Implementation of Active Direction-Pass Filter on Dynamically Reconfigurable Processor

Shunsuke Kurotaki[†], Noriaki Suzuki[†], Kazuhiro Nakadai[‡], Hiroshi G. Okuno[*], Hideharu Amano[†]

[†]Graduate School of Science and Technology, Keio University, Yokohama, Japan
[‡]HONDA Research Institute Japan Co.,Ltd. Saitama, Japan
[∗] Graduate School of Informatics, Kyoto University, Kyoto, Japan

drp@am.ics.keio.ac.jp, nakadai@jp.honda-ri.com, okuno@kuis.kyoto-u.ac.jp

Abstract— **In this paper, we report the design and implementation of a sound source separation system using a dynamically reconfigurable device. A robot in real-world environments should have an ability to treat a mixture of multiple sound signals. Active Direction-Pass Filter (ADPF) which extracts sound from a specific direction by using a pair of microphones has been developed as such a method of sound source separation. The ADPF was used as a front-end for an automatic speech recognition system, and recognition of three simultaneous speech signals has been reported. The ADPF, however, requires a lot of computational power, while the battery capacity and the physical size of the robot are limited. To reduce the power consumption and the size of the system, we adopted the dynamically reconfigurable device, "DRP" developed by NEC Electronics. We implemented the ADPF on DRP, and investigated the effectiveness of dynamically reconfigurable device for these applications. The preliminary experiment shows that ADPF on DRP separates a mixture of sound sources in real-time with practical accuracy.**

Index Terms— **Robot Audition, Sound Source Separation, Active Direction-Pass Filter (ADPF), Dynamically Reconfigurable Device, DRP**

I. INTRODUCTION

An auditory function is essential for a humanoid for natural human-robot communication. Although a lot of works for establishment of artificial auditory functions have been exerted, most of them are hard to be applied in the practical communication situation of robots.

Since general sound that robots hear in daily environments consists of not a single sound source but multiple sounds including voices, music, acoustic signals and other noises, a robot audition in a real world environment should have an ability to treat a mixture of multiple sound signals as people do. People with normal hearing can separate particular sounds from a mixture of sounds, and focus on a specific voice or sound even in a noisy environment. This capability is known as the *cocktail party effect*, so robots in the real world should also have a comparable ability. A function of sound source separation is necessary for the front-end processing for speech recognition of simultaneous speech signals.

A traditional model for computational auditory processing in binaural research is often based on calculating *Interaural Phase Difference* (IPD) and *Interaural Intensity Difference* (IID) obtained by a *head-related transfer function* (HRTF). The HRTF, however, depends on the shape of a head and also room acoustic so that the measurement has to be repeated whenever the system is installed at a different room. Considering the real world, acoustic features of the environment are usually not known in advance. It is inadequate for any practical system that such an extensive measurement is required for every operating space. Therefore, these approaches based on the HRTF do not suit for a mobile system, and an alternate method is required.

Active Direction-Pass Filter (ADPF) [4] has been proposed to overcome these obstacles. It can extract the sound of a specific direction by using a pair of microphones. Two significant features attain accurate and fast sound source separation under a real world environment. One is that the IPD and IID are estimated without using the HRTF, and the other is that it controls its own parameters actively to improve the accuracy of separation.

The ADPF can be applied to an audition system equipped with a humanoid, and the recognition of three simultaneous speech signals has been reported [3]. The ADPF, however, requires a large amount of computational power for mobile robot on which battery capacity and the physical size are strictly limited. In terms of reduction of power consumption and the heavy load on the CPU, using an application-specific hardware is one of the effective solutions to these difficulties. By taking flexible modification of algorithms or parameters into account, reconfigurable devices are likely to be ideal. There is an FPGA (Field Programmable Gate Array) that is an typical reconfigurable device based on LUT (Look Up Table). So far, we implemented partial processing modules of the ADPF by using a commercial FPGA. Here, we propose to utilize a dynamically reconfigurable device which is expected to be a more efficient solution from the viewpoint of the size and power consumption.

This paper presents the prototype implementation of the ADPF on the dynamically reconfigurable device, ”DRP

0-7803-8912-3/05/$20.00 ©2005 IEEE

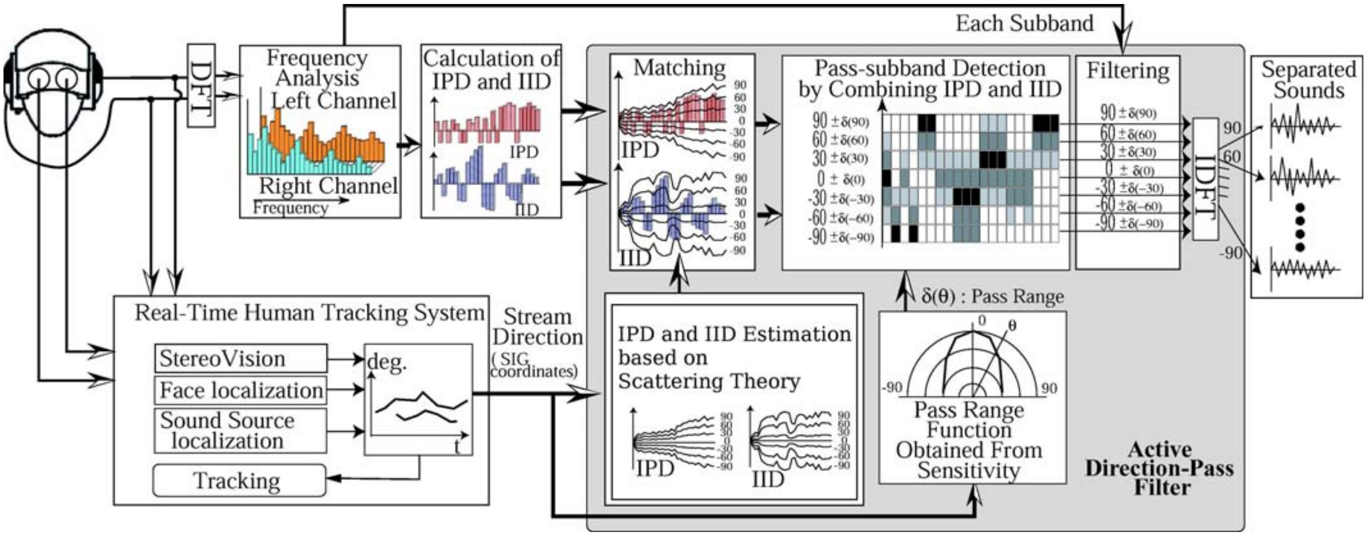


Fig. 1. Active Direction-Pass Filter

(Dynamically Reconfigurable Processor)” [6] developed by NEC Electronics, and explores an effectiveness of DRP for perceptual processing on mobile robots. As a result of experiments, the prototype system achieved real-time sound source separation with acceptable accuracy.

The rest of this paper is organized as follows: Section II explains an algorithm of the ADPF. Section III introduces what dynamically reconfigurable device is and describes an architecture of DRP. In Section IV, the design and implementation of sound source separation system on DRP are presented. The results of performance evaluation are shown in Section V, and this work is concluded in Section IV.

II. Active Direction Pass Filter

In this section, the binaural sound source separation method by the ADPF is described. The ADPF can extract sound signals of specific direction by collecting subbands with the use of IPD and IID. IPD and IID are modelled by *Scattering Theory*[2] instead of using HRTF which is often used in binaural research. That dispenses with any prior acoustic measurement to obtain IPD and IID, so this method is more suitable for applying to robots in a real world environment.

Then, it is generally known that an accuracy of the binaural sound source separation depends on a sound source direction. Its sensitivity is higher for the front direction while it becomes lower as deviating from front. The ADPF takes that into account, then a directional “pass range” of filtering is actively controlled according to the direction.

Pass range designates the range which sounds within should be extracted. On a single sound source, the wider pass range realizes the higher signal-to-noise ratio of sound extraction. However, background noise and other sound sources should be considered in real environment, so a narrower pass range is the better in a sense of noise cancellation.

The pass range function, which is used to decide the pass range, is defined as Figure 2. It shows the pass range is narrow angles in front direction, and wider angles in the periphery. For example, the sound source exists at the front of the robot (direction of 0°), the pass range of ±10° is necessary to extract the sound properly. In case of the 90° sound from the front, at least, the pass range of ±35° is necessary.

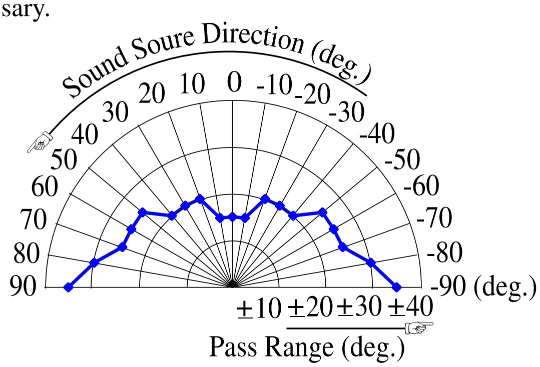


Fig. 2. Pass Range Function

The architecture of the ADPF is shown in the dark area in Figure 1. The algorithm of the ADPF consists of six steps as follows:

1) The actual IPD $\Delta\varphi'$ and IID $\Delta\rho'$ in each subband are calculated by the difference between left and right channels.

$$\Delta\varphi' = \arctan\left(\frac{\Im[Sp_l(f)]}{\Re[Sp_l(f)]}\right) - \arctan\left(\frac{\Im[Sp_r(f)]}{\Re[Sp_r(f)]}\right)$$

$$\Delta\rho' = 20\log_{10}\left(\frac{|Sp_l(f)|}{|Sp_r(f)|}\right)$$

where, Sp_l, and Sp_r are the spectrum of frequency f obtained by FFT from the left and right signals. $\Re[S]$, $\Im[S]$ indicate real part and imaginary part of spectrum S.

2) Let θ_s azimuth be a direction that should be extracted.
3) The pass range $\delta(\theta_s)$ of the ADPF is selected according to θ_s. The pass range function δ is discribed above. Let $\theta_l = \theta_s - \delta(\theta_s)$, and $\theta_h = \theta_s + \delta(\theta_s)$. It means the direction between θ_l and θ_h should be extracted.
4) The presumed IPD $\Delta\varphi(\theta)$ is estimated for each sub-band by Scattering Theory. Likewise, the presumed IID $\Delta\rho(\theta)$ is obtained.
5) The sub-bands are collected if the IPD and IID satisfy the following conditions.

$$\begin{aligned} f < f_{th}: &\quad \Delta\varphi(\theta_l) \leq \Delta\varphi' \leq \Delta\varphi(\theta_h), \\ f \geq f_{th}: &\quad \Delta\rho(\theta_l) \leq \Delta\rho' \leq \Delta\rho(\theta_h) \end{aligned}$$

   where, the f_{th} is the upper boundary of frequency which is efficient for IPD. It is used as the threshold to decide which IPD or IID are applied for condition. Its value depends on the baseline of the microphones.
6) A wave consisting of collected subbands is constructed by IDFT.

III. Dynamically Reconfigurable Processor

A. The Concept of Dynamically Reconfigurable Device

An application specific hardware for off-loading heavy weight processes of CPU is widely used in recent SoC(System on-a Chip) or ASIC (Application Specific IC) to achieve both high performance and low power consumption. However, such a specialized chip cannot be used by another application and new standards or techniques are hard to be built in after shipment.

A chip combining a CPU and a reconfigurable device is an attractive solution especially in application field of robots in which high performance with low power consumption is required for various kind of jobs, and technical innovations are rapidly introduced. Since the configuration of a reconfigurable device is flexible, the same chip can be used for various applications. It can also be "refitted" after shipment by rewriting the configuration data, and the development cost and period can be drastically reduced compared with an ASIC (show Table I).

Though there is an FPGA as a commercially available reconfigurable device, these fine grain reconfigurable architectures using LUTs are not always efficient in a performance and an area. Recent coarse grain dynamically reconfigurable devices[1][6][7] have been developed to achieve high performance and flexibility for a fraction of the cost of an FPGA. A dynamically reconfigurable device consists of a coarse grain cell including ALU, data manipulator, and register files as the primitive processing element. These architectures are prospective for high performance especially for a stream application such as an auditory processing.

Furthermore, a dynamically reconfigurable device is believed to be more suitable for mobile terminals whose hardware resources and battery capacity are strictly limited because of "Time-multiplexed Execution" introduced with multi-context functionality. Multiple configurations of circuits called "Context" are stored in internal memory, and it can be rapidly (often in one clock) interchanged to implement different tasks. This multi-context functionality gives reducing the physical die size and improving power efficiency. Because a whole task is divided into multiple contexts, and only one context which is required at that point should be mapped and executed.

TABLE I

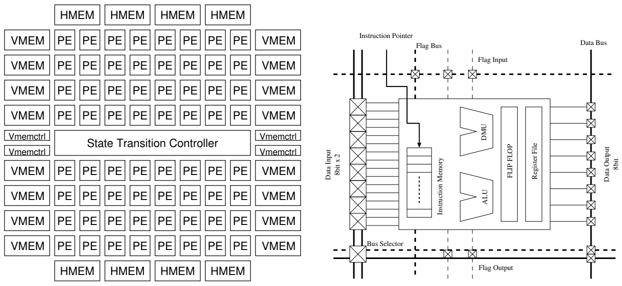
Comparison of Core Device Features

	CPU	ASIC	Reconfigurable Device
Computational Power	Inadequate	Highest	Adequate
Flexibility	Highest	Poor	High
Development Cost	Lowest	High	Low
Power Consumption	High	Lowest	Low

B. The Architecture of "DRP"

In this paper, we focused on Dynamically Reconfigurable Processor (DRP) developed by NEC/NEC electronics. DRP is a coarse-grain dynamically reconfigurable processor core which can be integrated into an SoC with a simple CPU core. The primitive unit of DRP core is called a "Tile", and a DRP core consists of arbitrary number of Tiles. The number of Tiles can be expandable horizontally and vertically.

The primitive modules of a Tile are processing elements (PEs), a state transition controller (STC), 2-ported memories (Vertical Memories or VMEMs), its controller (VMEMCtrl), and 1-ported memories (Horizontal Memories or HMEMs). The structure of a Tile is shown in Figure 3(a).



(a)

(b)

Fig. 3. The Primitive Modules of DRP

There are 64 PEs located in one Tile. The architecture of a PE is shown in Figure 3(b). It has an 8-bit ALU, an 8-bit data manipulation unit (DMU; for shifts/masks), sixteen 8-bit register file units (RFUs), and an 8-bit flip-flop unit (FFU). These units are connected by programmable wires specified by instruction data. Each PE has 16-depth instruction memories and supports multiple context operation. An instruction pointer is delivered from the STC and indicates an instruction data of instruction memories.

The STC is a programmable sequencer in which finite state machine (FSM) can be stored. The STC has 64 states, and each state is associated with an instruction pointer. FSM of the STC operates synchronized with the internal clock, and generates the instruction pointer for each clock cycle according to the state. Also, the STC can receive event signals from PEs to branch conditionally. The maximum number of branch is four.

As for the memory units, a Tile has sixteen 2-ported VMEMs on its right and left sides, and eight 1-ported HMEMs on its upper and lower boundary. Each VMEM is 8-bit in width and 256-word in depth. 8 of them are provided for each Tile. The HMEM is a single-ported 8-bit memory with 8K entries. Contents of these memories, flip-flops, and register files of PEs are all connected and shared by the contexts.

The DRP core, consisting of several Tiles, can change its contexts every cycle with the instruction pointer distributed from the central STC (CSTC). The individual STCs within the Tiles can also be ran independently by programming different FSMs.

C. The Prototype Chip DRP-1 and its Design Environment

As shown in Figure 4, the prototype chip DRP-1 is consisting of 8-Tile DRP core, eight 32-bit multipliers, an external SRAM controller, a PCI interface, and 256-bit I/Os. It is fabricated with 0.15-μm 8-metal layer CMOS process, and the maximum operation frequency is 100-MHz. Although DRP-1 can be used as a stand-alone reconfigurable device, Tiles of DRP can be used as an intellectual property (IP) on ASICs with an embedded processor. In this case, the number of Tiles can be chosen so as to achieve the required performance with minimum area.

The integrated design environment for DRP-1 which includes a high level synthesis tool, a design mapper for DRP, simulators, and a layout/viewer tool is provided. Applications can be written in a C-based high level hardware description language, synthesized, and mapped directly onto the DRP-1.

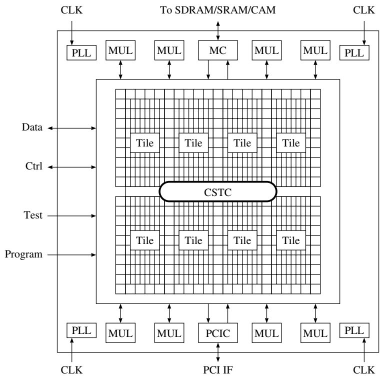


Fig. 4. The DRP-1 Architecture and its Core

IV. Implementation

A. The Specification of the Evaluation Board

We have implemented a prototype of the ADPF by using a DRP evaluation board shown in Figure 5. As shown in Figure 5(b), it is composed of DRP-1 core described in the previous section, a PCI Interface, an FPGA which is used for I/O control, and external SRAM. This evaluation board is connected with a host PC through the Host-PCI bus whose transfer rate is 64/32bit at 33MHz. The data between the host PC and DRP are transferred via the input/output FIFO on the FPGA by each 64bit. The host program is allowed to control this board and its data transfer by using provided API.

Note that this board is just a prototype, and a simpler board which equips a embedded CPU and DRP core will be used when the proposed system is implemented into a real robot.

B. The Detail of Implementation

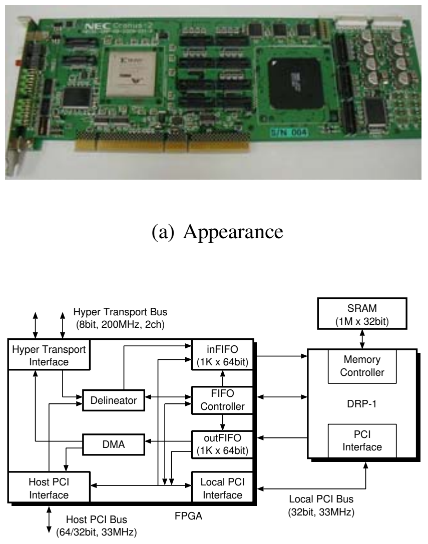
In this work, the ADPF has been designed on assumption of that sampling rate is 16kHz, frame length is 2048 points, and window shift is 512 points. 16bit fixed-point is used as data format for internal computations.

The diagram of the implemented ADPF is presented as Figure 6. Input signals are written from the host CPU into internal memory on DRP by each 512-sample. It has an extra buffer to preserve the previous inputs, since the results of an internal operation designed for handling 2048-samples are overwritten into the same memory. After the operation, central 512 data of 2048 is read out to the host CPU as an extracted stream, because the date of both ends include noise arising from linkage error.

In this implementation, only the IPD without IID is taken into account for subband selection. But IPD is not effective enough for high frequency domain, so subband filtering is operated for only low frequency domain. Actually, subbands of high frequency domain (in this work, the threshold is 1500Hz) are not all collected. Because it is reported that the subband with frequencies of more than 1500kHz does not have much effect on separation[4].

As shown in Figure 6, the whole operation of sound source separation is roughly divided to FFT, ADPF, IFFT. Additionally, the ADPF consists of “Calculation of IPD” and “Matching IPDs”. The summaries of each module implementation are describes below.

- *FFT/IFFT* : The simplest FFT/IFFT algorithm has been implemented as a subsidiary module of the ADPF[8]. Same contexts are shared by both FFT and IFFT so as to save occupied resources. The function of FFT/IFFT can be switched by setting the value of an internal register.
- *Calculation of actual IPD* : On the occasion of calculating IPD, ArcTan function is needed. However, the precise ArcTan arithmetic circuit occupies a large amount of hardware resources. So the pseudo function has been implemented. An index of 10-bit is generated



(a) Appearance

(b) Block Diagram

Fig. 5. DRP Evaluation Board

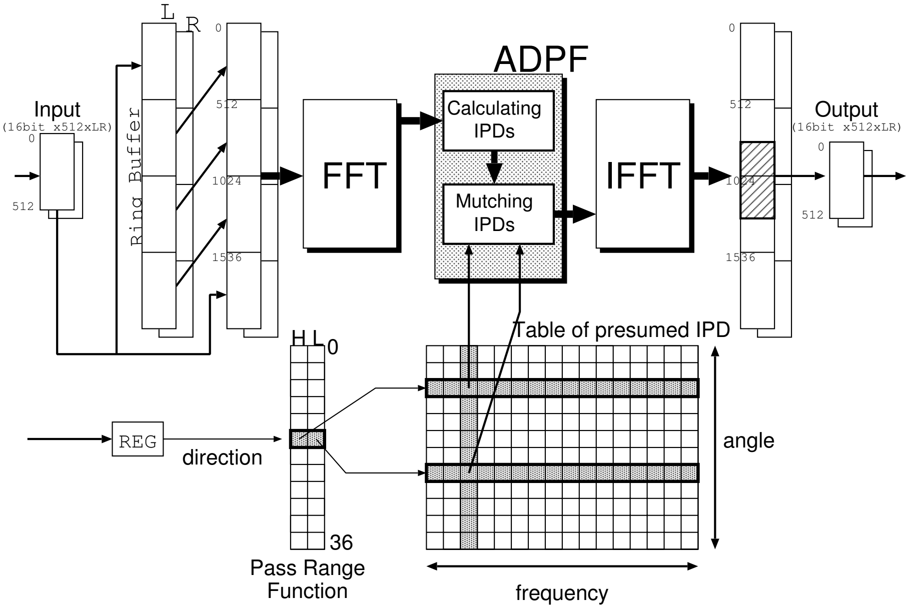


Fig. 6. Active Direction-Pass Filter on DRP

from real and imaginary part of spectrum. The phase is obtained by referring to the table according to this index. It has been designed to satisfy at least the minimum precision required for this application.

- *Matching between actual and presumed IPD* : The direction to be extracted can be designated by the numbers of 0-36 which are assigned to every 5 degrees from -90 to +90 of robot coordinate. It is directly written into a specific register on DRP by using dedicated API. The presumed IPD is obtained by referring table created pursuant to the scattering theory. The pass range function δ is also implemented by the table reference. By using both tables, the upper and lower bound of the IPD $\Delta\varphi(\theta_l)$, $\Delta\varphi(\theta_h)$ are obtained (As shown in Figure 6). The subband filtering is pipelined to improve throughput, and operations for each subband are processed in parallel.

In consequence of implementation, the prototype of the ADPF on DRP can operate with 28.9MHz clock. All tasks were divided into sixteen contexts, The maximum number of PE consumed in each context was 105; it is approximately comparable to only the 1/5 of the resources of DRP-1 chip. While, a number of PE required for whole process was 988. This means that consumed resources are reduced to about 1/10 by time-multiplexed execution.

V. Performance Evaluation

A. The accuracy of extraction

Using the ADPF implemented on DRP, we separated some mixture of two simultaneous speeches. These speeches, in which the sound source exists at 0° and 60° of robot coordinate, were recorded by using the upper torso humanoid SIG[5]. The improvement of SNR (signal-to-noise ratio) between input and separated speech, which is defined by following equation, was calculated as a metric for evaluating an accuracy of extraction.

$$R = 10\log_{10}\left(\frac{\sum_{j=1}^{n}\sum_{i=1}^{m}(|s_o(i,j)| - \beta|s_c(i,j)|)^2}{\sum_{j=1}^{n}\sum_{i=1}^{m}(|s_o(i,j)| - \beta|s_s(i,j)|)^2}\right)$$

where, $s_o(n)$, $s_c(n)$, and $s_s(n)$ are the original signals, the signals captured by robot microphones and the signals separated by the ADPF respectively. β is the attenuation ratio of amplitude between original and captured signals. For an evaluation, we separated 10 speeches and used an average of their results. The results of an experiment are shown as Figure 7 in which the case of separation by software is also shown for comparison.

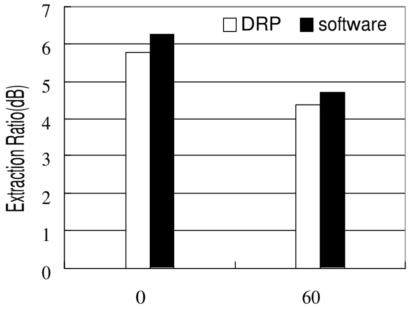


Fig. 7. The improvement of SNR

Figure 7 shows that ADPF on DRP can separate the speech of specific direction from mixture of two simultaneous speeches with comparable accuracy to software. It indicates the feasibility of using DRP for reducing CPU work. One of the possible reason why the improvement of DRP is inferior to the software is the precision of computing which is resulted from fixed-point calculation. And the other is that ADPF of software uses IID for high-frequency domain against DRP which does not take IID into account. Therefore it admits of improvement about using IID by which a better accuracy is expected.

In the case of an FPGA, extension of new function such as using IID implies an increase of chip area required. While, a DRP can be extended with almost no additional cost because of multi-context functionality. Thus, it is one of the advantages of using dynamically reconfigurable device that a chip area is not a critical issue when new functions are appended.

B. The performance for operation

As the other metric, we measured the execution time of separation. In terms of practical use, there is a strict requirement for operation with a certain deadline. When the sampling rate is 16kHz and window shift is 512points, the throughput requirement for real-time separation is more than 2.1Mbps.

Table II shows the execution speed of the ADPF with the DRP and the PC with Pentium 4 2.80GHz. The operating system of the PC is Linux 2.4.27 and gcc-3.3.5 with -O3 option are used. The algorithm used here is the simplified one without using the IID(the same one as DRP). "Clock Cycles" and "Time" stand for a demand per each operation for 2048 samples. Then, their throughputs are worked out for the whole operation and the part confined to ADPF in which an operation of FFT/IFFT is excluded.

TABLE II

THE EXECUTION TIME FOR SEPARATION

	DRP	Pentium4
Frequency	28.9 MHz	2.80 GHz
Clock Cycles	100,928	5,111,368
Time (msec/op)	3.48	1.83
Throughput(ALL)	18.8 Mbps	35.9 Mbps
Throughput(ADPF)	309 Mbps	244 Mbps

Although the performance of the DRP is about a half of that of the common PC with Pentium 4, the throughput requirement for real-time separation will be fulfilled. When we focus attention on the part of the ADPF, the DRP can outstrip the CPU. That is resulted from pipelined and parallel execution of the ADPF on the DRP.

The common PC with Pentium 4 cannot be used in most mobile robots because of its high power consumption and exothermic heat. The DRP is more suitable for mobile robots because it achieve the performance required for real-time operation with much lower operating frequency which implies that power consumption and heat become lower.

In the near future, the DRP core will be integrated with a simple embedded CPU. This performance evaluation suggests that the ADPF execution on such a system can be useful for mobile robots.

VI. CONCLUSION AND FUTURE WORK

We implemented the first prototype of Active Direction-Pass Filter on a Dynamically Reconfigurable Processor DRP-1. It appears that the performance for sound source separation in the certain deadline with acceptable accuracy is possible. That indicates the feasibility of dynamically reconfigurable device for perceptual processing on robots such as the ADPF. The advantage of dynamically reconfigurable device is time-multiplexed execution which leads the reduction of a physical die size. This work actually proved a significant reduction of resources consumed than the case without time-multiplexed execution.

Although the power consumption is expected to be low because of its low operating frequency and its feature that only necessary tasks are carried out, a concrete evaluation of power consumption is our indispensable future work. The comparison of power and area with commercial FPGA is especially essential.

The dynamically reconfigurable device is beneficial for building a small, low power and flexible system, hence it is noted in the field of embedded systems. As shown in this work, it is also congruous with a requirement of applications for mobile robots.

ACKNOWLEDGMENT

Throughout this work, the DRP-1 Device and its design and synthesis tools were provided from NEC/NEC Electronics. The authors would like to show our gratitude to all members of the DRP development group at NEC Electronics and NEC Laboratories for their design tool support and considerable amount of technical advice.

REFERENCES

[1] IPFlex Inc. http://www.ipflex.com/.

[2] K.Nakadai, D.Matsuura, H.G.Okuno, and H.Kitano. Applying Scattering Theory to Robot Audition System. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-2003)*, pages 1147–1152, 2003.

[3] K.Nakadai, H.G.Okuno, and H.Kitano. Robot Recognizes Three Simultaneous Speech By Active Audition. In *Proceedings of IEEE-RAS International Conference on Robots and Automation (ICRA-2003)*, pages 398–403, 2003.

[4] K.Nakadai, K.Hidai, H.G.Okuno, and H.Kitano. Real-Time Speaker Localization and Speech Separation by Audio-Visual Integration. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA-2002)*, pages 1043–1049, 2002.

[5] K.Nakadai, H.G. Okuno, and H.Kitano. Active Audition for Humanoid. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence (AAAI-2000)*, pages 832–839, 2000.

[6] M. Motomura. A Dynamically Reconfigurable Processor Architecture. *Microprocessor Forum*, October 2002.

[7] QuichSilver Technology, Inc. http://www.qstech.com/.

[8] N. Suzuki, S. Kurotaki, M. Suzuki, N. Kaneko, Y. Yamada, K. Deguchi, Y. Hasegawa, H. Amano, K. Anjo, M. Motomura, K. Wakabayashi, T. Toi, and T. Awashima. Implementing and Evaluating Stream Applications on the Dynamically Reconfigurable Processor. In *Proceedings of International Conference on Field-Programmable Custom Computing Machines (FCCM2004)*, pages 328–329, 2004.