# Automatic Speech Recognition for Indoor HRI Scenarios

JOSÉ NOVOA, RODRIGO MAHU, JORGE WUTH, JUAN PABLO ESCUDERO, JOSUÉ FREDES, and NÉSTOR BECERRA YOMA, Speech Processing and Transmission Laboratory, University of Chile

This article presents a stand-alone automatic speech recognition system that accounts for listener movement, time-varying reverberation effects, environmental noise, and user position information for beamforming approaches in an HRI setting. We raise the importance of replacing the classical black-box integration of automatic speech recognition technology in HRI applications with the incorporation of the acoustic environment representation and modeling, and of the target source direction. Test data were recorded on a real robot under various moving conditions. For addressing the time-varying acoustic channel problem and incorporating environmental effect during training, clean speech samples were passed through estimated static channel responses and noise was added. Beamforming is investigated regarding oracle source tracking using, for instance, image processing. The proposed strategy is interesting for the robotics community, because it allows the development of voice-based HRI with limited training data and without relying on third-party technologies or Internet access eliminating the need to upload data to the cloud. In our mobile HRI scenario, the resulting speech recognition engine provided an average word error rate that is at least 19% and 34% lower than publicly available speech recognition APIs with the playback (i.e., loudspeaker) and human testing modalities, respectively.

CCS Concepts: • **Computing methodologies** → **Speech recognition**; *Neural networks*; • **Human-centered computing** → **Contextual design**; **Scenario-based design**;

Additional Key Words and Phrases: Beamforming, indoor environments, DNN-HMM, time-varying acoustic channel, ASR

**ACM Reference format:**
José Novoa, Rodrigo Mahu, Jorge Wuth, Juan Pablo Escudero, Josué Fredes, and Néstor Becerra Yoma. 2021. Automatic Speech Recognition for Indoor HRI Scenarios. *ACM Trans. Hum.-Robot Interact.* 10, 2, Article 17 (March 2021), 30 pages.
https://doi.org/10.1145/3442629



## 1 INTRODUCTION

### 1.1 HRI and Speech Technology

If social robotics is a reality, then the appropriate social integration between humans and robots could greatly improve the cooperation between users and machines. There are several

The research reported here was funded by grants Conicyt-Fondecyt 1151306 and ONRG N°62909-17-1-2002.
Authors' addresses: J. Novoa, R. Mahu, J. Wuth, J. P. Escudero, J. Fredes, and N. Becerra Yoma, Av. Tupper 2007, Santiago, Chile, PC 8370451; emails: {jose.novoa, rmahu, jwuth, jescudero, jfredes, nbecerra}@ing.uchile.cl.

2573-9522/2021/03-ART17 $15.00
https://doi.org/10.1145/3442629

applications in defense, hostile environments, mining, industry, forestry, education, and natural disasters where some integration and collaboration between humans and robots will be required [1]. HRI is especially relevant in those situations when robots are not fully autonomous and require interaction with humans to receive instructions or information in decision-making applications [2–5]. In this context, humanlike communication between people and robots is essential for a successful human–robot collaborative symbiosis [6–7]. Additionally, speech is the most straightforward and natural way that humans employ to communicate [8–10]. Consequently, voice-based HRI should be the most natural way to facilitate a collaborative human–robot synergy and speech technology, particularly automatic speech recognition (ASR), should play an important role in social robotics.

It is well known that computer vision is an important research topic in robotics. Recent challenges such as DARPA Robotics Challenge [11] and Robocup [12] have led to great improvements in computer vision [13–16]. However, there has also been a significant progress in ASR, but most of this advancement has taken place outside the HRI field. ASR has gained relevance in robotics over the past few years, but its status is still far from the one enjoyed by computer vision in the robotic research.

### 1.2 Automatic Speech Recognition Technology

ASR is the process and related technology for transcribing human speech into words. By using Bayes's rule, the ASR problem can be formulated as follows [17]:

$$\hat{W} = \underset{W}{\operatorname{argmax}}\, p(W|\boldsymbol{X}) = \underset{W}{\operatorname{argmax}}\, p(\boldsymbol{X}|W) \cdot p(W), \tag{1}$$

where $\hat{W}$ is the optimal label (word or phone) sequence, $\boldsymbol{X}$ is the input speech observation sequence that represents a given speech utterance, $p(W)$ denotes the language model describing the probabilities of word combinations, and $p(\boldsymbol{X}|W)$ indicates the acoustic model. Consequently, the task of an ASR system is to find (by means of a process called decoding, performed with the Viterbi algorithm [18]) the most likely label sequence $\hat{W}$ given an observed sequence of feature vectors that corresponds to the speech utterance. The language model can be represented with [19] statistical models, stochastic context-free grammars (SCFG), or stochastic finite-state models. In the case of statistical models, which are widely employed in research, the prior probability of a word sequence $W = w_1, \ldots, w_L$ in (1) can be approximated with N-grams.

$$p(W) \cong \prod_{l=1}^{L} p(w_l|w_{l-1}, w_{l-2}, \ldots, w_{l-N+1}), \tag{2}$$

where $N$ is typically between 2 and 4. The language model defines the transition probability from one N-gram to the next word to guide the search for an interpretation of the acoustic input. Additionally, the size of the vocabulary and perplexity [20] are critical for ASR accuracy. Perplexity measures the uncertainty about the words that may follow a given N-gram. A low-perplexity language model defined by a given task or context will constrain the decoding and perform better than a high-perplexity one. Accordingly, language models can be adapted using context information to reduce perplexity and improve the ASR accuracy [21–23].

Acoustic modeling defines the statistical representations for the sequence of acoustic feature vectors $\boldsymbol{X}$ obtained from the speech waveform. The utterances are divided into 20 or 30 ms windows with overlap (e.g., 50%). The set of acoustic features is usually obtained from the short-term fast Fourier transform within each window [18, 24–25]. Speed and acceleration coefficients (called delta and delta-delta coefficients) can also be used, and the final feature vector is composed

of the static features, plus the delta and delta-delta coefficients [26]. Mean and variance normalization of the coefficients can also be employed. Until a few years ago, most speech recognition systems adopted hidden Markov models (HMMs), to deal with the temporal variability of speech, and Gaussian mixture models (GMMs) to represent $p(\boldsymbol{X}|W)$. Given a set of speech feature vectors $\boldsymbol{X} = \{\boldsymbol{x}_t\}_{t=1}^{T}$, the state observation probability density function of feature vector $\boldsymbol{x}_t$ at frame $t$ in state $\mathrm{s}_i$ is expressed by [18]

$$p(\boldsymbol{x}_t|\mathrm{s}_i) = \sum_{m=1}^{M} c_{i,m} \cdot \mathcal{N}(\boldsymbol{x}_t; \mu_{i,m}, \Sigma_{i,m}), \tag{3}$$

where $c_{i,m}$, $\mu_{i,m}$, and $\Sigma_{i,m}$ correspond to the mixture weights, mean vectors, and covariance matrices, respectively, for $M$ Gaussian mixture components. In the last few years, artificial neural networks (ANNs), e.g., deep neural networks (DNNs), have shown significant performance improvement over GMM-based models. In a DNN-HMM system, the DNN provides a pseudo-log-likelihood defined as

$$\log[p(\boldsymbol{x}_t|s_j)] = \log[p(s_j|\boldsymbol{x}_t)] - \log[p(s_j)], \tag{4}$$

where $s_j$ denotes one of the states or senones, and the state priors $\log[p(s_j)]$ can be trained using the state alignments obtained with the training speech data. The final decoded word string, $\hat{W}$, is determined by

$$\hat{W} = \underset{W}{\operatorname{argmax}} \left\{\log[p(\boldsymbol{X}|\mathrm{W})] + \lambda \cdot \log[p(W)]\right\}, \tag{5}$$

where the acoustic model probability $p(\boldsymbol{X}|W)$ depends on the pseudo log-likelihood $\log[p(\boldsymbol{x}_t s)]$ delivered by the DNN and $\lambda$ is the constant that is employed to balance the acoustic model and language model scores [27]. The results reported in Reference [28] showed that the DNN-HMM ASR can lead to a WER reduction of 32% relative when compared to the ordinary GMM-HMM system on the Switchboard task [29]. However, training a DNN is not an easy task. The objective function can be highly non-convex and the training algorithm can easily converge to a suboptimal local minimum. Also, ANNs need more training data than GMM-HMM systems [30]. It is worth mentioning that public ANN-based ASR application programming interfaces (APIs) employ at least thousands of hours of speech data for training [31–33]. Other ANN architectures have also been applied to ASR: long short term memory (LSTM) [34], convolutional neural network (CNN) [35], and recurrent neural networks (RNN) [36]. The results obtained using DNN-HMM systems are competitive when compared to those reported with other ANN architectures [37–41]. In some cases, systems employing combinations of ANN architectures, RNN, very deep CNN [42], or fCNN [43] have outperformed DNN, LSTM, or the ordinary CNN approaches. However, the higher the number of the ANN parameters, the higher the required amount of training data.

In matched conditions between training and testing data, ASR shows large performance gain. In contrast, models will have difficulties recognizing test samples if they differ from data used in training. For this reason, noise robustness of ANN-based systems can be achieved by using multistyle or multicondition training [44–46]. For instance, a DNN trained with several types of noise and signal-to-noise ratio (SNR) levels can lead to a high accuracy improvement in real applications [47].

The reverberation effect is usually modeled as the convolution of clean speech with a room impulse response. The time-invariant hypothesis is necessary for this model. If the robot or its head moves, then the acoustic channel becomes time-varying, and the reverberation effect cannot be represented as a convolution and treated analytically anymore. This problem is denoted by time-varying acoustic channel (TVAC) and is discussed in Section 3.3.1. In this article, we propose that ASR technology should be investigated, designed and developed to address HRI applications


Fig. 1. Ordinary black box-based ASR integration in HRI scenarios.

by taking into consideration the acoustic environment. Following this strategy, first this article focuses on the acoustic environment representation and modeling by training an ASR engine by combining clean utterances with the acoustic channel impulse responses and noise that were estimated and recorded, respectively, with an HRI testbed. The acoustic channel impulse responses address the TVAC problem, and noise was generated by the robot itself and up to two external noise sources. This testbed represents the generic problem of HRI in mobile robotics and the resulting ASR accuracy can outperform publicly available ASR APIs with a limited amount of training data. Second, beamforming technology and toolkits were evaluated by assuming that the direction of speech target source was known by other means than audio analysis, e.g., image detection and classification technology. This strategy represents a step toward a more complete ASR integration to HRI scenarios. Finally, another motivation for the research described in this article is to generate voice-based HRI technology that does not depend on cloud-based ASR, not only for privacy concerns but also to make it independent of Internet access by making use of limited training data.

## 2 RELATED WORK

### 2.1 Black Box-based Integration of ASR Technology

Most of the research that considers ASR in HRI scenarios use ASR toolkits or APIs as black boxes. A non-exhaustive list of available options that support ASR includes systems such as HTK [48], SPHINX [49–50], JULIUS [51], KALDI [52], and BAVIECA [53] and general-purpose ASR APIs provided by, for instance, Google, Microsoft, and IBM. These toolkits and APIs have been employed in HRI applications to incorporate ASR capabilities to a robot on a plug-and-play fashion [54–58], i.e., a speech signal is input to the ASR to obtain a text transcription (see Figure 1) without taking into consideration operation conditions such as noise, relative movement between the speaker and the robot, microphones directivity and response, or user or robot context.

In Reference [54], a project that integrates smart home technology and a socially assistive robot to extend independent living for elderly people is described. A Nao robot plays the role of communication interface between the elderly, the smart home, and the external world. The robot can recognize simple answers from the user such as “yes” and “no” by using Sphinx 4.0 from Carnegie-Mellon University. Despite the fact that the Nao robot has a built-in microphone, its quality is too low for practical indoor applications, and a ceiling-mounted microphone was used to capture user speech. CMU Sphinx engine was also employed in Reference [57], as part of a voice control system for a robotic endoscope holder during minimally invasive surgery. In Reference [56], a general framework for multimodal human–robot communication is proposed. This framework allows users to interact with robots using speech and gestures. The Google Speech API was chosen because it offered speaker and vocabulary independency, which in turn could allow a natural speech interaction with no constraints. Google Speech API was also employed in Reference [58] to provide ASR capabilities to a robot that needed to understand the intentions of users without requiring specialized user training. It comprises a recognition model that combines language, gestures, and visual attributes. In Reference [59], four ASR engines were compared by making use of different grammars: Google Speech API, the Microsoft Speech APIM, Pocket Sphinx from CMU, and the NAO-embedded Nuance VoCon 4.7 engine. Experimental results showed that

the Google Speech API led to the highest accuracy. Smart speakers such as Amazon Echo have also been employed in HRI studies to control, for instance, robotic arms [60–62]. In References [63–64], the semantic model of a small set of vocal commands applicable to HRI is learnt by making use of non-negative matrix factorization. In Reference [65], multimodal categorization, word discovery, and a double articulation analysis were studied regarding their capability to enable a robot to obtain words and their embodied meanings from raw sensory–motor information, including visual information, haptic information, auditory information, and acoustic speech signals.

The integration of ASR technology on a black-box basis can lead to poor performance, because the chosen ASR system is not designed necessarily to comply with specific scenarios or tasks. In Reference [55], an evaluation with children aged from 4 to 10 years old playing versions of a language-based game hosted by an animated character is described. Speech recognition results using Sphinx3 on children utterance showed a poor performance, partially due to the mismatch between the children's voices and the adult acoustic model of the ASR engine. General purpose speech toolkits or APIs have been widely used as an easy solution to integrate ASR to some platforms. However, while those ASR engines provide good results in several scenarios, they may not provide an optimal solution to specific tasks, because they are not considered in the training procedure, or the technology simply does not compensate for unexpected distortions. As an example, in Reference [66], it was investigated whether the open-source speech recognizer Sphinx can be tuned to outperform Google cloud-based speech recognition API in a spoken dialog system task. By training a domain-specific language and making adjustments, Sphinx could outperform the Google Web Speech API (Google API) by 3.3%. It is worth emphasizing that commercial APIs may also allow to generate several language models per task. By doing so we no longer are treating those APIs as simple black boxes. However, in this context, we highlight that the language models need to be fully trained beforehand, and there is little room to generate language models dynamically or "on demand." Even if the language models could be generated "on the fly" or dynamically, a strategy is needed to decide which language model should be generated, and robust HRI may require that we handle many different scenarios that can hardly be modeled beforehand, which in turn welcomes specific research on language modelling.

### 2.2 Simulating ASR with WoZ Evaluations

One of the challenges in HRI interaction that may require an ad hoc solution instead of a multipurpose API, is the speech recognition with relative movements between the speaker and the robot. In scenarios where ASR is performed by moving robots, the corruption of speech produced by the additive noise of the robot's motors should be taken into consideration. Speech recognition experiments with moving robots in Reference [67] led the authors to recommend that the robot should pause its actions as soon as it realizes that it is being talked to, which in some applications is unacceptable. They also suggest that the only reliable speech recognition engine for HRI is another human being. Given the fragility of ASR technology that was unveiled in HRI environments, many researchers have adopted interaction mechanisms that do not rely on speech recognition technology and Wizard of Oz– (WoZ) based approaches have been chosen by several authors [68–75].

### 2.3 Evaluation of Optimal Physical Setup and Operating Conditions

There is an alternative strategy, which instead of making the ASR technology more suitable to target operating conditions or adopting WoZ schemes, attempts to find the optimal operating environment that maximizes the ASR accuracy. In Reference [59], the following variables were evaluated: different noise scenarios, different distances and angles of the speaker with respect to the microphones, and three types of microphones, i.e., desktop, studio, and the robot-mounted

microphone. According to the experimental results, the authors provide recommendations regarding how the speech-based HRI with children should be deployed so as to achieve a smoother interaction. Some of the recommendations are as follows, using additional input/output devices, even replacing verbal language input with a touchscreen and to place the user in an optimal location with respect to the microphones. Although these recommendations are based on evaluations with children, the authors suggest that they are applicable to HRI in general.

A speech recognition friendly artificial language (ROILA) was compared to English spoken language when talking to a Nao robot in Reference [67]. The experiment considered three microphone types (the ones built-in in the robot, a headset, and a desktop microphone), two conditions of head movement (static and moving) for the Nao robot, and two types of spoken languages (English and ROILA). The authors concluded that ROILA does not provide a significant improvement when compared to ordinary spoken English. However, the type of microphone and the robot head movement are critical for the ASR accuracy. If ideal operating conditions are not met, then one strategy is to try to cancel the corrupting environments. For instance, in Reference [76] and Reference [77] the external noise sources or ego-noise caused by motors and fans of the robot are removed with enhancement methods.

### 2.4 Beamforming

Another alternative to select a specific source or reduce the effect of noise is to use microphone arrays to direct the main lobe of this array toward the target speech source or user. This technique is known as beamforming and there are several methods to combine the channels of microphone arrays. Some beamforming techniques that are widely employed in the literature are the well-known delay-and-sum that uses destructive interferences to reduce the received acoustic power from those directions that are different from the target source angular position [78]; the minimum variance distortionless response (MVDR) that seeks to minimize the variance of the noise, constrained to the distortionless restriction of the beamformer output in the desired direction [79]; and the linear constraint minimum variance that generalizes the idea of MVDR by imposing multiple constraints [80]. However, conventional beamforming techniques have a limited capacity to reduce diffuse noise and reverberation [81]. This behavior is mainly due to the fact that to estimate delays accurately is a difficult task in reverberant environments [82]. Consequently, the applicability of beamforming methods is reduced in indoor scenarios where the successive reflections in walls can generate a diffuse or reverberant field.

## 3 ASR TESTBED FOR MOBILE HRI

In contrast to the ASR integration on a black-box basis as discussed above, in this article we propose to consider not only the acoustic signal but also the acoustic environment (see Figure 2). By acoustic environment we mean the acoustic channel, reverberation conditions and the additive noise caused by the robot movement and external sources. Additionally, the model presented in Figure 2 considers the states and contexts of the robot and user as inputs. Even though state and context may have broad meanings in HRI, they can be denoted briefly by very precise information from the ASR technology point of view. For example, robot state and context would denote all the information about human beings around the robot and current variables and operating conditions of the machine to generate a list of feasible or acceptable commands or information that could be input by the user. User state and context would designate, among others, the user's position, attitude, emotional conditions, and task completion status that can also predict user's command and info input to the robot. Consequently, the states and contexts of the robot and user can condition the adaptation of the ASR language model. However, the full accomplishment of this kind of integration is far beyond the scope of a single paper, and we focus here on the acoustic

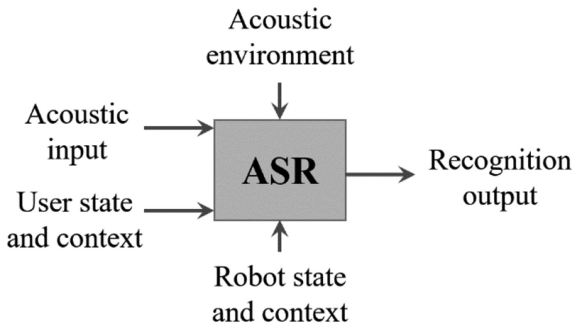


Fig. 2. A more complete ASR integration for HRI scenarios.

environment representation and modeling and on beamforming oracle source tracking. First, we trained our ASR engine with clean utterances combined with the acoustic-channel responses and noise that were estimated and recorded, respectively. Noise was generated by the robot itself and up to two noise sources. Second, beamforming technology was evaluated by assuming that the direction of speech target source was known by other means than audio analysis, e.g., image detection and classification technology. To carry out this study, we implemented a testbed that attempts to represent a generic acoustic environment of HRI, from the kinematic point of view, when a single user interacts with a mobile robot.

First, for instance, consider some real human social scenarios where robots could be very useful: a museum guide giving a tour, a student in a classroom asking the teacher a question, a rescue team helping a survivor, and a team of chefs working in a restaurant. All these situations have something in common: A person talks to somebody else who is busy accomplishing a task and is not looking to who is talking to him/her. Also, the two individuals may be moving one with respect to the other. Moreover, there could be sources of interferent noise.

As shown in Figure 2, the integration model employed here considers the information related to the acoustic environment as one of the inputs of the ASR engine. In this article, we represent the acoustic environment with the impulse responses that characterize the TVAC, and the additive noise generated by the robot movement and other sources. The main advantage of this methodology is the fact that it is much more efficient than recording the training database in all the possible operating conditions. To record the testing speech data in a real mobile robot scenario, to estimate the channel impulse responses and to record the additive noise, we implemented a testbed that employs a loudspeaker and human speakers as target speech sources, loudspeakers of interferent noise sources and a moving robot as a receiver.

A preliminary version of this testbed was described in Reference [83] where pilot experiments were reported. Because of the high relevance to the HRI community, a more complete version of this type of HRI scenario is proposed and described in the following sections (see Figure 3). Particularly, beamforming technology was evaluated by considering that the direction of speech target source was known by other means than audio analysis, e.g., image detection and classification technology. Beamforming spatial filtering was introduced to filter out the noise from the robot and external sources to increase the SNR with respect to the target speech source. Moreover, the additive noise, including the one generated by the robot and the external sources was recorded and included in the training procedure to represent more accurately the robot movement-conditions and the acoustic environment. Also, additional test sets were recorded by replacing the loudspeaker with human speakers in the same context.

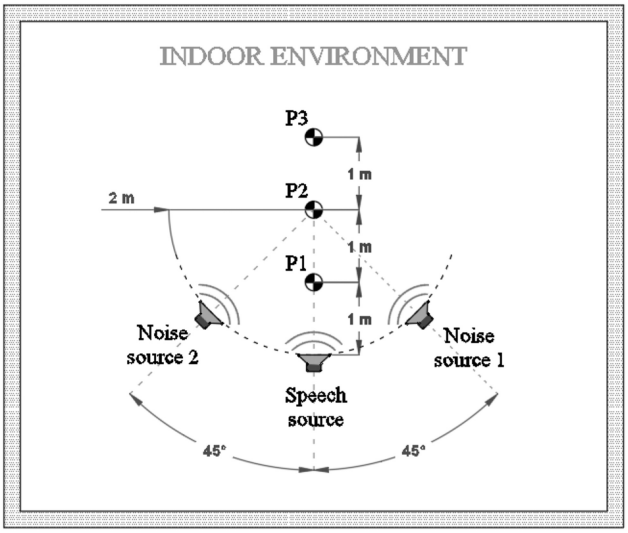


Fig. 3. Generic indoor environment floor plan where the HRI scenario for the testing Databases I, II, and III was implemented. The robot moved toward and away from the speech source (i.e., the loudspeaker or the human speakers) between positions P1 and P3 in dynamic conditions. The static conditions were recorded with the robot in positions P1 and P2. For the testing Database-II only the external noise source 1 was considered, while for the testing Database-III both external noise sources 1 and 2 were included.

### 3.1 Robotic Platform and TVAC

The experimental platform makes use of the Personal Robot 2 (PR2). Our PR2 robot is equipped with a Microsoft Xbox 360 Kinect sensor mounted on top of its head. The Kinect sensor has been widely used in the HRI community [84–88] and has recently been adopted in the recording of the Fifth CHiME Challenge database [89]. Plug-and-play devices such as Kinect represent a practical solution for a wide variety of HRI applications, because they allow image and audio processing, as it has RGB and depth cameras, as well as a linear array of four microphones channels. However, the methodology presented here is also applicable to other solutions based on sensors embedded in robots. We re-recorded 330 clean testing utterances of the Aurora-4 database with our HRI testbed located in an indoor environment, denominated Room 1, according to (Figure 3), including different specifications of the relative motion between the robot and the sources. The Aurora-4 clean test set was chosen for the generation of the test data to compare the results with those already reported by the speech community. A database representing a specific HRI application could be composed of a small lexicon or provide a low perplexity language model, which in turn would reduce the representativeness of the results. In contrast, using a generic database with a larger lexicon and a higher language model perplexity improves the representativeness and applicability of the proposed methods. Note that when the speech source and the robot are static one with respect to the other is a special case in relation to the more general situation. The recording was performed by the PR2 Microsoft Kinect sensor, which contains a four-microphone array. The recording procedure considered the relative movements of the robot microphones with respect to the speech sources by simultaneously applying translational movement to the robot body and angular rotation to the robot's head.

The robot moved toward and away from the speech source (i.e., the loudspeaker or the human speakers) between positions P1 and P3 in Figure 3. Three maximum robot displacement velocities were defined as follows: $Vmax_1 = 0.30$ m/s, $Vmax_2 = 0.45$ m/s, and $Vmax_3 = 0.60$ m/s. Those

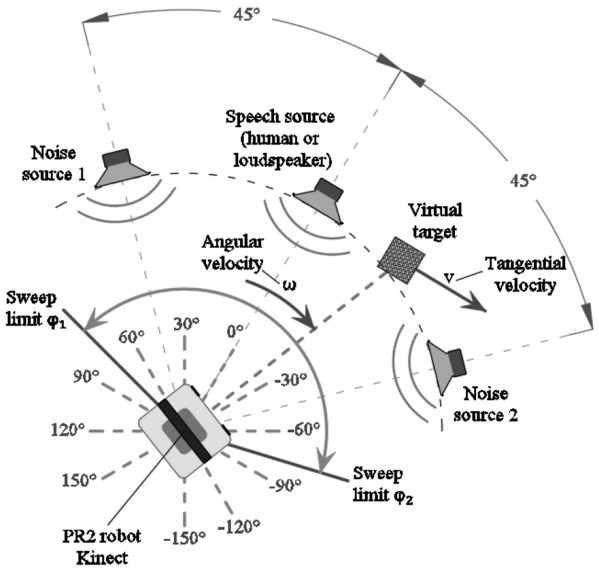


Fig. 4. Movement of the PR2 robot head during the testing databases recording. The head moves periodically from sweep limits $\varphi_1$ and $\varphi_2$ and back at angular velocities equal to $\omega$ rad/s. Recordings with static head are performed at 0° and 45°. The selected angular velocities for the robot head emulate the situation where the robot follows with the head a virtual target located two meters away and moving with tangential velocity of $v$ km/h. The speech sources can be a loudspeaker or a human speaker. In both cases the sources were located at 0° with respect to the robot front.

velocities were inspired by the discussions in Reference [90], where a robot approached to a seated person at 0.2 and 0.4 m/s. In those conditions, none of the human participants found these robot speeds were too fast. Then, the maximum velocities mentioned above were multiplied by an acceleration and deceleration function. Additionally, the recording of the test database was also performed with the robot in some static conditions with respect to the speech source at positions P1 and P2 in Figure 3, i.e., located at one and two meters away from the speech source, respectively.

The robot makes turns with the head as shown in Figure 4 for the displacement conditions described above. The sources were located at 0° (i.e., facing straight the robot). The three angular velocities $\omega$ for the robot head were made equal to 0.28, 0.42, and 0.56 rad/s. The chosen angular velocities correspond to the angular speed of the head rotation necessary for the robot to follow a virtual target with its head movement. The virtual target would be located two meters away from the robot and it is moving with tangential velocities of 2, 3, and 4 km/h, respectively, as shown in Figure 4. A fourth angular motion condition was 0 rad/s, fixing the robot's head at a given angle.

Considering sources of external noise as part of the problem is important from the real-world application point of view. These noise sources are part of a generic HRI scenario but can be considered uncorrelated with the interaction itself. To address this problem of speech recognition with external noise sources and TVAC, up to two speakers were positioned at 45° from the speech source at 2 m away from position P2 as shown in Figures 3 and 4.

### 3.2 HRI Scenario Testing Databases

Three testing databases were recorded using the generic HRI testbed shown in Figure 3 as described as follows: Database-I, Database-II, and Database-III. In Database-I several robot moving conditions were evaluated but no noise external noise source was employed. In contrast,

Table 1. Testing Databases

| | Condition ID | Displacement Vel. [m/s] | Angular Vel. [rad/s] | Head Angle | Ext. Noise Sources |
|---|---|---|---|---|---|
| **Database-I** | | 0 | 0 | 0° | 0 |
| | | | 0.28 | — | 0 |
| | | | 0.42 | – | 0 |
| | | | 0.56 | – | 0 |
| | | 0.3 | 0 | 0° | 0 |
| | | | 0.28 | – | 0 |
| | | | 0.42 | – | 0 |
| | | | 0.56 | – | 0 |
| | | 0.45 | 0 | 0° | 0 |
| | | | 0.28 | – | 0 |
| | | | 0.42 | – | 0 |
| | | | 0.56 | – | 0 |
| | | 0.6 | 0 | 0° | 0 |
| | | | 0.28 | – | 0 |
| | | | 0.42 | – | 0 |
| | | | 0.56 | – | 0 |
| **Database-II** | Static 1 | 0 | 0 | 0° | 1 |
| | Static 2 | 0 | 0 | 45° | 1 |
| | Dynamic 1 | 0 | 0.42 | – | 1 |
| | Dynamic 2 | 0.45 | 0.42 | – | 1 |
| **Database-III** | Static 1 | 0 | 0 | 0° | 2 |
| | Static 2 | 0 | 0 | 45° | 2 |
| | Dynamic 1 | 0 | 0.42 | – | 2 |
| | Dynamic 2 | 0.45 | 0.42 | – | 2 |

Database-II and Database-III made use of one and two external noise sources, respectively. These databases are described in detail in the following subsections and summarized in Table 1.

*3.2.1 Testing Database-I.* This database contains 16 conditions of robot movement with two speech sources: loudspeaker and human speaker. The two speech sources corresponded to a studio loudspeaker and four native American English speakers (two males and two females). The external noise sources in Figure 3 were not employed in this case.

The combination of four conditions for robot displacement defined in Section 3.1 (i.e., three translational movements between P1 and P3 with maximum velocities $Vmax_1$, $Vmax_2$, and $Vmax_3$ and a static position at P1) and four robot head angular movements defined in Section 3.1 (robot head angular velocities equal to 0, 0.28, 0.42, and 0.56 rad/s) produces 16 test database recording conditions. In this database, the robot head moves periodically from $\varphi_1 = -150^\circ$ to $\varphi_2 = 150^\circ$ and back in the dynamic conditions, as shown in Figure 4. Consequently, the total number of Aurora-4 clean testing utterances reproduced with the studio loudspeaker is equal to 330 utterances/robot-movement-conditions x 16 robot-movement-conditions = 5,280 utterances. However, each of the four native American English speakers pronounced ten sentences from the Aurora-4 corpus per robot-movement-conditions. Those sentences were the same for all the four speakers. As a result, the human speakers recorded 4 × 10 utterances/robot-movement-conditions x 16 robot-movement-conditions = 640 utterances. Figure 5 shows the experimental setup used to record this

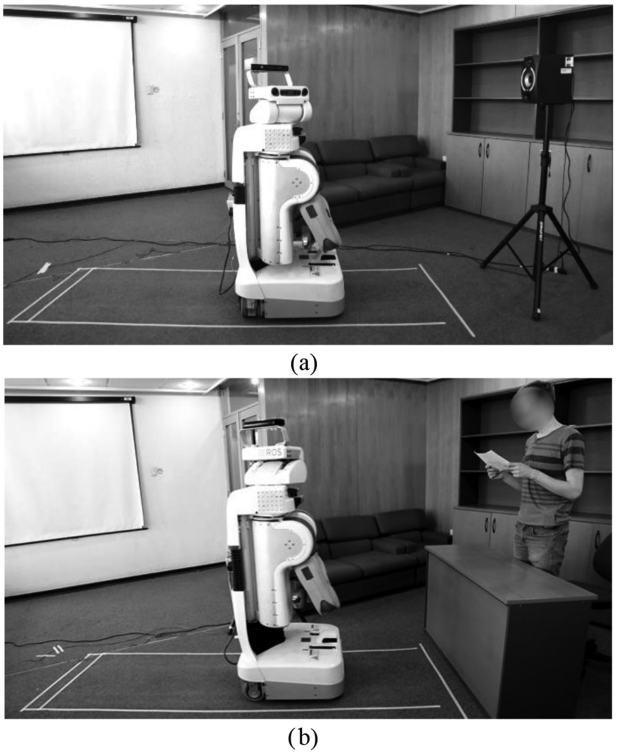


Fig. 5. PR2 robot equipped with a Microsoft Kinect that was used to record the testing Database-I: (a) The speech source corresponds to a studio loudspeaker that was employed to reproduce clean utterances from a database and (b) the speech source is a human speaker reading sentences from the same corpus.

database. The average number of words per utterances is equal to 16.2 words. The vocabulary size in the testing data is 1,270 words. It is important to mention that background noise was kept under control and measured before recording the test database at each robot movement condition. The equivalent sound pressure level over ten minutes was equal to 39 dBA. Instructions for requesting the playback modality of the testing Database-I are available at http://www.lptv.cl/en/hri-asr/. Further information about the testing database recording can be found in Reference [91].

*3.2.2 Testing Database-II.* This database represents a more extreme situation than the one described in Database-I (Section 3.2.1), because it considers one external noise source (noise source 1 in Figure 3). Two static and two dynamic sets were generated. These four movement conditions are summarized in Table 1. The first static condition, Static 1, was recorded by fixing the robot's head at 0° (i.e., oriented toward the speech target source). The second static condition, Static 2, corresponds with the head oriented toward the external noise source 1 located at 45° from the speech target (Figure 3).

The first and second dynamic conditions, Dynamic 1 and Dynamic 2, were recorded with the robot head moving periodically between $\varphi_1 = -50°$ to $\varphi_2 = 50°$ (sweep limits in Figure 4) at angular velocity $\omega$ equal to 0.42 rad/s while the translational velocity was 0 and 0.45 m/s, respectively, as detailed in Table 1.

Consequently, the number of Aurora-4 clean testing utterances re-recorded in this database is equal to 330 utterances/robot-movement-conditions x 4 robot-movement-conditions = 1,320

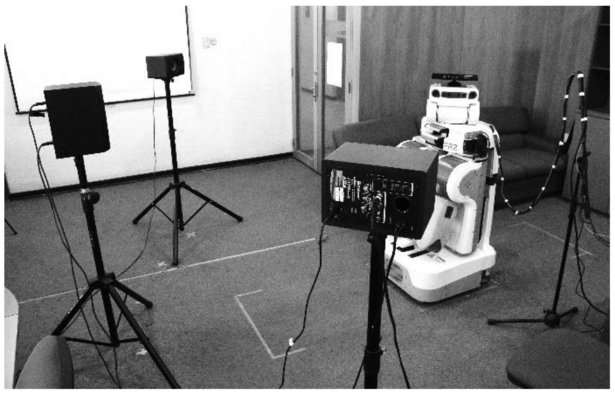

Fig. 6. The experimental setup used to record testing Database-II and Database-III, i.e., considering one and two external noise sources calibrated at an SNR equal to 5 dB, respectively.

utterances. An SNR equal to 5 dB was adopted, where the estimation of the signal energy was made based on clean data (330 utterances) reproduced at two meters from the array of microphones, i.e., with the robot at position P2, and the robot's head at 0° (i.e., oriented toward the speech target source).

The noise energy estimation was performed by considering 1 minute of additive noise reproduced with one of the additional loudspeakers located at 45° (noise source 1 in Figure 3) and by adjusting the received power until the desired SNR was achieved. As a reference, the restaurant noise was employed. For the estimation of the signal and noise energies, the recordings were made with the robot servers turned on, i.e., there is a noise produced by the servers' fans as well as the cooling system at the robot base. Figure 6 shows the experimental setup employed to record this database. The Database-II recording made use of the library available in the software developer's kit (SDK) for Microsoft Kinect where the four channels and the beamformed signal were recorded. In addition, the robot saved the azimuthal angle of the robot's head in conditions Dynamic 1 and Dynamic 2.

*3.2.3 Testing Database-III.* This database represents an even more challenging situation than testing Database-II with two external noise sources (see Figure 3). The configuration used here is also shown in Figure 4 with more details. The noise sources 1 and 2 were located two meters from position P2 and at 45° from speech target source as shown in Figures 3 and 4. This database made use of the same static and dynamic conditions in Table 1 that were employed for Database-II. Therefore, the number of Aurora-4 clean testing utterances re-recorded in this database is the same as in Database-II, i.e., 1320 utterances. The SDK libraries for Microsoft Kinect were also used here, and the four individual channels and the beamformed signal were recorded. The azimuthal angle of the head was saved in the robot's servers at conditions Dynamic 1 and Dynamic 2. The SNR was also made equal to 5 dB. The experimental set up shown in Figure 6 was also employed to record this database.

### 3.3 Characterization of the Acoustic Environment for ASR Training

The generic HRI test bed in Figure 3 needs to be characterized and modelled to include this information in the speech recognition training procedure. As explained above, the main motivation is to soften the requirements of the training database to achieve an accuracy similar or superior to the one that would be obtained by a large database representing all the testing conditions. As a

matter of fact, as a result, we employed a small clean database to integrate our acoustic model and environmental noise into the training procedure. The proposed method is not just data augmentation that increases the training set size by incorporating, for instance, simple speed perturbations on utterances to generate copies of the training data. Actually, the number of training utterances was not modified. Also, the proposed scheme is not multicondition training either and to record training speech in several operating conditions is not necessary. We believe that the procedure presented here is interesting for the robotic community to achieve high recognition accuracy in complex real HRI scenarios with limited training data.

*3.3.1 TVAC.* If the relative position between the user and the robot varies during the interaction, then the acoustic channel will be time dependent. Consequently, the signal received at each microphone, $y$, can be described by

$$y[i] = \sum_{p=0}^{\infty} h_i[p]x[i-p], \tag{6}$$

where $x$ denotes the original clean signal, $h_i$ is the time-dependent impulsive response representing the acoustic channel, and $i$ is the discrete time index. The TVAC problem cannot be solved with conventional compensation techniques, because the channel cannot be assumed constant. This challenging problem has hardly been addressed in the literature. In this article, we propose a solution to this problem applicable to moving dynamics HRI scenarios.

The TVAC in a generic HRI scenario (Figure 3) can be modeled using a set of samples of the acoustic channel impulse responses. In this article, 33 four-channel impulse responses (IRs) were computed with the robot placed at P1, P2, and P3 (Figure 3), and for each robot position the head was oriented at 11 different angles with respect to the source. The head angle was varied from $-150°$ to $150°$ in steps of $30°$. The $0°$ angle corresponds to the Microsoft Kinect microphones oriented toward the sources in Figure 4. The impulse responses were estimated using the Farina's sine sweep method [92]. An exponential sine sweep signal was generated from 64 Hz to 8 kHz and reproduced with a studio loudspeaker. The sweep audio was recorded with the four channel Microsoft Kinect sensor. An impulse response was estimated for each channel by convolving the corresponding recorded signal with the time reversal of the original exponential sine sweep.

*3.3.2 Robot Noise.* To incorporate additional information about the acoustic environment in our HRI scenario, different robot noise levels were recorded by the Kinect microphone array in the 16 robot movement conditions defined for Database-I (Section 3.2.1). The robot noise is generated by its internal fans and electrical motors operating at different translational and angular velocities. Finally, the four Kinect channels were summed to obtain a single channel signal. This noise was included in the ASR training based on the acoustic environment modelling described in Section 4.

*3.3.3 External Noise Sources.* As mentioned above, considering external noise sources is important to represent HRI in real applications. In addition, according to the model proposed in Section 3 and shown in Figure 2, all the information related to the acoustic environment that is available can be very useful to improve the performance of the ASR engine. To fully characterize the possible test conditions in Databases II and III, the two external sources of noise were recorded (noise sources 1 and 2 in Figure 3). The type of noise reproduced by these additional loudspeakers was restaurant noise. This noise was employed in the environment-based training described in Section 4.2.

## 4 ENVIRONMENT-BASED ASR TRAINING

As mentioned above, instead of integrating the ASR technology on a black-box basis, we propose to consider the HRI operation conditions such as the acoustic environment (see Figure 2). In this

article we tackled the environment representation and modelling in the ASR training procedure to soften the requirements on the training database and achieve high recognition accuracy in complex HRI environments with limited training data. By "environment", we mean the acoustic channel, reverberation conditions, and the additive noise caused by the robot movement and external sources. Accordingly, we generated two **environment-based training** data: **EbT-I** and **EbT-II**. In EbT-I, the training data included the impulse responses that model the microphone responses, the reverberation, and the robot's noise. In EbT-II, we also incorporated the additive noise from external sources.

The experiments were performed with a DNN-HMM ASR using Kaldi, which is a state-of-the-art and competitive ASR technology. Kaldi is an open source speech recognition toolkit that is popular among the speech research community. It was first described in Reference [52] and is regularly updated with implementations of new techniques and recipes for speech recognition systems.[1] The Kaldi project is hosted in Github.[2] To build a DNN-HMM system with Kaldi, first a GMM-HMM is trained with the EbT training data, using the tri2b Kaldi recipe for the Aurora-4 database. In this recipe, a monophone system is trained; then, the alignments from that system are employed to generate an initial triphone system; finally, the triphone alignments are employed to train the final triphone system. Also, Mel-frequency cepstral coefficients parametrization of speech, linear discriminant analysis, and maximum likelihood linear transforms are part of the recipe. Once the GMM-HMM system is trained, the GMM is replaced with a DNN. The DNN is composed of seven hidden layers and 2,048 units per layer each, and the input considers a context window of 11 frames. The number of units of the output DNN layer is equal to the number of Gaussians in the corresponding GMM-HMM system. The DNN was trained using Mel filter bank features. The reference for the DNN training is the alignment obtained with the clean version of the whole training data and the GMM-HMM trained with the same clean data. This leads to a better reference for the DNN than using the noisy or corrupted speech data directly [93–94]. The DNN is trained firstly using the Cross-Entropy criterion. Then, the final system is obtained by re-training the DNN with the sMBR discriminative training [95]. It is important to mention that the ASR system was trained using mean and variance normalization (MVN) applied on a per-utterance basis. This type of normalization could partly solve the channel problem when the channel is a microphone or telephone line or considered time independent [96]. However, MVN-like approaches have a very limited effect with reverberated signals and simply do not solve the TVAC problem. Our final ASR systems are referred as EbT-I or EbT-II, as described above, depending on the training data. For comparison reasons, we also trained a DNN with the clean database without any information regarding the HRI testbed scenario. Furthermore, the statistical significance analysis was performed using the NIST matched-pair sentence-segment word error test [97]. For decoding, the standard 5K lexicon and trigram language model from Wall Street Journal (WSJ) were used [98]. As a result, the language model is tuned to the task, i.e., it is task dependent. The required files and scripts to generate the EbT-I and EbT-II training data, and the detailed Kaldi recipe to train the DNN-HMM-based ASR system employed here are available at http://www.lptv.cl/en/hri asr/.

### 4.1 EbT-I

The speech recognition experiments reported here made use of Aurora-4 database [99], which in turn was generated with the 5,000-word closed-loop vocabulary task based on the DARPA Wall Street Journal (WSJ0) Corpus. The Aurora-4 clean training set corresponds to the WSJ0 SI-84 database [100] of the Nov'92 ARPA CSR [101]. This database was used to generate the training and

[1] At the time of writing, the Kaldi source code was updated in July 2020.
[2] https://github.com/kaldi-asr/kaldi.

Table 2. Training Datasets used to Train the EbT-I and EbT-II ASR Systems Described in Sections 4.1 and 4.2, Respectively

| | Training data | |
|---|---|---|
| | **EbT-I** | **EbT-II** |
| **25% of data** | Convolved with 1 IR | Convolved with 1 IR |
| **75% of data** | Convolved with 32 IRs obtained at static positions, and noise added at SNR between 10 dB and 20 dB. | Convolved with 32 IRs obtained at static positions, and noise added at SNR between 10 dB and 20 dB. |
| **Noise type** | Robot noise | Robot noise and restaurant noise at SNR between −5 dB and 5 dB. |

"IR" indicates impulse response.

testing sets to allow comparison with results published by the speech community. To generate the EbT-I set, 25% of the clean training utterances of the Aurora-4 database, which consists of 7,138 utterances (i.e., 15.2 hours) from 83 native English speakers and contains only data recorded with a high-quality microphone (i.e., Sennheiser HMD-414), was convolved with the IRs, corresponding to the four Kinect channels, estimated when the robot-source distance was equal to 1 m and the angle between the robot head and the speech source was 0°. Then, the four convolution results were summed to obtain a single channel signal. The remaining 75% of the clean training set was convolved with the remaining 32 four-channel IRs by employing the same procedure described above, in such a way that the IRs were evenly distributed across the signals. The recorded robot's noise was added to this 75% of utterances using the Filtering and Noise Adding Tool (FaNT) [102] at SNR between 10 and 20 dB. It is worth highlighting that these training data are completely different from the testing databases described above, i.e., different speakers and different utterances.

### 4.2 EbT-II

A composed additive noise was generated by adding the robot's noise to the external source noise at a random SNR between −5 dB and 5 dB. Similarly to EbT-I, EbT-II considers 25% of the clean training utterances of the Aurora-4 database convolved with the IRs estimated when the robot-source distance was equal to 1 m and the angle between the robot head and the source was 0°, and the remaining 75% of the clean training set was convolved with the remaining 32 four-channel IRs. Then, the composed additive noise was added to this 75% using the FaNT tool at SNR between 10 and 20 dB. Table 2 shows the comparison between the EbT-I and EbT-II training sets.

## 5 BEAMFORMING IN TVAC WITH EXTERNAL NOISE SOURCES

Applying beamforming techniques to filter out spatially the external source noise can lead to an increase of SNR in the HRI scenarios represented by Database-II and Database-III (see Sections 3.2.2 and 3.2.3). In this article, we explored and evaluated four beamforming strategies that made use of the four-microphone array available in the testing Database-II and Database-III: Sum Without Delay (SWD), Kinect SDK, BeamformIt, and Beamforming with Oracle Source Tracking (BOST). These beamforming systems or schemes are described as follows:

*Sum Without Delay.* The SWD strategy corresponds to the simple direct sum of the four channels delivered by the Microsoft Kinect without considering the delays regarding the reference microphone (i.e., $\tau_1$, $\tau_2$, and $\tau_3$ in Figure 7). All the channels were weighted uniformly.

*Kinect SDK.* This library is part of the Microsoft Kinect SDK in its version 1.8 and is available on the Microsoft website [103]. From a practical point of view, this library operates as a black

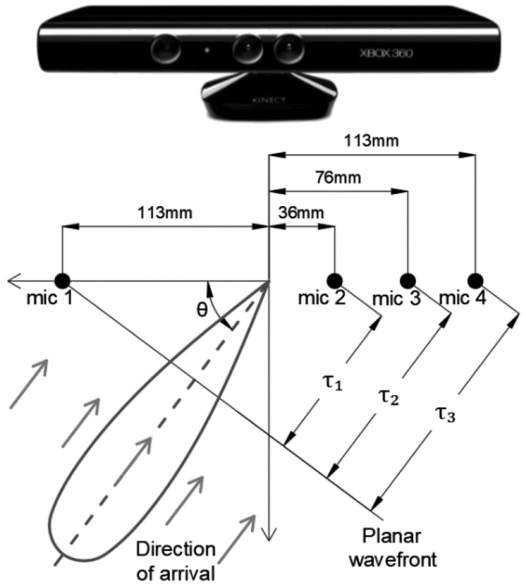


Fig. 7. Geometry of the four-channel microphone array in the Microsoft Kinect. This device supports beamforming and can employed in HRI scenarios.

box that delivers the signal resulting from beamforming. Nevertheless, one can have access to the estimated angle of the detected speech target source.

*BeamformIt.* This toolbox performs a weighted delay and sum where the delays and weights are estimated internally by the toolkit [104]. BeamformIt also operates on a black-box basis with the four Kinect audio channels and delivers a single beamformed signal.

*Beamforming with Oracle Source Tracking*. We proposed the BOST strategy to implement the ordinary delay-and-sum beamforming [78] provided that the channel phase shifts are known. These phase shifts are estimated from the time-of-arrival, which in turn is determined from the angle between the Kinect at the top of the robot's head and the speech target source by assuming a planar wave front as shown in Figure 7. This angle was estimated with the information provided by the robot moving condition and this scheme can be considered as an oracle source tracking. It is worth mentioning that the orientation of the robot head with respect to the target speech source position, which represents the user, is included in what can be called user and robot state and context information. Therefore, BOST represents a particular case of the ASR integration strategy to HRI scenarios discussed in Section 3 (see Figure 2).

## 6 RESULTS AND DISCUSSION

In this section, we present the results obtained with the environment-based trained ASR systems EbT-I (Section 4.1) and EbT-II (Section 4.2). The EbT-I system was tested with Database-I and the results were compared with publicly available APIs. In the case of EbT-II, the ASR system was tested with Database-II and Database-III in combination with beamforming methods. Moreover, additional experiments with Amazon Echo Dot are presented in the appendix to show the difficulty of the kind of tasks addressed here.

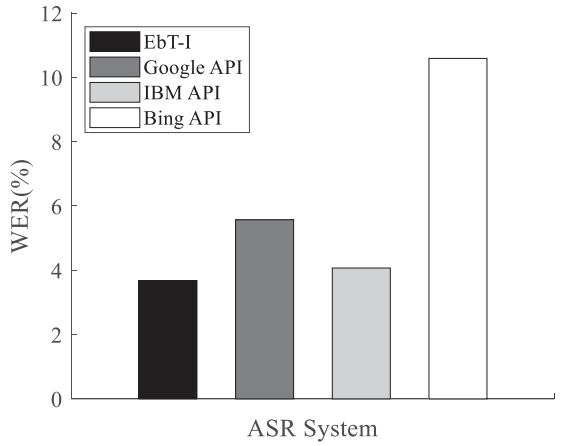


Fig. 8. WERs obtained with the EbT-I system and the publicly available ASR APIs. The testing data corresponds to the original clean test set from Aurora-4 database (330 utterances).

### 6.1 EbT-I System and the APIs

The average WER obtained with the 5,280 utterances recorded in our HRI scenario with the studio loudspeaker (Section 3.2.1), was equal to 65.0% using our ASR system trained with clean data. When only the IRs were incorporated in the training procedure, the average WER was substantially reduced to 31.4%. Moreover, our EbT-I system (i.e., that includes both IRs and robot noise) provided a much lower WER: 11.5%. This important increase of the ASR accuracy strongly supports our proposed approach to model the acoustic environment of an HRI scenario with channel impulse responses and robot additive noise. Observe that this WER was achieved with only 15.2 hours of training data. This result was corroborated by making use of our testing dataset that was recorded with the four native American English speakers that led to an average WER of 73.5% and 21.3% with clean training and EbT-I, respectively. These WERs are higher than those obtained with the playback testing data. This must be due to the fact that the human speakers pronounced the utterances with a lower volume resulting in a lower signal to noise ratio. The average SNRs were equal to 11 and 18 dB for human speakers and loudspeaker data, respectively.

For comparison reasons, we also ran ASR experiments with three publicly available APIs by using the "Speech Recognition" Python library (Version 3.7) [105]: the Google Web Speech API (Google API), the IBM Speech to Text API (IBM API), and the Bing Voice Recognition API (Bing API). Figure 8 shows the WER obtained with our EbT-I system and the three API mentioned above with the 330 clean utterances from Aurora-4. The result obtained with the EbT-I system is very competitive when compared with those published elsewhere with the Aurora-4 database by making use of multicondition training [106, 107]. As can be seen in Figure 8, the EbT-I system provided the lowest WER that is 10% lower than the second best, i.e., IBM API. Although this relative improvement seems important, the differences between the two best systems were not significant. This result suggests that adopting a better tuned language model, as done in our EbT-I ASR system, does not provide a clear advantage over a flatter or more general-purpose language model.

In our HRI test sets, we observed that in challenging scenarios the APIs evaluated here delivered empty strings as the result of the ASR queries. Given this situation, the WERs were estimated with the non-empty returned text strings. Figure 9 shows the ASR results obtained with our EbT-I system, Google API, and IBM API, in all the robot motion conditions, with the playback loudspeaker testing Database-I (Section 3.2.1). In the case of Bing API, the empty string rate (ESR) increased

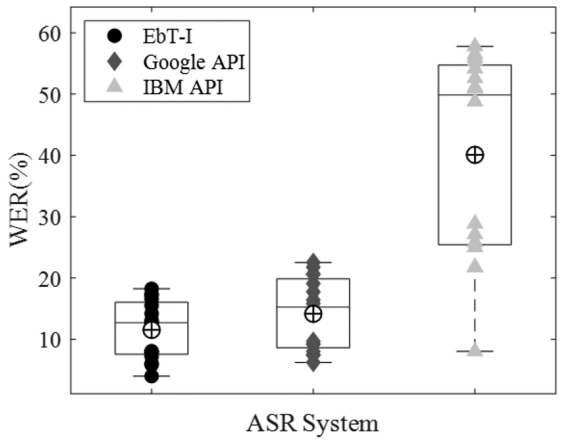


Fig. 9. Results with *playback loudspeaker* testing Database-I (Section 3.2.1). These plots show the WERs obtained with the EbT-I system, Google API and IBM API in all the robot movement conditions. The average WERs were 11.5%, 14.2%, and 40.1% with EbT-I, Google API, and IBM API, respectively. In the boxplots the plus sign in circle, "⊕", indicates the average WER in each case.

considerably and prevented us from showing a representative WER. All the ASR results with the APIs shown in Figure 9 were carried out between September 6 and 12, 2017. As expected, the lowest and highest WERs were achieved with the static condition (i.e., translation and angular velocities equal to zero) and with the highest displacement and rotational velocities, respectively. Also, for each robot movement condition in the testing Database-I the lowest WER was achieved with the EbT-I system.

As can be seen in Figure 9, the lowest WER corresponds to our EbT-I system. The average WER achieved with the EbT-I system is 19% lower than the second best (statistically significant with $p < 0.001$), i.e., Google API. According to Figure 9, the EbT-I system and Google API provided the lowest WER dispersion. Also, the observed average empty string rates or ESRs were equal to 0%, 0.3%, and 6.5% with EbT-I, Google API, and IBM API, respectively. If we include the empty strings in the computation of the error rates, then the average WERs increased to 14.2% and 41.9% with Google API and IBM API, respectively. With EbT-I the WER was not modified, because ESR is equal to zero in this case. It is worth highlighting that these results are due to the proposed acoustic environment modelling and representation rather than to the language model. Observe that, as mentioned above, the average WER with the ASR trained with clean speech is 65.0%.

For validation purposes, Figure 10 summarizes the WERs obtained with EbT-I, Google API, and IBM API, with the native American English speakers testing Database-I (Section 3.2.1). According to Figure 10, the lowest value and dispersion for WER also corresponds to our EbT-I system. The average WER achieved with system EbT-I is 34% lower than the second best, i.e., Google API. The average ESRs with the human speaker testing dataset are equal to 0%, 5.8% and 5.6%. If we include the empty strings in the computation of the error rates, then the average WERs increased to 34.9% and 56.8% with Google API and IBM API, respectively. The results with the ASR APIs using the native American English speaker as speech source in the testing Database-I were obtained between September 25 and October 5, 2017. The results shown in Figure 9 and Figure 10 are strong empirical evidence that support the proposed scheme, which in turn can lead to competitive ASR accuracies when compared with publicly APIs by making use of limited training data.

By comparing Figure 8 with Figure 9, we can observe that the lowest WER is achieved with our EbT-I system. However, the EbT-I system also provides the highest relative increase in average

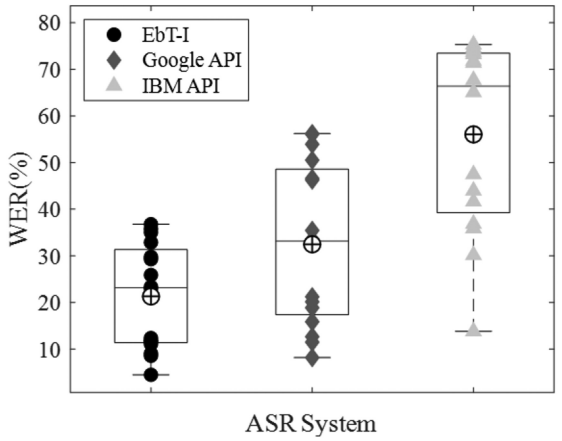


Fig. 10. Results with *native American English speakers* testing Database-I (Section 3.2.1). These plots show the WERs obtained with the EbT-I system, Google API, and IBM API in all the robot movement conditions. The average WERs were 21.3%, 32.5%, and 56.0% with EbT-I, Google API, and IBM API, respectively. In the boxplots the plus sign in circle, "⊕", indicates the average WER in each case.

Table 3. Characterization of Rooms where the EbT-I Training System Strategy Was Applied and Evaluated

| Room | Description | Volume ($m^3$) | RT (s) |
|---|---|---|---|
| 1 | Meeting room | 104 | 0.50 |
| 2 | Classroom | 244 | 0.64 |
| 3 | Classroom | 208 | 1.05 |

RT denotes the measured reverberation time.

WER, i.e., 214%, from the clean testing data to the playback loudspeaker testing Database-I (Section 3.2.1) in the HRI scenario. In contrast, Google API, for instance, shows a relative increase in average WER equal to 154%. This result can be explained according to Reference [31], Reference [32], and Reference [33], where it is said that the ASR engines that support the APIs evaluated here could have been trained with at least thousands of hours of speech covering a wide diversity of acoustic conditions. In contrast, our EbT-I system was trained with only 15.2 hours of clean speech utterances that were convolved with channel impulse responses and had noise added (Section 4.1). The proposed procedure is applicable to any HRI environment, being only necessary the capture of the robot noise and the estimation of the acoustic impulse responses to get a new environment-based ASR system. This procedure requires just a couple of days and a few hours of training data. At this point it is worth highlighting that the adequate use of user and robot states and contexts can reduce the language model perplexity, and lead to further improvements in recognition accuracy.

Further two versions of the EbT-I system were generated by making use of two additional rooms, i.e., Room 2 and Room 3, to illustrate the effectiveness and easy replicability of the proposed methodology. Table 3 shows the characterization of the rooms employed here. Three test conditions representative of the 16 conditions available in Database-I were also recorded in Room 2 and Room 3. These three movement conditions were C.1, v = 0 m/s and w = 0 rad/s; C.2, v = 0 m/s and w = 0.42 rad/s; and C.3. v = 0.45 m/s and w = 0.42 rad/s. Table 4 shows the WERs obtained with the EbT-I systems generated in each one of the three rooms characterized in Table 3 separately. Aditionally, a fourth EbT-I system was trained with the acoustic channels from all

Table 4. WER Obtained with the EbT-I Systems Generated for Room 1, Room 2, and Room 3

| Training | | Clean | | | EbT-Itrained for Room 1 | EbT-Itrained for Room 2 | EbT-Itrained forRoom 3 | EbT-I trained with the acoustic information from Rooms 1, 2 and 3. | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Testing Room** | | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
| **Testing** | C.1 | 8.6% | 7.6% | 17.3% | 4.0% | 4.1% | 5.7% | 4.2% | 4.1% | 4.5% |
| **Condition** | C.2 | 10.7% | 49.4% | 75.1% | 7.9% | 7.8% | 14.5% | 8.7% | 7.9% | 19.5% |
| | C.3 | 83.5% | 78.2% | 90.8% | 14.2% | 19.0% | 32.6% | 16.7% | 19.2% | 39.1% |

the rooms. Table 4 also shows the WERs obtained with the ASR trained with clean speech as a reference.

As expected, the results in Table 4 show that the WERs increase as the movement conditions become more dynamic, i.e., C.2 and C.3, independently of the room and training condition. As can be seen, EbT-I systems provided significant reductions in WER within the corresponding room when compared with the systems trained with clean signal. For comparison purposes, the WERs obtained for condition C.3 with the Google API in Room 1, Room 2, and Room 3 were 34.4%, 60.1%, and 18.6% higher than the WERs obtained by the corresponding EbT-I systems, respectively.

The results with Google API on Room 2 and Room 3 data were obtained on September 13, 2019. In addition, a single EbT-I ASR system was trained using the acoustic channel impulse responses from all the rooms. In average, this room independent EbT-I system provided a WER just 7.6% higher than the room dependent EbT-I systems. Finally, it is worth highlighting that despite the fact that the adaptation from one acoustic scenario to another is beyond the scope of the current manuscript, the estimation of reverberant impulse responses is a problem that is currently being addressed by making use of audio and visual information.

### 6.2 EbT-II and Beamforming Methods

The EbT-II-based trained ASR was tested with Database-II and Database-III in Room 1. Beamforming systems were employed to filter out spatially the additive noise from external sources and to increase the SNR. The error in the direction of arrival (DOA) estimation provided by the Kinect SDK, see Figure 7), was obtained as the absolute value of the difference between the Kinect DOA and the oracle one. This error was averaged across all the testing utterances and the experimental conditions and was equal to 32° and 27° with one and two noise sources, respectively.

*6.2.1 Results with a Single Noise Source.* In this section, we analyze the results obtained in an HRI scenario considering one external noise source. The results obtained with testing Database-II are shown in Figure 11. For the first static condition contained in this database, i.e., with the robot head oriented toward the speech source, the lowest WERs were achieved by SWD and BOST beamforming strategies. No significant differences were found between the SWD and BOST. The WERs obtained by these strategies were 31% and 45% lower (statistically significant with $p < 0.001$) than the BeamformIt and Kinect SDK toolkits, respectively. In the case of the second static condition, i.e., with the robot head oriented toward the noise source, the lowest WER was achieved by the BOST technique, which is 55% and 54% lower (statistically significant with $p < 0.001$) than those obtained with the BeamformIt and Kinect SDK toolkits, respectively. In the first dynamic condition, i.e., only considering the rotational movement of the robot's head, the best result was reached by the BOST technique and is 44% and 53% lower (statistically significant with $p < 0.001$) than the WERs with BeamformIt and Kinect SDK toolkits, respectively.

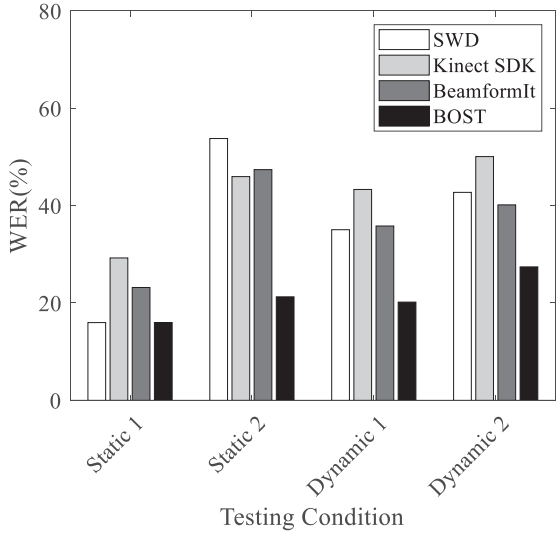


Fig. 11. Results obtained with testing Database-II using the direct sum of the microphone array channels (i.e., SWD), the Kinect SDK library, the BeamformIt toolkit, and the BOST strategy.

Finally, for the second dynamic condition, i.e., translational robot movement and robot's head rotation, the best result was also achieved by the BOST scheme that provided a WER that was 32% and 45% lower (statistically significant with $p < 0.001$) than those obtained with the BeamformIt and Kinect SDK toolkits, respectively.

In all the conditions, except the first one (i.e., static robot with the head looking at 0° with respect to the speech target source), the BOST strategy outperformed SWD, BeamformIt and the Kinect SDK. It is worth highlighting that the Kinect SDK and BeamformIt toolkits achieved improvements when compared to SWD only in one and two out of four conditions, respectively. This must be due to the fact that the direction of the speech target source was not accurately estimated in this scenario with reverberation and external additive noise by making use of acoustic information only.

*6.2.2 Results with Two Noise Sources.* This scenario is even more complex than the previous one, i.e., Database-II, due to the successive reflections of both sources of external noise in the walls of the room. The results obtained with testing Database-III are shown in Figure 12. In the first static condition, i.e., with the robot head oriented toward the speech source, the lowest WER was achieved by SWD, which is 12% lower (statistically significant with $p < 0.001$) than the BeamformIt toolkit. Although the WER achieved by the SWD strategy is lower than the one obtained with the Kinect SDK library, no significant differences were found between these two results. In the second static condition, i.e., with the robot head oriented toward one of the noise sources, the best result was achieved by the BOST technique, which delivers a WER that is 24% and 33% lower (statistically significant with $p < 0.001$) than the BeamformIt and Kinect SDK toolkits, respectively. In the first dynamic condition, i.e., considering only the rotational movement of the robot's head, the best result was also achieved by the BOST scheme that is 31% and 42% lower (statistically significant with $p < 0.001$) than the WERs with BeamformIt and Kinect SDK toolkits, respectively. Finally, in the second dynamic condition, i.e., by simultaneously applying translational movement to the robot body and angular rotation to the robot's head, the lowest WER was also achieved by the BOST strategy, which is 22% and 35% lower (statistically significant with $p < 0.001$) than the BeamformIt and Kinect SDK toolkits, respectively. Only in the first static condition, BOST did

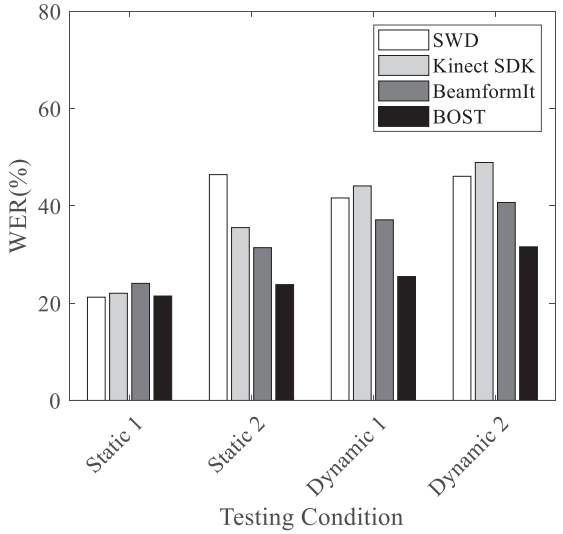


Fig. 12. Results with testing Database-III using the direct sum of the microphone array channels (i.e., SWD), the Kinect SDK library, the BeamformIt toolkit, and the BOST strategy.

not provide the highest accuracy, but the difference with the lowest WER, i.e., with SWD, was not significant. Similarly to the tests performed considering only one source of external noise, Kinect SDK and BeamformIt led to improvements with respect to SWD s only in one and three of four conditions, respectively.

This result corroborates our findings with one external noise source and suggests that to estimate the direction of the speech target source by making use of acoustic information only in the presence of reverberation and external additive noise is not a trivial problem.

*6.2.3 Beamforming with Oracle Source Tracking.* Even though the beamforming toolkits in Section 5 (i.e., Kinect SDK and BeamformIt) achieved a significant improvement when compared with SWD in some conditions, the improvements obtained with the proposed BOST scheme suggest that to estimate the direction of the target speech source by employing acoustic information only in the presence of reverberation and external noise sources is a difficult problem. In this context, localizing the target source by using other means such as image processing is an interesting solution and the accuracy improvements provided by BOST can be considered as oracle results with source tracking in complex indoor HRI scenarios. As mentioned above, the orientation of the robot head with respect to the target speech source position, which represents the user, can be considered user and robot state and context information. Consequently, BOST corresponds to a particular case of the ASR integration strategy to HRI scenarios discussed in Section 3 (see Figure 2).

## 7 CONCLUSIONS

In this article, we point out the issue of ASR degradation in a dynamic situation where the robot can be moving or rotating the microphone. This is proven by systematic evaluation on various conditions. We showed that to incorporate our time-varying acoustic channel model combined with environmental noise in the ASR training procedure can help improve the results, outperforming black-box APIs. Then, we evaluate the use of other information that in this case is the oracle beamforming direction to point out the potential benefit of building a more integrated ASR solution with the robot. This article is focused on the acoustic environment representation and

modeling by training a DNN-HMM model-based automatic speech recognition engine by means of the combination of clean utterances with the acoustic channel responses and noise that were estimated and recorded, respectively, with an HRI testbed built with a PR2 robot. The proposed procedure addresses the TVAC problem and is much more effective and efficient than recording a training database in all the possible acoustic environments, given an HRI scenario. Consequently, we present a solution to the problem of acoustic environment representation and modelling in the ASR training procedure to soften the requirements on the training database to achieve high recognition accuracy in complex HRI environments with limited training data. This is an interesting outcome for the robotics community due to the potential application in HRI of Kinect or equivalent solutions. Different speech recognition testing conditions were generated by recording two types of speech sources, i.e., a loudspeaker and human speakers, and by including external source of noise using the PR2 robot, which has a Microsoft Kinect sensor mounted on top, while performing head rotations and movements toward and away from the fixed sources. This testbed models the generic problem of HRI in mobile robotics, from the kinematic point of view, when a user is interacting with a robot, and the resulting automatic speech recognition accuracy outperformed publicly available speech recognition APIs. The average WER achieved by our system is at least 19% and 34% lower than the evaluated APIs with the loudspeaker and human modalities, respectively, with a limited amount of training data. Observe that the proposed method is not just data augmentation that attempts only to increase the training set size by introducing, for instance, simple speed perturbations on utterances to generate copies of the training data. This is not the case of the current article where the acoustic environment was modelled and incorporated in the training procedure. Moreover, the number of training utterances was not modified. The proposed scheme is not multicondition training either and to record training speech in several operating condition is not necessary. It is worth highlighting that unlike a system implemented in the cloud, such as the APIs tested here, an off-line system provides greater security regarding the confidentiality of the analyzed data and greater stability of the system, since it does not depend on transmission networks.

Using the BOST strategy, lower WERs were achieved than using publicly available beamforming toolkits, i.e., BeamformIt and Kinect SDK. The accuracy improvements obtained with the BOST strategy can be considered as oracle results with target source tracking in complex indoor HRI scenarios. For the most challenging condition, i.e., translational robot movement and robot's head rotation, the best result employing one external noise source was achieved by the BOST strategy that provided a WER that is 32% and 45% lower than those obtained with the BeamformIt and Kinect SDK toolkits, respectively. With two external noise sources, the lowest WER was also achieved by the BOST strategy, which is 22% and 35% lower than the BeamformIt and Kinect SDK toolkits, respectively. These results strongly suggest that to use different sources of information such as those provided by cameras and other sensors can provide complementary information about the interaction environment and lead to better results than those obtained with beamforming toolkits in HRI situations.

At this point, it is interesting to highlight that the results presented were achieved by integrating knowledge to the ASR technology, not only raw data. The approach we propose can lead to standalone ASR systems that do not need to send data to the cloud and can provide competitive or higher accuracy than publicly available APIs with limited training data in real environments. We believe that both issues are significant improvements to HRI technology. An extra factor in HRI scenarios is that the user speech may be stressed in noisy conditions, i.e., Lombard effect. This problem and the incorporation of more complex aspects related to the states and contexts of users and robots, from the speech recognition point of view, are proposed for future research.

## APPENDIX

There has been a growing market for smart speakers in the past few years. Products such as Amazon Echo Dot and Google Home are becoming very popular and have already been used by some authors in the HRI community, e.g., References [62, 108, 109]. The scenarios in which our methodology is tested are common in mobile robotics but are very challenging. To show the complexity of the problem addressed here, we carried some experiments with Amazon Echo Dot in one of our testbeds. An additional testing database was generated for this purpose. The first 100 clean testing utterances of the Aurora-4 database were re-recorded simultaneously by using the Kinect and the Echo Dot (third generation) in two different testing conditions with the robot placed in position P2 according to Figure 3. The Echo Dot speaker was mounted on top of the PR2's Kinect sensor, which in turn is mounted on top of the robot´s head (see Figure A1). The first testing condition dataset, *Set 1*, was generated with the robot head fixed and oriented toward the speech source. The second condition dataset, *Set 2*, was recorded with the robot head rotating at 0.42 (rad/s). The speech source corresponded to the same studio loudspeaker employed in Section 3.2. The robot

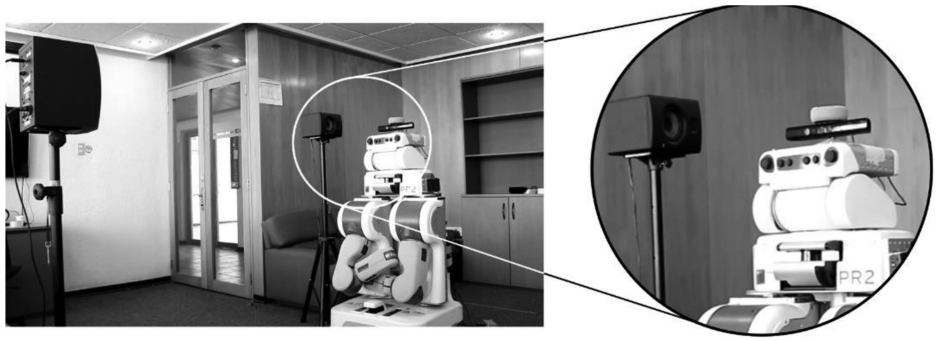


Fig. A1. Experimental setup used for simultaneous recording with Amazon Echo Dot and Microsoft Kinect.

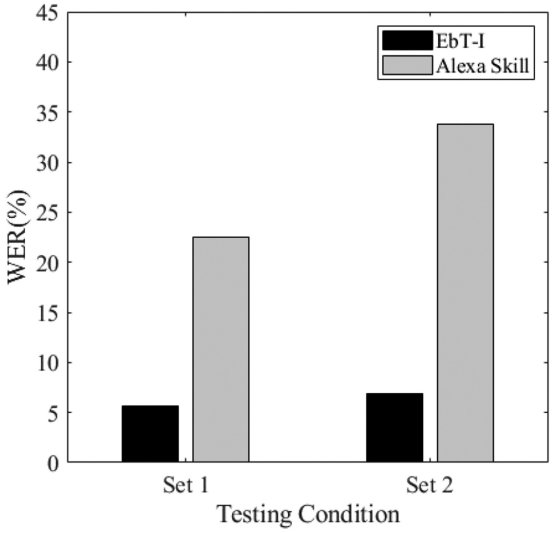


Fig. A2. WERs obtained with Alexa Skill and with the EbT-I system for the testing conditions in *Set 1* and *Set 2*.

engine was on in both conditions but the external noise sources in Figure 3 were not used in this case. The resulting SNRs were equal to 18 and 15 dB with *Set 1* and *Set 2*, respectively.

A "skill" was generated in the Amazon development environment.[3] This "skill" started capturing the audio after receiving the "wake-up" word to send it to the Amazon cloud service. Then the result was stored in a S3 bucket (Amazon storage service) for the subsequent word error rate computation. The experiments were carried out on July 13 and 14, 2020. It is important to mention that to achieve a better response of the "skill" in the recordings, the "wake-up" word was reproduced with another studio speaker near the robot's head (see Figure A1). The results obtained are shown in Figure A2. The efficiency of the "wake-up" word was 70% and 75% for *Set 1* and *Set 2*, respectively. Therefore, the WERs shown in Figure A2 were computed for Echo Dot and Kinect datasets with those utterances that were decoded by our Alexa "skill" to make the results comparable.

As can be seen in Figure A2, our EbT-I system provided WERs 75% and 80% lower than those obtained with Alexa "skill"/Echo Dot with *Set 1* and *Set 2*, respectively. Moreover, the WER increase from *Set 1* (robot´s head is static) to *Set 2* (robot´s head rotates) is 50% and 20% with Alexa "skill"/Echo Dot and our EbT-I system, respectively. Basically, these results confirm the difficulty of the problem addressed here and the pertinence of the proposed methodology.

## REFERENCES


[1] Michael A. Goodrich and Alan C. Schultz. 2007. Human-robot interaction: A survey. *Found. Trends Hum.-Comput. Interact.* 1, 3 (2007), 203–275.

[2] Luis S. Lopes and Antonio Teixeira. 2000. Human-robot interaction through spoken language dialogue. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 528–534.

[3] Guy Hoffman and Keinan Vanunu. 2013. Effects of robotic companionship on music enjoyment and agent perception. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*. 317–324.

[4] Chao-Yu Lin, Kai-Tai Song, Yi-Wen Chen, Shuo-Cheng Chien, Sin-Horng Chen, Chen-Yu Chiang, Jyh-Her Yang, Yi-Chiao Wu and Tzu-Jui Liu. 2012. User identification design by fusion of face recognition and speaker recognition. In *Proceedings of the 12th International Conference on Control, Automation and Systems*. 1480–1485.

[5] Kuanhao Zheng, Dylan F. Glas, Takayuki Kanda, Hiroshi Ishiguro, and Norihiro Hagita. 2013. Designing and implementing a human-robot team for social interactions. *IEEE Trans. Syst. Man Cybernet. Syst.* 13, 4 (2013), 843–859.

[6] Yutaka Kondo, Kentaro Takemura, Jun Takamatsu, and Tsukasa Ogasawara. 2013. A gesture-centric android system for multi-party human-robot interaction. *J. Hum.-Robot Interact.* 2, 1 (2013), 133–151.

[7] Donglin Wang, Henry Leung, Ajeesh P. Kurian, Hye-Jin Kim, and Hosub Yoon. 2010. A deconvolutive neural network for speech classification with applications to home service robot. *IEEE Trans. Instrum. Meas.* 59, 12 (2010), 3237–3243.

[8] Erica L. Meszaros, Meghan Chandarana, Anna Trujillo, and B. Danette Allen. 2017. Compensating for limitations in speech-based natural language processing with multimodal interfaces in UAV operation. In *Proceedings of the International Conference on Applied Human Factors and Ergonomics*. 183–194.

[9] Seungho Han, Jungpyo Hong, Sangbae Jeong, and Minsoo Hahn. 2010. Robust GSC-based speech enhancement for human machine interface. *IEEE Trans. Consum. Electr.* 56, 2 (2010), 965–970.

[10] Maria Staudte and Matthew W. Crocker. 2011. Investigating joint attention mechanisms through spoken human-robot interaction. *Cognition* 120, 2 (2011), 268–291.

[11] Henrique A. Polido. 2014. DARPA Robotics Challenge. Major Qualifying Project. Worcester Polytechnic Institute (WPI), Worcester, MA.

[12] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, and Eiichi Osawa. 1997. Robocup: The robot world cup initiative. In *Proceedings of the 1st International Conference on Autonomous Agents*. 340–347.

[13] Liangpei Zhang, Lefei Zhang, and Bo Du. 2016. Deep Learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geosci. Remote Sens. Mag.* 4, 2 (2016), 22–40.

[14] Scott E. Umbaugh. 2011. *Digital Image Processing and Analysis: Human and Computer Vision Applications with CVIPtools* (2nd. ed.). CRC Press.

[15] Wilhelm Burger and Mark J. Burge. 2016. *Digital Image Processing: An Algorithmic Introduction Using Java*. Springer.

[16] Junichi Nakamura. 2016. *Image Sensors and Signal Processing for Digital Still Cameras*. CRC Press.

[17] Steve Young. 2008. HMMs and related speech recognition technologies. In *Springer Handbook of Speech Processing*. Jacob Benesty, M. Mohan Sondhi, and Yiteng Huang (Eds.). Springer. 539–558.


[3] https://developer.amazon.com.

[18] Xuedong D. Huang, Yasuo Ariki, and Mervyn A. Jack. 1990. *Hidden Markov Models for Speech Recognition*. Edinburgh University Press.

[19] Raquel Justo and M. Inés Torres. 2015. Integration of complex language models in ASR and LU systems. *Pattern Anal. Appl.* 18, 3 (2015), 493–505.

[20] Stanley F. Chen, Douglas Beeferman, and Ronald Rosenfeld. 1998. Evaluation metrics for language models. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*. 275–280.

[21] Anirudh Raju, Behnam Hedayatnia, Linda Liu, Ankur Gandhe, Chandra Khatri, Angeliki Metallinou, Anu Venkatesh, and Ariya Rastrow. 2018. Contextual language model adaptation for conversational agents. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH'18)*. 3333–3337.

[22] David Rybach, Michael Riley, and Johan Schalkwyk. 2017. On lattice generation for large vocabulary speech recognition. In *Proceedings of the 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU'17)*, 228–235.

[23] Cyril Allauzen and Michael Riley. 2015. Rapid vocabulary addition to context-dependent decoder graphs. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH'15)*. 2112–2116.

[24] Mohamed Chetouani, Bruno Gas, and Jean Luc Zarader. 2002. Discriminative training for neural predictive coding applied to speech features extraction. In *Proceedings of the 2002 International Joint Conference on Neural Networks*. 852–857.

[25] Namrata Dave. 2013. Feature extraction methods LPC, PLP and MFCC in speech recognition. *Int. J. Adv. Res. Eng. Technol.* 1, 6 (2013), 1–5.

[26] Sadaoki Furui. 1986. Speaker independent isolated word recognition using dynamic features of speech spectrum. *IEEE Trans. Acoust. Speech Sign. Process.* 34, 1 (1986), 52–59.

[27] Lalit R. Bahl. 1980. Language-model/acoustic channel balance mechanism. *IBM Techn. Disclos. Bull.* 23, 7B (1980), 3464–3465.

[28] Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Sign. Process. Mag.* 29, 6 (2012), 82–97.

[29] John J. Godfrey and Edward Holliman. 1997. Switchboard-1 Release 2. LDC97S62. Linguistic Data Consortium, Philadelphia, PA.

[30] Jens Schröder, Jörn Anemüller, and Stefan Goetze. 2016. Performance comparison of GMM, HMM and DNN based approaches for acoustic event detection within Task 3 of the DCASE 2016 challenge. In *Proceedings of the Workshop on Detection and Classification of Acoustic Scenes and Events*. 80–84.

[31] Bo Li, Tara N. Sainath, Arun Narayanan, Joe Caroselli, Michiel Bacchiani, Ananya Misra, Izhak Shafran, Hasim Sak, Golan Pundak, Kean Chin, Khe Chai Sim, Ron J. Weiss, Kevin W. Wilson, Ehsan Variani, Chanwoo Kim, Olivier Siohan, Mitchel Weintraub, Erik McDermott, Richard Rose, and Matt Shannon. 2017. Acoustic Modeling for google home. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH'17)*. 399–403.

[32] George Saon, Hong-Kwang J. Kuo, Steven Rennie, and Michael Picheny. 2015. The IBM 2015 english conversational telephone speech recognition system. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH'15)*. 3140–3144.

[33] Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Michael L. Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig. 2017. The microsoft 2016 conversational speech recognition system. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*. 5255–5259.

[34] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* 9, 8 (1997), 1735–1780.

[35] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. 2014. Convolutional neural networks for speech recognition. *IEEE/ACM Trans. Audio Speech Lang. Process.* 22, 10 (2014), 1533–1545.

[36] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*. 6645–6649.

[37] Zhiyuan Tang, Dong Wang, and Zhiyong Zhang. 2016. Recurrent neural network training with dark knowledge transfer. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*. 5900–5904.

[38] Jinyu Li, Abdelrahman Mohamed, Geoffrey Zweig, and Yifan Gong. 2015. LSTM time and frequency recurrence for automatic speech recognition. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*. 187–191.

[39] Tara N. Sainath and Bo. Li. 2016. Modeling time-frequency patterns with LSTM vs. Convolutional Architectures for LVCSR Tasks. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH'16)*. 813–817.

[40] Yuzong Liu and Katrin Kirchhoff. 2016. Novel front-end features based on neural graph embeddings for DNNHMM and LSTM-CTC acoustic modeling. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH'16)*. 793–797.

[41] Dong Yu, Wayne Xiong, Jasha Droppo, Andreas Stolcke, Guoli Ye, Jinyu Li, and Geoffrey Zweig. 2016. Deep convolutional neural networks with layer-wise context expansion and attention. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH'16)*. 17–21.

[42] Yanmin Qian, Mengxiao Bi, Tian Tan, and Kai Yu. 2016. Very deep convolutional neural networks for noise robust speech recognition. *IEEE/ACM Trans. Audio Speech Lang Process.* 24, 12 (2016), 2263–2276.

[43] Vikramjit Mitra and Horacio Franco. 2016. Coping with unseen data conditions: Investigating neural net architectures, robust features, and information fusion for robust speech recognition. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH'16)*. 3783–3787.

[44] Michael L. Seltzer, Dong Yu, and Yongqiang Wang. 2013. An investigation of deep neural networks for noise robust speech recognition. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP'13)*, 7398–7402.

[45] Jiri Malek and Jindrich Zdansky. 2019. On practical aspects of multi-condition training based on augmentation for reverberation-/noise-robust speech recognition. In *Text, Speech, and Dialogue,* Lecture Notes in Computer Science, Vol. 11697. Springer, Cham.

[46] Md Jahangir Alam, Vishwa Gupta, Patrick Kenny, and Pierre Dumouchel. 2015. Speech recognition in reverberant and noisy environments employing multiple feature extractors and i-vector speaker adaptation. *EURASIP J. Adv. Sign. Process.* 1, 50 (2015), 1–13.

[47] Chao Weng, Dong Yu, Michael L. Seltzer, and Jasha Droppo. 2014, Single-channel mixed speech recognition using deep neural networks. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing.* 5632–5636.

[48] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, and others. 2006. *The HTK Book*. Cambridge University Engineering Department.

[49] Kai-Fu Lee, Hsiao-Wuen Hon, and Raj Reddy. 1990. An overview of the SPHINX speech recognition system. *IEEE Trans. Acoust. Speech Sign. Process.* 38, 1 (1990), 35–45.

[50] Willie Walker, Paul Lamere, Philip Kwok, Bhiksha Raj, Rita Singh, Evandro Gouvea, Peter Wolf, and Joe Woelfel. 2004. Sphinx-4: A flexible open source framework for speech recognition. Sun Microsystems, Inc., SMLI TR-2004–139. Sun Microsystems, Inc. Menlo Park, CA.

[51] Akinobu Lee, Tatsuya Kawahara, and Kiyohiro Shikano. 2001. JULIUS - an open source real-time large vocabulary recognition engine. In *Proceeding of the Annual Conference of the International Speech Communication Association (INTERSPEECH'01)*. 1691–1694.

[52] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, and others. 2011. The kaldi speech recognition toolkit. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*.

[53] Daniel Bolaños. 2012. The bavieca open-source speech recognition toolkit. In *Proceedings of the IEEE Spoken Language Technology Workshop (SLT Workshop'12)*. 354–359.

[54] David O. Johnson, Raymond H. Cuijpers, James F. Juola, Elena Torta, Mikhail Simonov, Antonella Frisiello, Marco Bazzani, Wenjie Yan, Cornelius Weber, Stefan Wermter, Nils Meins, Johannes Oberzaucher, Paul Panek, Georg Edelmayer, Peter Mayer and Christian Beck. 2014. Socially assistive robots: A comprehensive approach to extending independent living. *Int. J. Soc. Robot.* 6, 2 (2014), 195–211.

[55] Jill Fain Lehman. 2014. Robo fashion world: a multimodal corpus of multi-child human-computer interaction. In *Proceedings of the 2014 Workshop on Understanding and Modeling Multiparty, Multimodal Interactions*. 15–20.

[56] Francesco Cutugno, Alberto Finzi, Michelangelo Fiore, Enrico Leone and Silvia Rossi. 2013. Interacting with robots via speech and gestures, an integrated architecture. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH'13)*. 3727–3731.

[57] Kateryna Zinchenko, Chien-Yu Wu, and Kai-Tai Song. 2017. A study on speech recognition control for a surgical robot. *IEEE Trans. Industr. Inf.* 13, 2 (2017), 607–615.

[58] Cynthia Matuszek, Liefeng Bo, Luke Zettlemoyer, and Dieter Fox. 2014. Learning from unscripted deictic gesture and language for human-robot interactions. In *Proceedings of the 28th National Conference on Artificial Intelligence.* 2556–2563.

[59] James Kennedy, Séverin Lemaignan, Caroline Montassier, Pauline Lavalade, Bahar Irfan, Fotios Papadopoulos, Emmanuel Senft, and Tony Belpaeme. 2017. Child speech recognition in human-robot interaction: evaluations and recommendations. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*. 82–90.

[60] Rishi Kapadia, Sam Staszak, Lisa Jian, and Ken Goldberg. 2017. EchoBot: Facilitating data collection for robot learning with the Amazon echo. In *Proceedings of the IEEE Conference on Automation Science and Engineering (CASE'17)*. 159–165.

[61] Michael Fischer, Samir Menon, and Oussama Khatib. 2016. From bot to bot: Using a chat bot to synthesize robot motion. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'16)*.
[62] Mohit Shridhar and David Hsu. 2018. Interactive visual grounding of referring expressions for human-robot interaction. In *Proceedings of Robotics: Science and Systems*.
[63] Vincent Renkens, Steven Janssens, Bart Ons, Jort F. Gemmeke, and others. 2014. Acquisition of ordinal words using weakly supervised NMF. In *Proceedings of the IEEE Spoken Language Technology Workshop (SLT'14)*. 30–35.
[64] Vincent Renkens and Hugo Van Hamme. 2015. Mutually exclusive grounding for weakly supervised non-negative matrix factorisation. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH'15)*.
[65] Tadahiro Taniguchi, Takayuki Nagai, Tomoaki Nakamura, Naoto Iwahashi, Tetsuya Ogata, and Hideki Asoh. 2016. Symbol emergence in robotics: A survey. *Adv Robot.* 30, 11–12 (2016), 706–728.
[66] Patrick Lange and David Suendermann-Oeft. 2014. Tuning sphinx to outperform google's speech recognition API. In *Proceedings of the Conference on Electronic Speech Signal Processing*. 1–10.
[67] Omar Mubin, Joshua Henderson, and Christoph Bartneck. 2014. You just do not understand me! speech recognition in human robot interaction. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication*. 637–642.
[68] Matthew Marge, Claire Bonial, Brendan Byrne, Taylor Cassidy, A. William Evans, Susan G. Hill, and Clare Voss. 2017. Applying the wizard-of-oz technique to multimodal human-robot dialogue. arXiv:1703.03714. Retrieved from https://arxiv.org/abs/1703.03714.
[69] Pedro Sequeira, Patrícia Alves-Oliveira, Tiago Ribeiro, Eugenio Di Tullio, Sofia Petisca, Francisco S. Melo, Ginevra Castellano, and Ana Paiva. 2016. Discovering social interaction strategies for robots from restricted-perception wizard-of-oz studies. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*. 197–204.
[70] Kristyn Hensby, Janet Wiles, Marie Boden, Scott Heath, Mark Nielsen, Paul Pounds, Joshua Riddell, Kristopher Rogers, Nikodem Rybak, Virginia Slaughter, Michael Smith, Jonathon Taufatofua, Peter Worthy, and Jason Weigel. 2016. Hand in hand: Tools and techniques for understanding children's touch with a social robot. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*. 437–438.
[71] Guy Hoffman. 2016. OpenWoZ: A runtime-configurable wizard-of-oz framework for human-robot interaction. In *Proceedings of the AAAI Spring Symposium Series*.
[72] Nikolas Martelaro. 2016. Wizard-of-oz interfaces as a step towards autonomous HRI. In *Proceedings of the AAAI Spring Symposium Series*.
[73] Shokoofeh Pourmehr, Jack Thomas, and Richard Vaughan. 2016. What untrained people do when asked "make the robot come to you." In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*. 495–496.
[74] Emmanuel Senft, Paul Baxter, James Kennedy, Séverin Lemaignan, and Tony Belpaeme. 2016. Providing a robot with learning abilities improves its perception by users. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*. 513–514.
[75] Jacqueline Kory Westlund and Cynthia Breazeal. 2016. Transparency, teleoperation, and children's understanding of social robots. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*. 625–626.
[76] Heinrich Löllmann, Alastair Moore, Patrick Naylor, Boaz Rafaely, Radu Horaud, Alexandre Mazel, and Walter Kellermann. 2017. Microphone array signal processing for robot audition. In *Proceedings of the Hands-free Speech Communications and Microphone Arrays*. 51–55.
[77] Antoine Deleforge and Walter Kellermann. 2015. Phase-optimized K-SVD for signal extraction from underdetermined multichannel sparse mixtures. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*. 355–359.
[78] Maurizio Omologo, Marco Matassoni, and Piergiorgio Svaizer. 2001. Speech recognition with microphone arrays. In *Microphone Arrays, Signal Processing Techniques and Applications*. Michael Brandstein and Darren Ward (Eds.). Springer-Verlag, 331–353.
[79] Joerg Bitzer and K. Uwe Simmer. 2001. Superdirective microphone arrays. In *Microphone Arrays, Signal Processing Techniques and Applications*. Michael Brandstein and Darren Ward (Eds.). Springer-Verlag, 19–38.
[80] Zhibao Li and Ka-Fai Cedric You. 2016. Beamformer configuration design in reverberant environments. *Eng. Appl. Artif. Intell.* 47 (2016), 81–87.
[81] K. Uwe Simmer, Joerg Bitzer, and Claude Marro. 2001. Post-filtering techniques. In *Microphone Arrays, Signal Processing Techniques and Applications*. Michael Brandstein and Darren Ward (Eds.). Springer-Verlag, 39–60.
[82] Jae Choi, Jeunghun Kim, and Nam Soo Kim. 2017. Robust time-delay estimation for acoustic indoor localization in reverberant environments. *IEEE Sign. Process. Lett.* 24, 2 (2017), 226–230.
[83] José Novoa, Jorge Wuth, Juan Pablo Escudero, Josué Fredes, Rodrigo Mahu, and Néstor Becerra Yoma. 2018. DNNHMM based automatic speech recognition for HRI scenarios. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*. 150–159.

[84] Jr. Ding and Jia-Yi Shi. 2017. Kinect microphone array-based speech and speaker recognition for the exhibition control of humanoid robots. *Comput. Electr. Eng.* 62 (2017), 719–729.
[85] V. Ramu Reddy, Parijat Deshpande, and Ranjan Dasgupta. 2015. Robotics audition using kinect. In *Proceeding of the International Conference on Automation, Robotics and Applications (ICARA'15)*.
[86] Riccardo Levorato and Pagello Enrico. 2014. Probabilistic 2D acoustic source localization using direction of arrivals in robot sensor networks. In *Proceedings of the International Conference on Simulation, Modeling, and Programming for Autonomous Robots*. 474–485.
[87] Parijat Deshpande, V. Ramu Reddy, Arindam Saha, Karthikeyan Vaiapury, Keshaw Dewangan, and Ranjan Dasgupta. 2015. A next generation mobile robot with multi-mode sense of 3D perception. In *Proceedings of International Conference on Advanced Robotics (ICAR'15)*.
[88] Shuopeng Wang, Peng Yang and Hao Sun. 2016. Design and implementation of auditory system for mobile robot based on kinect sensor. In *Proceedings of World Congress on Intelligent Control and Automation (WCICA'16)*.
[89] Jon Barker, Shinji Watanabe, Emmanuel Vincent, and Jan Trmal. 2018. The fifth 'CHiME' speech separation and recognition challenge: Dataset, task and baselines. arXiv:1803.10609. Retrieved from https://arxiv.org/abs/1803.10609.
[90] Kerstin Dautenhahn, Michael Walters, Sarah Woods, Kheng Lee Koay, Chrystopher L. Nehaniv, A. Sisbot, Rachid Alami, and Thierry Siméon. 2006. How may I serve you?: A robot companion approaching a seated person in a helping context. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*. 172–179.
[91] José Novoa, Juan Pablo Escudero, Josué Fredes, Jorge Wuth, Rodrigo Mahu, and Néstor Becerra Yoma. 2017. Multichannel robot speech recognition database: MChRSR. arXiv:1801.00061. https://arxiv.org/abs/1801.00061.
[92] Angelo Farina. 2000. Simultaneous measurement of impulse response and distortion with a swept-sine technique. In *Proceedings of the Audio Engineering Society Convention* 108. 1–23.
[93] Sunit Sivasankaran, Emmanuel Vincent, and Irina Illina. 2017. A combined evaluation of established and new approaches for speech recognition in varied reverberation conditions. *Comput. Speech Lang.* 46, Suppl. C (2017), 444–460.
[94] Payton Lin, Dau-Cheng Lyu, Fei Chen, Syu-Siang Wang, and Yu Tsao. 2017. Multi-style learning with denoising autoencoders for acoustic modeling in the internet of things (IoT). *Comput. Speech Lang.* 46, Suppl. C (2017), 481–495.
[95] Karel Veselý, Arnab Ghoshal, Lukáš Burget, and Daniel Povey. 2013. Sequence-discriminative training of deep neural networks. In *Proceeding of the Annual Conference of the International Speech Communication Association (INTERSPEECH'13)*. 2345–2349.
[96] Herman Kamper and Thomas Niesler. 2009. Characterisation and simulation of telephone channels using the TIMIT and NTIMIT databases. In *Proceedings of the Symposium of the Pattern Recognition Association of South Africa (PRASA'09)*, 47–52.
[97] David S. Pallett, William M. Fisher, and Jonathan G. Fiscus. 1990. Tools for the analysis of benchmark speech recognition tests. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*. 97–100.
[98] Jean-Luc Gauvain, Lori Lamel, and Martine Adda-Decker. 1995. Developments in continuous speech dictation using the ARPA WSJ task. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*. 65–68.
[99] Guenter Hirsch. 2002. Experimental framework for the performance evaluation of speech recognition front-ends on a large vocabulary task, Version 2.0, AU/417/02. ETSI STQ Aurora DSR Working Group.
[100] Douglas B. Paul and Janet M. Baker. 1992. The design for the wall street journal-based CSR corpus. In *Proceedings of the Workshop on Speech and Natural Language, Harriman*, 357–362.
[101] Philip C. Woodland, Julian J. Odell, Valtcho Valtchev, and Steve J. Young. 1994. Large vocabulary continuous speech recognition using HTK. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP'94)*, vol. II, II/125-II/128.
[102] Guenter Hirsch. 2005. FaNT Filtering and Noise Adding Tool. *Software*. Retrieved from http://dnt.kr.hs-niederrhein.de/.
[103] Microsoft. 2013. Kinect for Windows Software Development kit v1.8. Retrieved from https://www.microsoft.com/en-us/download/details.aspx?id=40278.
[104] Xavier Anguera, Chuck Wooters, and Javier Hernando. 2007. Acoustic beamforming for speaker diarization of meetings. *IEEE Trans. Audio Speech Lang. Process.* 15, 7 (2007), 2011–2023.
[105] Anthony Zhang. 2017. Speech Recognition (v3.7). Software. Retrieved from https://github.com/Uberi/speech_recognition.
[106] Gabriel Synnaeve. 2020. WER Are We?. Retrieved July 21, 2020 from https://github.com/syhw/wer_are_we.
[107] Josué Fredes, José Novoa, Simon King, Richard M. Stern, and Nestor Becerra Yoma. 2017. Locally normalized filter banks applied to deep neural-network-based robust speech recognition. *IEEE Sign. Process. Lett.* 24, 4 (2017), 377–381.

[108] Rishi Kapadia, Sam Staszak, Lisa Jian, and Ken Goldberg. 2017. Echobot: Facilitating data collection for robot learning with the amazon echo. In *Proceedings of the 2017 13th IEEE Conference on Automation Science and Engineering (CASE'17)*, 159–165.

[109] Michael Fischer, Samir Menon, and Oussama Khatib. 2016. From bot to bot: Using a chat bot to synthesize robot motion. In *Proceedings of the 2016 AAAI Fall Symposium Series*.

Received December 2018; revised June 2020; accepted November 2020