since we use an acoustic feature vector of 48 spectral-related acoustic features, the missing feature mask is a vector of 48 corresponding features. Each element of a vector represents the reliability of each acoustic feature. The value may be binary (1, reliable, or 0, unreliable) or continuous between 0 and 1. In this paper, we used a binary missing feature mask. The detailed method is presented in [13].

F. Missing Feature Theory Based Automatic Speech Recognition

Missing Feature Theory Based Speech Recognition (MFT-ASR) [14] outputs a sequence of phonemes from acoustic features of separated speech and the corresponding MFMs. MFT-ASR is an HMM based recognizer, which is commonly used in conventional ASR systems. The difference is only in their decoding processes. In conventional ASR systems, estimation of a path with maximum likelihood is based on state transition probabilities and output probability in HMM. This estimation process of output probability is modified in MFT-ASR as follows: let $M(i)$ be a MFM vector which represents the reliability of the i-th acoustic feature. The output probability $b_j(x)$ is given by

$$b_j(x) = \sum_{l=1}^{L} P(l|S_j) \exp \left\{ \sum_{i=1}^{N} M(i) \log f(x(i)|l, S_j) \right\}, \tag{1}$$

where $P(\cdot)$ is a probability operator, $x(i)$ is an acoustic feature vector, N is the size of the acoustic feature vector, S_j is the j-th state.

For MFT-ASR, we used Multiband Julius [15], which is based on the Japanese real-time large vocabulary speech recognition engine Julius [16]. It supports various types of HMMs such as shared-state triphones and tied-mixture models. Stochastic language models are also supported. In decoding, an ordered word bi-gram is used in the first pass, and a reverse ordered word tri-gram in the second pass. It works as a standalone or client-server application. To run as a server, we modified the system to be able to communicate acoustic features and MFM via a network.

G. Parameter Selection

Parameter selection module selects an appropriate parameter set for a current state by using results of SSL. There are eleven parameters in our system described in Section II-B, which concern the performance of SSS and ASR. We optimized these parameter values for two simultaneous speech.

We prepared parameter set database, $P(\boldsymbol{\theta}(i))$ which denotes a set of eleven parameters optimized for a combination of sound source locations $\boldsymbol{\theta}(i) = (\theta_1(i), \theta_2(i), \cdots, \theta_M(i))$. M is the number of sound sources, that is, two in our experiments. When the results of SSL at t-th time frame are $\boldsymbol{\phi} = (\phi_1, \phi_2, \cdots, \phi_M)$, where ϕ_m is an azimuth of sound source m in the microphone array's coordinates, a parameter set $P(\boldsymbol{\theta}(i))$ for t-th frame is selected to satisfy the following conditions:

$$\forall m \; |\phi_m - \theta_m(i)| < \theta_\delta \tag{2}$$

where θ_δ is a threshold to assign ϕ_m to $\theta_m(i)$. If there exists no parameter sets that satisfy the above conditions, hand-tuned parameter set is selected.

IV. Evaluation

To evaluate our robot audition system by recognition performance and processing speed, we performed experiments where the robot with a microphone array separated two simultaneous speech signals, and recognized speech of a speaker in front of it. As for recognition performance, we compared the recognition results when optimized parameters were used or not, and when VAD was used or not. As for processing speed, we measured processing time, processing delay, and CPU occupancy of our robot audition system.

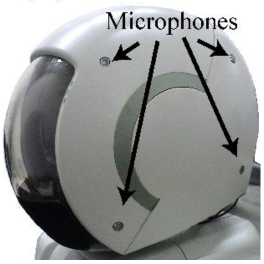

Fig. 4. ASIMO with eight microphones

TABLE II
POSITIONS OF TWO SPEAKERS IN THE EXPERIMENTS

	Distance	speaker A (front)	speaker B (left)
Position 1	100 cm	0°	30°
Position 2	200 cm	0°	30°
Position 3	100 cm	0°	60°
Position 4	200 cm	0°	60°
Position 5	100 cm	0°	90°
Position 6	200 cm	0°	90°

TABLE III
CONDITIONS OF THE EXPERIMENTS

	Localization	Voice activity duration	Parameter optimization
Condition 1	given	given	hand-tuned
Condition 2	given	given	GA-optimized
Condition 3	estimated	not used	hand-tuned
Condition 4	estimated	not used	GA-optimized
Condition 5	estimated	estimated	hand-tuned
Condition 6	estimated	estimated	GA-optimized

A. Conditions

We used Honda ASIMO as a testbed, and eight microphones were installed in the head of ASIMO (Fig. 4). The positions of the microphones are bilaterally-symmetric. The robot was located at the center of a room which was $7\,m \times 4\,m$. Three walls were covered with sound absorbing materials, while the other wall was made of glass which makes strong echoes. A reverberation time (RT20) of the room is about 0.2 seconds. However, the reverberation is not uniform in the room because of an asymmetrical echo generated by the glass wall. Two simultaneous speech signals were recorded in the room. We used two loudspeakers to create simultaneous speech. Table II shows six positions of the loudspeakers in the experiments. The system had the same acoustic and grammar language models as those used in parameter optimization. Table III shows six conditions of the experiments. "given" in Table III means that the corresponding parameters such as localization result and voice active duration, were given by hand. "estimated" means that they were estimated by the system automatically.

B. Results

Our robot audition system separated two simultaneous speech signals and recognized speech from speaker A. Fig. 5

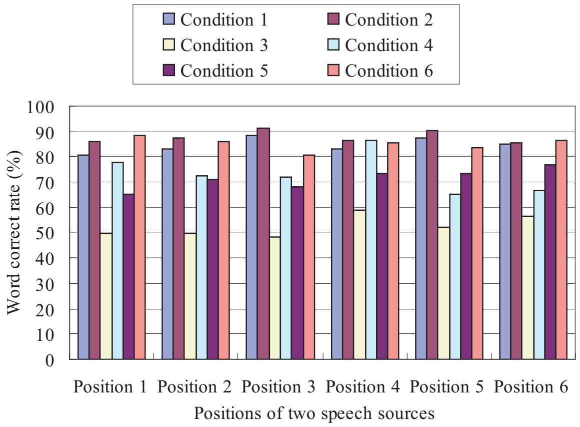

Fig. 5. Word correct rates of speech recognition in center direction

shows the results of recognition in word correct rate. The results by using GA-optimized parameters outperformed those by using the hand-tuned parameters in every case. The results showed that parameter optimization using GA was successful.

When the results with/without VAD were compared, VAD improved word correct rates in most cases. When the recognition performance was low, VAD worked more effectively. This proves that reduction of voice activity detection errors improved the system performance drastically.

As for processing speed, we measured processing time when our robot audition system separated and recognized speech signals of 800 seconds. It took an average of 499 seconds for our robot audition system to recognize the speech signal. FlowDesigner consumed 369 seconds on average, and Multiband Julius consumed the remaining time. A CPU occupancy of FlowDesigner was constantly about 75% in running alone, while it ranged from 50% to 80% in Multiband Julius running. On the other hand, a CPU occupancy of Multiband Julius ranged from 30% to 40% in recognizing speech, otherwise it was 0%. A delay between a beginning of localization and a beginning of recognition was about 0.40 seconds, and a delay between an ending of separation and an ending of recognition was about 0.046 seconds. As a whole, our robot audition system ran fast as in real time. When our system separated two sound sources and recognized one separated sound, one CPU was sufficient. However, when our system separates more sound sources and recognizes more separated sounds, it will require more CPUs.

V. Conclusion

We reported a microphone array embedded robot audition system that recognizes simultaneous speech in the real world. To attain online processing, we introduced FlowDesigner based architecture for module integration. To optimize parameters, we developed GA based parameter optimization. To improve VAD, we integrated a new VAD method based on the power and location of sound sources. We performed the experiments to evaluate these approaches through the recognition of two simultaneous speech. As a result, we confirmed fast online processing, effectiveness of parameter optimization, and efficiency of VAD.

A future work is the adaptation to changing environments around a robot. Our system was optimized for the case when a robot was located under a specific environment. Parameter optimization using GA requires a high computational cost. We should develop a faster method of parameter optimization to adapt to a new and a dynamically changing environment. Another future work is recognition of environmental sounds except for speech to understand more general sounds.

Acknowledgment

Our research is partially supported by Grant-in-Aid for Scientific Research and 21 COE Program. We thank Jean Rouat and François Michaud, Université of Sherbrooke for their help. We thank the members of Honda Research Institute Japan, Co., Ltd. and the members of Okuno laboratory, Kyoto University.

References

[1] I. Hara, F. Asano, H. Asoh, J. Ogata, N. Ichimura, Y. Kawai, F. Kanehiro, H. Hirukawa, and K. Yamamoto, "Robust speech interface based on audio and video information fusion for humanoid HRP-2," in *Proceedings of IROS 2004*. pp. 2404–2410.

[2] K. Nakadai, H. G. Okuno, and H. Kitano, "Robot recognizes three simultaneous speech by active audition," in *Proceedings of ICRA-2003*. pp. 398–403.

[3] S. Yamamoto, K. Nakadai, H. Tsujino, T. Yokoyama, and H. G. Okuno, "Assessment of general applicability of robot audition system by recognizing three simultaneous speeches," in *Proceedings of IROS 2004*. pp. 2111–2116.

[4] J. Barker, M. Cooke, and P. Green, "Robust ASR based on clean speech models: An evaluation of missing data techniques for connected digit recognition in noise," in *Proc. of Eurospeech-2001*. pp. 213–216.

[5] M. Cooke, P. Green, L. Josifovski, and A. Vizinho, "Robust automatic speech recognition with missing and unreliable acoustic data," *Speech Communication*, vol. 34, pp. 267–285, May 2001.

[6] D. E. Goldberg, *Genetic Algorithm in Search, Optimization, and Machine Learning*, I. Addison-Wesley Publishing Company, Ed. Addison-Wesley Publishing Company, Inc., 1989.

[7] C. Côté, D. Létourneau, F. Michaud, J.-M. Valin, Y. Brosseau, C. Raievsky, M. Lemay, and V. Tran, in *Proceedings of IROS 2004*. pp. 1820–1825.

[8] N. Ando, T. Suehiro, K. Kitagaki, T. Kotoku, and W.-K. Yoon, "Rt-middleware: Distributed component middleware for rt (robot technology)," in *Proceedings of IROS 2005*. , pp. 3555–3560.

[9] J.-M. Valin, F. Michaud, B. Hadjou, and J. Rouat, "Localization of simultaneous moving sound sources for mobile robot using a frequency-domain steered beamformer approach," in *Proceedings of ICRA 2004*. pp. 1033–1038.

[10] J.-M. Valin, J. Rouat, and F. Michaud, "Enhanced robot audition based on microphone array source separation with post-filter," in *Proceedings of IROS 2004*. pp. 2123–2128.

[11] L. C. Parra and C. V. Alvino, "Geometric source separation: Mergin convolutive source separation with geometric beamforming," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 6, pp. 352–362, 2002.

[12] I. Cohen and B. Berdugo, "Microphone array post-filtering for non-stationary noise suppression," in *ICASSP-2002*, 2002, pp. 901–904.

[13] S. Yamamoto, J.-M. Valin, K. Nakadai, T. Ogata, and H. G. Okuno, "Enhanced robot speech recognition based on microphone array source separation and missing feature theory," in *Proceedings of ICRA 2005*. pp. 1489–1494.

[14] B. Raj and R. M. Stern, "Missing-feature approaches in speech recognition," *Signal Processing Magazine*, vol. 22, no. 5, pp. 101–116, 2005.

[15] Multiband Julius, "http://www.furui.cs.titech.ac.jp/mband_julius/."

[16] T. Kawahara and A. Lee, "Free software toolkit for japanese large vocabulary continuous speech recognition," in *International Conference on Spoken Language Processing (ICSLP)*, vol. 4, 2000, pp. 476–479.