https://doi.org/10.20965/jrm.2017.p0016

Paper:

Development, Deployment and Applications of Robot Audition Open Source Software HARK

Kazuhiro Nakadai[*,***], **Hiroshi G. Okuno**[**], **and Takeshi Mizumoto**[*]

[*]Honda Research Institute Japan Co., Ltd.
8-1 Honcho, Wako-shi, Saitama 351-0114, Japan
E-mail: {nakadai, t.mizumoto}@jp.honda-ri.com
[**]Graduate Program for Embodiment Informatics, Waseda University
2-4-12 Okubo, Shinjuku, Tokyo 169-0072, Japan
E-mail: okuno@aoni.waseda.jp
[***]Graduate School of Information Science and Engineering, Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552, Japan
[Received July 29, 2016; accepted October 5, 2016]

Robot audition is a research field that focuses on developing technologies so that robots can hear sound through their own ears (microphones). By compiling robot audition studies performed over more than 10 years, open source software for research purposes called HARK (Honda Research Institute Japan Audition for Robots with Kyoto University) was released to the public in 2008. HARK is updated every year, and free tutorials are often held for its promotion. In this paper, the major functions of HARK – such as sound source localization, sound source separation, and automatic speech recognition – are explained. In order to promote HARK, HARK-Embedded for embedding purposes and HARK-SaaS used as Software as a Service (SaaS) have been actively studied and developed in recent years; these technologies are also described in the paper. In addition, applications of HARK are introduced as case studies.

Keywords: robot audition, open source software, microphone array processing, embedded software, cloud service

1. Introduction

We proposed robot audition studies with the aim of enabling a robot to distinguish sound by using a microphone attached to the robot [1]. Its uniqueness lies in that this research field is originated from Japan, while, in Japan, many research fields are imported from other countries. If this capability is achieved, it would be applicable not only in human-robot verbal communication but also in numerous applications such as human search in a disaster-stricken area, noise detection, hands-free application in automobiles and mobile devices, support for persons having a hearing difficulty, and sound monitoring in a natural environment for the behavior analysis of animals. *HARK (Honda Research Institute Japan Audition for Robots with Kyoto University)*[1] is *Open Source Software (OSS)* developed to share robot audition research results obtained over a period of more than 10 years [2]. HARK was released to the public in 2008 and continues to be developed and promoted.[2]

HARK is equipped with multiple online algorithms for robust signal processing of sound source localization, separation, and tracking against noise or environmental change by using a microphone array consisting of multiple microphones. These algorithms are encapsulated for different functions and are provided as modules; even researchers who lack sufficient knowledge of signal processing or speech processing can easily combine and embed the modules in their own systems. This feature can help in further development of the research field and can simultaneously stabilize the HARK system by incorporating feedback from users.

In the following sections, we introduce the functions of the latest version of HARK, HARK 2.3, released in December 2016; further, we describe HARK-Embedded, which is intended to be embedded in a system, and HARK-SaaS, which can be used as a cloud service, i.e., *Software as a Service (SaaS)*. These extensions are part of the HARK development activities. In addition, a HARK promotion activity, which has been conducted since the release of the software, is explained.

2. Overview of HARK

HARK has been developed with a primary emphasis on 1) the use of HARK not only on offline simulation but also in an implementation on an actual robot and 2) the use of HARK by those who lack sufficient knowledge of signal processing or speech processing. The following sections discuss these two issues and related work.

1. Hark is a medieval word that means "listen."
2. http://www.hark.jp/ [Accessed January 23, 2017]

2.1. Online Processing

In order to achieve the first objective, the processing flow control program "batchflow," which is included in the middleware FlowDesigner [3], is used to realize a modular structure, while maintaining minimal overhead between modules. The modules are implemented as a shared library and are dynamically linked for execution. Therefore, communication between the modules can be realized by simple function calls, and the overhead is less than that in middleware based on socket-based data communication. The number and layout of the microphones can be changed for each robot. The microphones must be calibrated; therefore, a transfer function between the microphone array and sound source is required in advance. The transfer function can be acquired by acoustic simulation with the given positions of the microphone and sound source, or by actual acoustic measurement. HARK also provides the tools that support these operations.

In addition to the microphones, an A/D conversion device for multiple channels is also necessary for the development of a microphone array. In principle, a device that supports drivers such as ALSA (Advanced Linux Sound Architecture), Direct X, and ASIO (Audio Stream Input Output) can be used directly without any changes.

In general, offline signal processing exhibits better performance, and therefore, in many cases, only offline processing implementation is programmed; however, because robots are required to work online and perform real-time actions, all the modules are enhanced for online execution. Further, processing such as sound source localization or sound source separation enables processing that can adaptively respond to an environmental change. Also, batch processing of data saved in a file or offline processing without adaptive processing can be achieved. The realization of offline processing and online processing in the same execution environment could significantly improve work efficiency.

Users may want to connect HARK implemented on a robot to an existing system that they have developed. In order to meet this requirement, seamless connection to the de facto standard middleware *Robot Operating System (ROS)*[3] is made available to robots. Further, a function to create a module by using Python, not C/C++, is available to meet the demand of users who wish to develop their original module in a relatively easy manner.

2.2. User-Friendly Design

With regard to the second objective, HARK provides a programing environment based on *Graphical User Interface (GUI)*. HARK Designer, shown in **Fig. 1**, is a web browser-based GUI environment that is independent of the operating system and allows programming in an environment with almost the same look-and-feel. Each function is modularized and presented as a single node (box in the right panel). Programming is performed by arranging the nodes on the GUI screen and connecting them with lines. Each function can be configured in a setting window, which appears when the corresponding node is double-clicked. Perspicuity is enhanced by encapsulating the functions and listing minimal information.

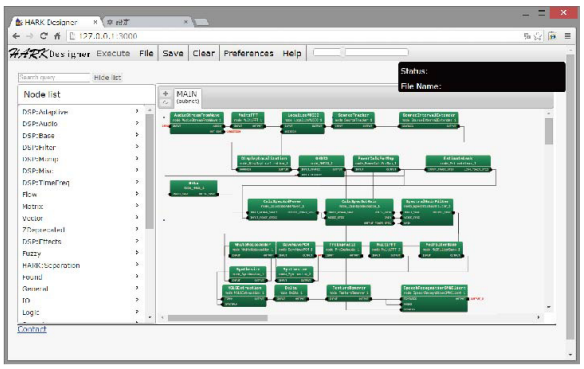


Fig. 1. GUI environment of HARK: HARK Designer.

Most of the function settings can be used without modifications, although some settings must be adjusted. A reduction in user effort by minimizing the number of settings to be adjusted is one of the guidelines in the design of HARK. If users need to set certain parameters (e.g., in order to use the original microphone array, the parameters must be adjusted), some knowledge is required; this requirement could be a barrier for users who are introduced to HARK for the first time. As mentioned in Section 7, free tutorials are provided to share information about parameter adjustment. Further, over 300-page rich documentations such as the HARK Document and the HARK Cook Book explaining the usage of HARK are released to provide the information in written form in both Japanese and English editions. There is also a help desk that responds to questions sent by e-mail.

Installation is easy; apt-get on Linux and a specific installer on Windows are used for easy installation. Although installation would be easier if all necessary components are provided in a single binary file, in some cases, the components must be supplied in separate packages owing to different licenses from third-party libraries. Users must accept license agreements by pressing an OK button each by each during the installation; this process is cumbersome for both users and developers.

Although users can use microphone arrays of their own making, they select commercially available microphones. HARK supports several microphone arrays including Microsoft Kinect for Windows (four microphones), Sony PlayStation®Eye (four microphones), Dev-Audio Microcone (seven microphones),[4] and System in Frontier Tamago (eight microphones). Transfer function files of these microphone arrays are available on the HARK website to enable easy installation, and users need not perform the measurement operation described above.

3. http://www.ros.org/ [Accessed January 23, 2017]

4. Shipment has been stopped owing to corporate acquisition, as of September 2014.

2.3. Related Studies and Projects

ManyEars,[5] which shared middleware with HARK, is the only package with functions similar to HARK; however, microphone array processing has been investigated further, and the following packages have been released in recent years:

- BeamformIT (University of California, Berkeley)

  This package consists of filter-and-sum beamformers and was developed at UC Berkeley. It is implemented using C++ and can be linked to the *Automatic Speech Recognition (ASR)* engine Kaldi.

- BTK (Karlsruhe University, Saarland University, Carnegie Mellon University)

  This package has been released as a tool kit for distant speech recognition (DSR). Several types of beamforming methods such as delay-and-sum beamforming are implemented. It is implemented using C++ and an interface to Python is also provided. It can be linked to Millennium ASR, which is based on finite state transducer.

- ManyEars (University of Sherbrooke)

  In this package, sound source localization by high-speed 2D beamforming, and sound source separation by Geometric Source Separation which a hybrid algorithm between blind separation and beamforming are implemented. The previous version worked on FlowDesigner and was compatible with HARK; however, the current version is a stand-alone package without middleware.

- MESSL (Model-based Expectation-Maximization Source Separation and Localization) (Columbia University)

  It is a sound source localization and separation package based on a binaural model, and it provides functions that are similar to those of HARK-Binaural+.

- FASST (Flexible Audio Source Separation Toolbox) (Institut National de Recherche en Informatique et en Automatique (INRIA))

  It was originally developed as a MATLAB toolbox. The core component of the latest version is implemented using C++, and it also supports a Python interface. Further, it supports single-channel sound source separation such as Non-negative Matrix Factorization (NMF).

Links to the above packages are provided on INRIA's website.[6] When compared with these packages, the advantages of HARK are the following: all the processing from microphones to ASR is provided, 11 types of sound source separation algorithms are supported, and noise-robust sound source localization algorithms based on *MUltiple SIgnal Classification (MUSIC)* is supported. Another advantage of HARK is that in addition to its simple release, e-mail support, detailed documents in Japanese and English, free tutorials, hackathons, and other activities are available for continuous development and support.

5. Current implementation of ManyEars is a C-based package with no middleware.
6. https://wiki.inria.fr/rosp/Software#Speech_enhancement_and_separation [Accessed January 23, 2017]

Table 1. HARK package list.

Package name	Remarks
HARK	Module group of HARK
JuliusMFT	Automatic speech recognition
HARKDesigner	HARK GUI environment
HARK-ROS	Interface between HARK and ROS
HARK-Python	Interface for Python
HARK-OpenCV	Interface with OpenCV
HARK-Kinect	Interface with Kinect
HARK-MUSIC	Music processing
HARK-Binaural	Binaural processing
HARK-Binaural+	Binaural processing
HARK-SSS	11 sound source separation algorithms
HARK-Rescue	Speech enhancement
HARK-Ene	Ego-noise estimation
Wios	Recording tool
harktool4	Transfer function generation tool
HARK-For-Windows	Windows package

3. Major Functions of HARK

HARK 2.3 provides the packages listed in **Table 1** and supports commonly used libraries and languages such as ROS, Python, and OpenCV. The modules included in HARK are shown in **Table 2**. Three major functions of HARK – sound source localization, sound source separation, and ASR – are explained below.

3.1. Sound Source Localization

Sound source localization methods based on MUSIC [4] is provided. MUSIC is a method based on eigenvalue decomposition; it is noise-robust because it yields a sharper peak in the sound source direction than ordinary beamformers. However, a problem with the algorithm is that it mis-localizes a noise source as a target sound source if the noise level exceeds the target sound level. In order to solve this problem, HARK provides the GEVD-MUSIC method based on *Generalized Eigen-Value Decomposition (GEVD)* and the GSVD-MUSIC method based on *Generalized Singular Value Decomposition (GSVD)* [5, 6]. These methods allow the localization of a target sound source in the presence of an extremely high noise level by using noise knowledge called the noise correlation matrix. Further, iGEVD-MUSIC and iGSVD-MUSIC methods that incrementally estimate the noise correlation matrix are provided to respond to a dynamic change in noise. It was reported that with a microphone array with a quadrocopter, a sound source could be localized even in the presence of the propeller sound [7].

3.2. Sound Source Separation

A sound source separation method called *Geometric High-order Decorrelation Source Separation*

Table 2. List of modules provided by HARK.

Function	Category name	Module name	Remarks
Speech input/ output	AudioIO	AudioStreamFromMic	Acquiring sound from microphone
		AudioStreamFromWave	Acquiring sound from file
		SaveRawPCM/SaveWavePCM	Saving sound in file
		HarkDataStreamSender	Output of HARK data through network
		PlayAudio	Play audio signals
Sound source localization/ tracking	Localization	ConstantLocalization	Output of fixed localized value
		DisplayLocalization	Display of localization result
		LocalizeMUSIC	Sound source localization
		LoadSourceLocation	Acquiring localization information from file
		NormalizeMUSIC	Normalizing MUSIC spectrum for easy thresholding
		SaveSourceLocation	Storing localization information to file
		SourceIntervalExtender	Forward extension of tracking result
		SourceTracker	Sound source tracking
		SourceTrackerPF	Particle filter based sound source tracking
		Module starting with CM	Operation of correlation matrix (11 modules)
Sound source separation	Separation	BGNEstimator	Background noise estimation
		BeamForming	Sound source separation algorithms provided with HARK-SSS
		CalcSpecSubGain	Subtraction of noise spectrum and estimation of optimum gain coefficient
		CalcSpecAddPower	Power spectrum addition
		EstimateLeak	Estimation of leak noise between channels
		GHDSS	Sound source separation by GHDSS
		HRLE	Noise spectrum estimation
		PostFilter	Post-filter processing after sound source separation
		SpectralGainFilter	Speech spectrum estimation
Feature extraction	FeatureExtraction	Delta	Δ term calculation
		FeatureRemover	Term deletion
		MelFilterBank	Mel-filter bank processing
		MFCCExtraction	MFCC extraction
		MSLSExtraction	MSLS extraction
		PreEmphasis	Pre-emphasis
		SaveFeatures	Storing features
		SaveHTKFeatures	Storing features in HKT form
		SpectralMeanNormalization	Average normalization of spectrum
Missing feature mask	MFM	DeltaMask	Δ mask term calculation
		DeltaPowerMask	Δ power mask term calculation
		MFMGeneration	MFM creation
Communication with ASR	ASRIF	SpeechRecognitionClient	Sending features to ASR
		SpeechRecognitionSMNClient	Sending features and SMN to ASR
Others	MISC	ChannelSelector	Channel selection
		CombineSource	Combining two localization results
		DataLogger	Data log output
		HarkParamsDynReconf	Dynamic setting of module parameters
		MatrixToMap	Conversion from Matrix to Map
		MatrixToMatrix	Conversion from Matrix float to Matrix complex float
		MatrixToVector	Conversion from Matrix to Vector
		MultiDownSampler	Down sampling
		MultiFFT	Multichannel FFT
		MultiGain	Multichannel gain calculation
		PowerCalcForMap	Power calculation of Map input
		PowerCalcForMatrix	Power calculation of Matrix input
		SegmentAudioStreamByID	Sound stream segment selection by ID
		SourceSelectorByDirection	Sound source selection by direction
		SourceSelectorByID	Sound source selection by ID
		Synthesize	Waveform transformation
		VectorToMap	Conversion from Vector to Map
		VectorToMatrix	Conversion from Vector to Matrix
		VectorToVector	Conversion from Vector float to Vector complex float
		WhiteNoiseAdder	White noise addition
ASR	batch flow independent process	JuliusMFT	ASR for HARK based on Julius
		KaldiDecoder	ASR for HARK based on Kaldi
Data creation	External tool	hark-tool	Data visualization, creation of various configuration files

(GHDSS) [8] is provided. GHDSS is a hybrid-type sound source separation method between beamforming and blind separation. The method is further extended by using an adaptive step size method called the *GHDSS with Adaptive Stepsize control (GHDSS-AS)* to adapt to an environment with dynamic changes in sound, such as a moving sound source. In general, the GHDSS-AS method achieves high separation performance in an actual environment; further, this method is applied to several demonstrations such as simultaneous speech recognition using robots and so on. Various sound source separation algorithms with different features have been developed. Methods other than GHDSS-AS could be advantageous in some cases. Therefore, HARK has a package called HARK-SSS to provide 11 typical sound separation methods, including GHDSS-AS, as shown in **Table 3**. The adaptive step size method is also used to extend the methods that can be implemented. In addition to sound

Table 3. Sound source separation by HARK-SSS (BF: beamformer).

Fixed BF	Delay-and-Sum BF (DS-BF) Null-BF (NULL-BF) Weighted Delay-and-Sum BF (WDS-BF) Indefinite term and Least Square Estimator based BF (ILSE-BF) [9]
Explicit use of noise information	Maximum likelihood BF (ML-BF) [10, 11] Maximum Signal-to-Noise Ratio BF (MSNR-BF) [12]
Linearly constrained minimum variance (LCMV)	Base-type (LCMV-BF) [13] Griffith-Jim type (GJ-BF) [14]
Linearly constrained blind separation	Geometric Source Separation (GSS) [15] Geometric Independent Component Analysis (GICA) [16] Geometric High-order Decorrelation based Source Separation (GHDSS) [8]

source separation, speech enhancement and noise estimation techniques such as *Histogram-based Recursive Level Estimation (HRLE)* [17], *Semi-Blind Independent Component Analysis (SB-ICA)* [18], ego-noise estimation [19] and *Online Robust Principal Component Analysis (ORPCA)* [20] are provided.

3.3. Automatic Speech Recognition

An ASR engine *Julius based on Missing Feature Theory (MFT-Julius)*, which was developed based on Julius, is provided. MFT-Julius is implemented based on a missing feature theory – which masks the distortions produced in sound source separation or speech enhancement processing – to improve sound recognition performance.[7] In addition, *Mel-Scale Log Spectrum (MSLS)* [21] is provided as a feature that can minimize the influence of the distortions on sound features. Further, HARK provides the *Mel-Frequency Cepstrum Coefficient (MFCC)*, which is widely used for ASR. However, MSLS is more suitable for microphone array processing because spectrum distortions are distributed over all features if MFCC is used. Speech by a single speaker can be accurately recognized even when a *Signal-to-Noise (S/N)* ratio is approximately -3 dB [2]. HARK 2.3 and successive versions support deep learning-based ASR, Kaldi. A combination of HARK and Kaldi was reported to demonstrate good performance when used as an *In-Vehicle Information (IVI)* system [22].

4. Development for Embedding: HARK-Embedded

Noise mixing in remote speech heard by a robot could occur in various situations other than those involving robots. For example, in the field of *Human-Machine Interface (HMI)*, the operation by the driver has been replaced with hands-free speech recognition according to the driver distraction guideline of the *National Highway Traffic Safety Administration (NHTSA)*.[8] In this case, a microphone receives various noises such as driving noise, music, the sound of an air conditioner, and the speech of passengers. Speech HMI, to which robot audition technology is applied, is effective in solving this problem. In recent years, familiarity with *Information and Communication Technology (ICT)* has increased owing to the popularization of smartphones and tablet devices. In particular, speech recognition has been established as an input method replacing keyboards that require some technique, especially on a small device, or as an input method for children and the elderly who face difficulties in using keyboards. It is difficult to remove noise and recognize speech successfully unless we take our eyes off from the screen and speak close to the microphone. Robot audition technology could solve this problem. For the application of robot audition technology to these situations, a high degree of programming freedom such as the use of OSS for robot audition is not always necessary, and light software that use little resources are often advantageous.

HARK-Embedded has been developed as software that can be embedded to satisfy this requirement.

Architecture of HARK-Embedded

As explained in Section 2, module integration and flow control can be flexibly achieved for HARK by using batchflow as the middleware. However, when used for embedding, a middleware that increases processing overhead would lower the processing speed and increase the amount of resources used. For the embedded version of HARK, the code was optimized for the ARM® architecture and batchflow was eliminated. The elimination of batchflow decreases the flexibility of processing; however, the problem is not a major one because the general processing flow of HARK – from localization to separation, and then to recognition – is fixed. Further, the code was made compact, multi-threading was implemented, the signal processing speed was enhanced by using ARM®'s SIMD (Single Instruction Multiple Data) architecture extension function NEON™, and compiler options were optimized. **Table 4** shows the processing speeds measured for the four-channel microphone array. From the table, we observe that HARK-Embedded in Cortex™ A9 achieves a similar level of performance as conventional OSS-type HARK on Intel® Core™ i7.

7. It was extended from the implementation released by a former Furui Lab team at Tokyo Institute of Technology.
8. http://www.nhtsa.gov/ [Accessed January 23, 2017]

Table 4. Comparison of processing speed and load average (# of microphones: 4, input data length: 1,232 sec).

Software	OSS ver.	HARK-Embedded	
CPU	Core i7 2640M		Cortex A9 Quad (Exynos 4412 Prime)
CPU speed	2.8 GHz		1.7 GHz
BogoMIPS/core	5582		1992
Used Cores	1	4	4
Proc. time	72.8 sec	12.6 sec	69.3 sec
Load ave.	5.9%	0.8%	5.1%

Table 5. Specifications of RASP-MX.

CPU	TI i.MX6Q Cortex A9, 1 GHz Quad
Memory	DDR3, 2 GB, 533 MHz
FPGA	Xilinx Artix-7 (for mic array input)
Microphone	upto 16 ch
WiFi	IEEE 802.11b/g/n, Bluetooth
Voltage	5 V
Power	1 W (idle), 4 W (HARK-Embedded running)
Size	50 mm × 87 mm × 2 mm
Weight	16 g (34 g with heatsink)

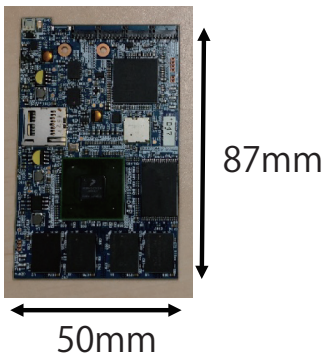


Fig. 2. Processor board RASP-MX6 for HARK-Embedded.

A processing board – RASP-MX[9] – of name card size, on which HARK-Embedded works, was also developed. **Table 5** and **Fig. 2** show the specifications and photograph, respectively, of RASP-MX. This board can be used as a versatile processing board, and a 16-channel microphone array can be configured with the board. With HARK-Embedded installed, the real-time output of separated sound can be obtained.

5. Development for SaaS: HARK-SaaS

OSS-type HARK must be installed on a local computer, and all processing is performed locally in the computer. Thus, in addition to the installation of HARK, a computer with the specifications given in **Table 4** is necessary. Therefore, a cloud service – HARK-SaaS – was developed to use an Internet-based HARK functionalities on a low-specification computer, without installation or other preparatory steps.[10] HARK-SaaS provides a Web API (Application Programming Interface). Therefore, all the functions – such as authentication; HARK processing request; acquisition, update, and deletion of processing results; and re-execution of HARK processing – can be performed by using an HTTPS request encapsulated with the *Transport Layer Security (TLS)* protocol. HARK-SaaS can also be used from a browser using its Web UI.

In HARK-SaaS, a session, which corresponds to a single multi-channel audio file, is handled as a processing unit. First, a session is created on HARK-SaaS and a session ID (unique character string assigned to the session) is acquired. Next, an audio file that corresponds to the session is uploaded. Then, HARK processing begins automatically and polling of the processing status of each session is performed. When the processing ends, the result can be acquired from the session, updated, and deleted. The data structure corresponding to the processing result consists of metadata, context information, and scene information, and it is represented in the JSON (JavaScript Object Notation) format. The metadata contain information – such as the tag of an arbitrary character string or parameters of HARK – for a user-specified session. The context information consists of the information for each sound event; it includes the start and end time of the sound event, direction of the sound, sound volume, and separated sound. The scene information corresponds to the information for the entire session, e.g., the total number of sound sources, the number of sound sources in each direction, and time-series data for the sound volume.

An example of sound environment visualization using HARK-SaaS is presented in Section 6.

6. Case study of HARK

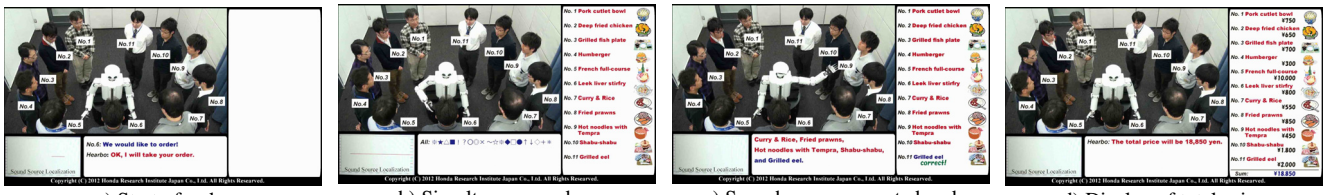
In this section, demonstrations using HARK are presented as case studies.

1. Prince Shotoku Robot: Understanding simultaneous orders from 11 people (**Fig. 3**).
2. Support for multi-language communication with a tablet (**Fig. 4**).
3. Multi-user hands-free information system mounted on a car (**Fig. 5**).
4. Example of sound environment visualization using HARK-SaaS (**Fig. 6**).

In (1), 16 microphones mounted on the head of a robot captured orders placed simultaneously by 11 people. GHDSS-AS was used to achieve sound source separation of the orders and ASR of the separated speeches. Then, the total price was announced based on the recognized orders. This demonstration was implemented with OSS-HARK, and the robot was controlled with ROS.

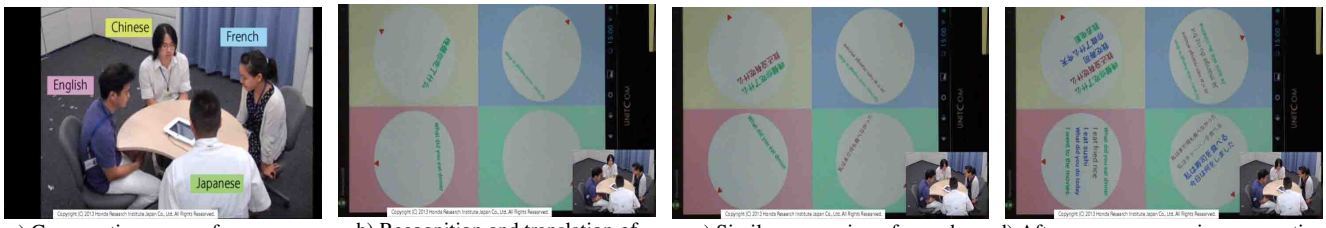
9. It will be released by System In Frontier Inc.

10. https://api.hark.jp/ [Accessed January 23, 2017]



a) Start of order
b) Simultaneous orders from 11 people
c) Speeches are separated and recognized
d) Display of total price

Fig. 3. Robot's understanding of simultaneous orders from 11 people.

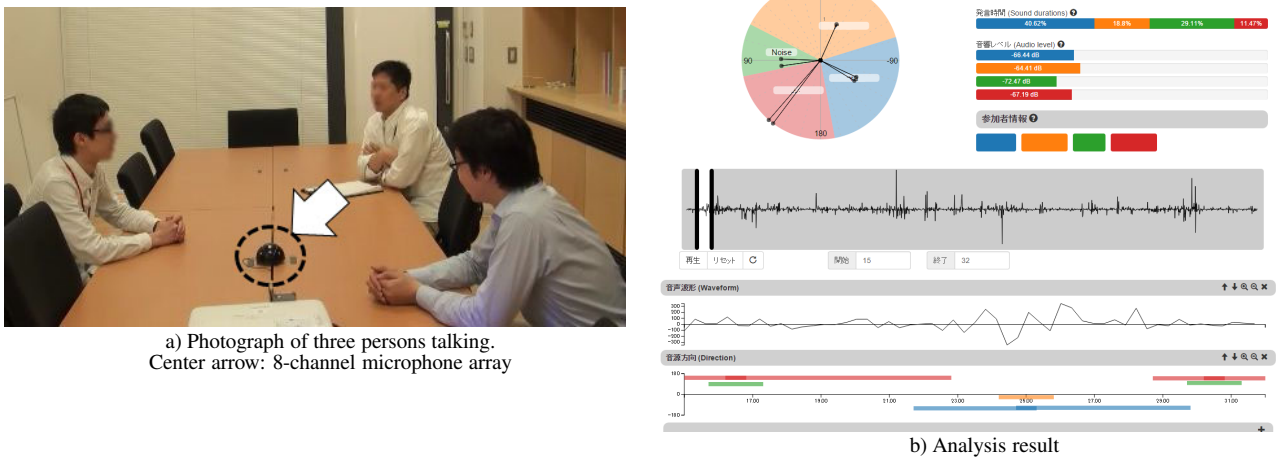


a) Conversation among four persons in different native languages (Japanese, English, Chinese, and French)
b) Recognition and translation of speech in Japanese (right-bottom) and display of results in a direction toward the persons
c) Similar processing of speech in English (left-bottom)
d) After some progress in conversation, display that follows a change in the speaker's position

Fig. 4. Multi-language communication support using a tablet.

a) Driver's seat: Route search (pressing talk button not necessary)
b) Notification of search result (robot: rotates toward driver's seat)
c) Front passenger's seat: Request of a song (passenger's speech accepted)
d) Control of noise-containing audio (speech recognition with music noise)

Fig. 5. Multi-user hands-free information system mounted on car.



a) Photograph of three persons talking. Center arrow: 8-channel microphone array
b) Analysis result

Fig. 6. Example of sound environment visualization using HARK-SaaS.

In (2), HARK-Embedded was used for the demonstration, and eight microphones were mounted around a tablet device. First, the system performed sound source localization to acknowledge the direction of each speaker. After speech recognition, the speech of each speaker was translated to the native languages. Finally, the translated text was displayed in a direction toward the speakers. Therefore, the speakers could see the text easily. In order to support people with hearing impairments and persons who are audibly handicapped, the cocktail party effect has been emphasized for listening to a single speech. However, support for a multi-party effect – i.e., for listening to speech from multiple persons – is necessary in daily life. Therefore, the application of the present system will be more important in the future. In this demonstration, cloud services for ASR and translation were used.

For the localization and separation of sound sources, an ARM®-based thin board RASP-MX – in which HARK-Embedded worked – was used without consuming the computational resource of the tablet for processing purposes. This board can be connected through Bluetooth®; therefore, the microphone array can be connected to an existing service such as Google Voice Search™ or Apple Siri®, and can be used as a Bluetooth® microphone.

In (3), an example in which HARK-Embedded was applied to an information system mounted on a car is presented. Typically, the talk button of a car-mounted ASR system is located near the steering wheel; the driver must press the button before beginning to speak. When the button is pressed, the volume of the music or the air flow of the air conditioner is automatically lowered to reduce the noise. This type of ASR system was designed for drivers; therefore, if a passenger speaks to the system, the system would ignore the speech or recognize the speech erroneously. Therefore, the sound source localization and separation functions of HARK were introduced to solve this problem. The car-mounted information system can be used for ASR at any point of time from the driver's seat and the passenger's seat by using a 4-channel microphone array mounted on the map lamp without changing the volume of music [22].

In (4), an example of the application of HARK-SaaS is presented. **Fig. 6a)** shows a photograph of a sound environment in which three speakers are chatting. The half-dome device in the center is a microphone array with eight microphones and is connected to a laptop computer through USB (Universal Serial Bus). **Fig. 6b)** shows the visualization result. When a speech file is uploaded, it is analyzed by HARK-SaaS; further, a visualization library d3.js[11] is used to display the direction of the sound, the number ratio of the sound events in each direction, and the sound level on a web browser. This application uses only HARK-SaaS; however, it can be easily combined with various other existing systems or cloud services. New ideas would extend robot audition.

7. Promotion of HARK

Table 6 shows the release dates for HARK and a list of tutorials. The software has been updated almost every year, and tutorials corresponding to version updates have been held in Japan and foreign countries. The number of participants in each tutorial was limited to approximately 50, and the attendance was almost always at maximum capacity. Although the software is for research purposes, the participants included many employees of companies. During and after 2014, in addition to tutorials, hackathons were held to accelerate the promotion of HARK.

Overseas expansion has also been an active area of focus. In March 2010, on invitation from Willow Garage in USA, the authors ported HARK to a telepresence robot, Texai [23]. Texai is an agent robot that has a physical body and moves around a room chatting with people; it is remotely controlled by a user (remote user). However, some problems exist: the remote user is unable to determine who is currently speaking and cannot understand what is being spoken if a large amount of noise exists. Therefore, we implemented new functions to identify the direction of a sound source on a camera image and to capture sound from a sound source located in a specific direction, based on the visualization of localization information and the development of GUI to control the direction of a separated sound source. Within a week, seven people – including three faculty members – completed robot head machining, microphone calibration, preliminary experiments, and the design and implementation of the GUI and operation commands. We believe that the high modularity of HARK and ROS helped in improving the productivity.

Over a period of a month from November to December 2010, two of our students ported HARK to HRP-2 at CNRS-LAAS (French National Center for Scientific Research – Laboratory for Analysis and Architecture of Systems), France. They conducted operation tests for HARK and created a sound input interface to use a microphone called Ear Sensor, which is under development at CNRS-LAAS, for HARK. Then, the collaboration with LAAS led to a joint research project, BINNAHR (BINaural Active Audition for Humanoid Robots).[12] This sequence of

Table 6. Release of HARK and list of tutorials.

Apr., 2008: First release (0.1.7) First tutorial meeting: 11/17/ 2008 at Kyoto University, Japan Second tutorial meeting: 12/5/2008 at KIST, Seoul, Korea
Nov., 2009: Pre-release of 1.0.0 Third tutorial meeting: 1/7/2009 at Keio University, Hiyoshi, Japan Fourth tutorial meeting: 12/7/2009 at UPMC, Paris, France
Nov., 2010: Major version upgrade (1.0.0) Enhancement of sound source separation, more detailed document Fifth tutorial meeting: 11/25/ 2010 at Kyoto University, Japan
Feb., 2012: Version upgrade (1.1) Enhancement of sound source separation, 64bit, ROS support Sixth tutorial meeting: 2/29/2012 at UPMC, Paris, France Seventh tutorial meeting: 3/9/2012 at Nagoya University, Japan
Mar., 2013: Version upgrade (1.2) Kinect, PSEye support Eighth tutorial meeting: 3/19/2013 at Kyoto University, Japan
Oct., 2013: Version upgrade (1.9.9) Windows & HarkDesigner version α Ninth tutorial meeting: 10/2/2013 at CNRS-LAAS, Toulouse, France
Dec., 2013: Major version upgrade (2.0) Windows & HarkDesigner support Tenth tutorial meeting: 12/5/2013 at Waseda University, Japan
Nov., 2014: Version upgrade (2.1) Ego-noise suppression Eleventh tutorial meeting: 11/19/2014 at Waseda University, Japan
Nov., 2015: Version upgrade (2.2) HARK-Python Windows, Binaural+, sound play module added Twelfth tutorial meeting: 11/10/2015 at Waseda University, Japan
Dec., 2016: Version upgrade (2.3) Kaldi support, Particle filter based tracking module added Thirteenth tutorial meeting: 6/12/2016 at Waseda University, Japan

11. http://d3js.org/ [Accessed January 23, 2017]

12. http://projects.laas.fr/BINAAHR/BINAAHR/Welcome.html [Accessed January 23, 2017]

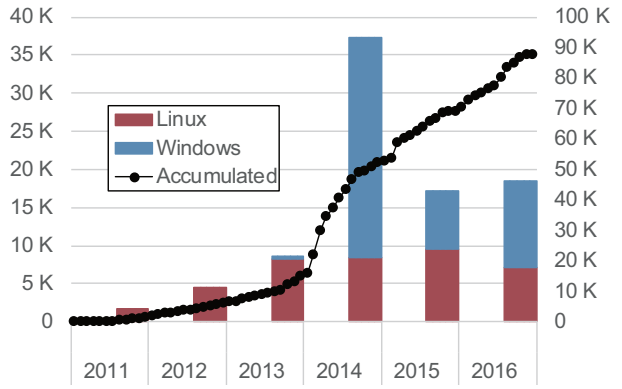


Fig. 7. Download history of HARK.

events is an example of the successful promotion activity of HARK.

Figure 7 shows the number of downloads until the end of November 2016. The total number of downloads was approximately 90,000. This number temporarily increased in 2014; the reason for the increase could be the numerous downloads of the newly released Windows version. Except for this phenomenon, the total number of downloads has increased every year; therefore, the activities described in this section appear to have been effective in the promotion of HARK.

8. Conclusion

In this paper, an overview of the open source software, HARK, released in 2008 for research purposes was provided as an achievement in robot audition studies. The creation of an embedded version of HARK (HARK-Embedded) and a cloud service of HARK (HARK-SaaS) were introduced as development activities of HARK. Prince Shotoku Robot, support for multi-language communication with a tablet, a multi-user hands-free information system mounted on a car, and an example of sound environment visualization were introduced as case studies of HARK. In Japan, robot audition research has been conducted as studies on sound in an extreme situation in the ImPACT Tough Robotics Challenge. In Europe, robot audition research projects such as TWO!EARS and EARS were launched. Thus, activity in robot audition studies has increased and the importance of these studies will increase. We are seeking not only users but also people who wish to participate in the development. Please feel free to contact us if you are interested.

Acknowledgements

For the overall development and support of HARK, the authors would like to thank all members of the HARK development team at Honda Research Institute Japan, Kyoto University and Tokyo Institute of Technology.

References:

[1] K. Nakadai et al., "Active Audition for Humanoid," AAAI-2000, pp. 832-839, 2000.
[2] K. Nakadai et al., "Design and Implementation of Robot Audition System "HARK"," Advanced Robotics, Vol.24, pp. 739-761, 2010.
[3] C. Côté et al., "Code reusability tools for programming mobile robots," IEEE/RSJ IROS 2004, pp. 1820-1825, 2004.
[4] R. Schmidt, "Multiple emitter location and signal parameter estimation," IEEE Trans. on Antennas and Propagation, Vol.34, No.3, pp. 276-280, 1986.
[5] F. Asano et al., "Localization and extraction of brain activity using generalized eigenvalue decomposition," IEEE ICASSP 2008, pp. 565-568, 2008.
[6] K. Nakamura et al., "A real-time super-resolution robot audition system that improves the robustness of simultaneous speech recognition," Advanced Robotics, Vol.27, No.12, pp. 933-945, 2013.
[7] T. Ohata et al, "Improvement in Outdoor Sound Source Detection Using a Quadrotor-Embedded Microphone Array," IEEE/RSJ IROS, 2014.
[8] H. Nakajima et al., "Blind Source Separation with Parameter-Free Adaptive Step-Size Method for Robot Audition," IEEE Trans. ASLP, Vol.18, No.6, pp. 1476-1484, 2010.
[9] H. Nakajima, N. Tanaka, and H. Tsuru, "Minimum sidelobe beamforming based on Mini-Max criterion," Acoust. Sci. & Tech., Vol.25, No.6, pp. 486-488, 2004.
[10] V. A. N. Barroso and J. M. F. Moura, "Maximum likelihood beamforming in the presence of outliers," IEEE ICASSP-91, pp. 1409-1412, 1991.
[11] M. L. Seltzer et al., "A Bayesian Framework for Spectrographic Mask Estimation for Missing Feature Speech Recognition," Speech Communication, Vol.43, No.4, pp. 379-393, 2004.
[12] R. A. Monzingo and T. W. Miller, "Introduction to adaptive arrays," SciTech Publishing, 1980.
[13] O. L. Frost, "An algorithm for linearly constrained adaptive array processing," Proc. of the IEEE, Vol.60, No.8, pp. 926-935, 1972.
[14] L. J. Griffiths and C. W. Jim, "An alternative approach to linearly constrained adaptive beamforming," IEEE Trans. on Antennas and Propagation, Vol.30, No.1, pp. 27-34, 1982.
[15] L. C. Parra and C. V. Alvino, "Geometric source separation: Mergin convolutive source separation with geometric beamforming," IEEE Trans. on Speech and Audio Processing, Vol.10, No.6, pp. 352-362, 2002.
[16] M. Knaak et al., "Geometrically Constrained Independent Component Analysis," IEEE Trans. on ASLP, Vol.15, No.2, pp. 715-726, 2007.
[17] H. Nakajima et al., "An easily-configurable robot audition system using Histogram-based Recursive Level Estimation," IEEE/RSJ IROS 2010, pp. 958-963, 2010.
[18] R. Takeda et al., "Efficient Blind Dereverberation and Echo Cancellation Based on Independent Component Analysis for Actual Acoustic Signals," Neural Computation, Vol.24, No.1, pp. 234-272, 2011.
[19] G. Ince et al., "Whole Body Motion Noise Cancellation of a Robot for Improved Automatic Speech Recognition," Advanced Robotics, Vol.25, No.11, pp. 1405-1426, 2011.
[20] Y. Bando et al., "Human-voice enhancement based on online RPCA for a hose-shaped rescue robot with a microphone array," 2015 IEEE Int. Symposium on Safety, Security, and Rescue Robotics (SSRR), pp. 1-6, 2015.
[21] S. Yamamoto et al., "Enhanced robot speech recognition based on microphone array source separation and missing feature theory," IEEE/RAS ICRA 2005, pp. 1427-1482, 2005.
[22] K. Nakadai et al., "Robot-Audition-based Human-Machine Interface for a Car," IEEE/RSJ IROS 2015, pp. 6129-6136, 2015.
[23] T. Mizumoto et al., "Design and implementation of selectable sound separation on the Texai telepresence system using HARK," IEEE/RAS ICRA-2011, pp. 2130-2137, 2011.

Name:
Kazuhiro Nakadai

Affiliation:
Honda Research Institute Japan Co., Ltd.
Tokyo Institute of Technology

Address:
8-1 Honcho, Wako-shi, Saitama 351-0188, Japan
2-12-1-W30 Ookayama, Meguro-ku, Tokyo 152-8552, Japan

Brief Biographical History:
1995 Received M.E. from The University of Tokyo
1995-1999 Engineer, Nippon Telegraph and Telephone and NTT Comware
1999-2003 Researcher, Kitano Symbiotic Systems Project, ERATO, JST
2003 Received Ph.D. from The University of Tokyo
2003-2009 Senior Researcher, Honda Research Institute Japan Co., Ltd.
2006-2010 Visiting Associate Professor, Tokyo Institute of Technology
2010- Principal Researcher, Honda Research Institute Japan Co., Ltd.
2011- Visiting Professor, Tokyo Institute of Technology
2011- Visiting Professor, Waseda University

Main Works:
• K. Nakamura, K. Nakadai, H. and G. Okuno, "A real-time super-resolution robot audition system that improves the robustness of simultaneous speech recognition," Advanced Robotics, Vol.27, Issue 12, pp. 933-945, 2013 (Received Best Paper Award).
• H. Miura, T. Yoshida, K. Nakamura, and K. Nakadai, "SLAM-based Online Calibration for Asynchronous Microphone Array," Advanced Robotics, Vol.26, No.17, pp. 1941-1965, 2012.
• R. Takeda, K. Nakadai, T. Takahashi, T. Ogata, and H. G. Okuno, "Efficient Blind Dereverberation and Echo Cancellation based on Independent Component Analysis for Actual Acoustic Signals," Neural Computation, Vol.24, No.1, pp. 234-272, 2012.
• K. Nakadai, T. Takahashi, H. G. Okuno et al., "Design and Implementation of Robot Audition System "HARK"," Advanced Robotics, Vol.24, No.5-6, pp. 739-761, 2010.
• K. Nakadai, D. Matsuura, H. G. Okuno, and H. Tsujino, "Improvement of recognition of simultaneous speech signals using AV integration and scattering theory for humanoid robots," Speech Communication, Vol.44, pp. 97-112, 2004.

Membership in Academic Societies:
• The Robotics Society of Japan (RSJ)
• The Japanese Society for Artificial Intelligence (JSAI)
• The Acoustic Society of Japan (ASJ)
• Information Processing Society of Japan (IPSJ)
• Human Interface Society (HIS)
• International Speech and Communication Association (ISCA)
• The Institute of Electrical and Electronics Engineers (IEEE)

Name:
Hiroshi G. Okuno

Affiliation:
Professor, Graduate School of Science and Engineering, Waseda University
Professor Emeritus, Kyoto University

Address:
Lambdax Bldg. 3F, 2-4-12 Okubo, Shinjuku, Tokyo 169-0072, Japan

Brief Biographical History:
1996 Received Ph.D. of Engineering from Graduate School of Engineering, The University of Tokyo
2001-2014 Professor, Graduate School of Informatics, Kyoto University
2014- Professor, Graduate School of Science and Engineering, Waseda University

Main Works:
• "Design and Implementation of Robot Audition System "HARK"," Advanced Robotics, Vol.24, No.5-6, pp. 739-761, 2010.
• "Computational Auditory Scene Analysis," Lawrence Erlbaum Associates, Mahmoh, NJ, 1998.

Membership in Academic Societies:
• The Institute of Electrical and Electronic Engineers (IEEE), Fellow
• The Japanese Society for Artificial Intelligence (JSAI), Fellow
• Information Processing Society Japan (IPSJ), Fellow
• The Robotics Society of Japan (RSJ), Fellow

Name:
Takeshi Mizumoto

Affiliation:
Researcher, Honda Research Institute Japan Co., Ltd.

Address:
8-1 Honcho, Wako-shi, Saitama 351-0188, Japan

Brief Biographical History:
2013 Received Ph.D. in Informatics from Graduate School of Informatics, Kyoto University
2013- Researcher, Honda Research Institute Japan Co., Ltd.

Main Works:
• "Sound Imaging of Nocturnal Animal Calls in Their Natural Habitat," J. of Comparative Physiology A, Vol.197, No.9, pp. 915-921, 2011. DOI: 10.1007/s00359-011-0652-7

Membership in Academic Societies:
• The Robotics Society of Japan (RSJ)
• The Institute of Electrical and Electronic Engineers (IEEE)
• The Japanese Society for Artificial Intelligence (JSAI)