Robotica (2015) volume 33, pp. 1–18. © Cambridge University Press 2014
doi:10.1017/S0263574714000010

Command-based voice teleoperation of a mobile robot via a human-robot interface

Alberto Poncela†* and Leticia Gallardo-Estrella‡

†*Departamento Tecnología Electrónica, ETSI Telecomunicación, Universidad de Málaga, Campus de Teatinos, 29071 Málaga, Spain*
‡*Department of Radiology, Radboud University Nijmegen Medical Centre, Geert Grooteplein 10, 6525 GA Nijmegen, The Netherlands*

(Accepted December 22, 2013. First published online: January 28, 2014)

SUMMARY
Verbal communication is the most natural way of human–robot interaction. Such an interaction is usually achieved by means of a human-robot interface (HRI). In this paper, a HRI is presented to teleoperate a robotic platform via the user's voice. Hence, a speech recognition system is necessary. In this work, a user-dependent acoustic model for Spanish speakers has been developed to teleoperate a robot with a set of commands. Experimental results have been successful, both in terms of a high recognition rate and the navigation of the robot under the control of the user's voice.

KEYWORDS: Teleoperation; Human-robot interface; Voice interface; Speech recognition; Acoustic model.

1. Introduction

Autonomy can be defined as the ability to make independent choices.[1] According to Sheridan,[2] there are three levels of autonomy a robot might have, depending on how much autonomy the machine has: (i) fully autonomous; (ii) shared control; and (iii) fully controlled. Among these approaches, the robot is considered to be fully autonomous when it can perform a given task in an unstructured environment without continuous human guidance. In a shared control scheme, the robot and an operator take decisions on how the robot should behave. In the latter case, a fully controlled system, an operator must directly control every action of the robot. This level of autonomy in usually referred to as teleoperation, which needs remote interaction between the human operator and the robot. As teleoperation implies both the human and the robot to be physically separated, a communication between the agent and the human is required, which is commonly achieved with a human-robot interface (HRI).

A HRI needs a set of control inputs to let human communicate with robot, such as a mouse,[3] a keyboard,[4] or a joystick.[5] More sophisticated devices have also been used in HRI to teleoperate a robot, such gloves[6] or muscular activity sensors.[7] Nevertheless, verbal communication is the primary and most natural way of human communication.[8] Voice has been previously used to perform different tasks, for example, to interact with a tour-guide robot,[9] where the authors present a methodology for voice-enabled interfaces to be applied in a tour-guide robot. Although the system was tested in real conditions with multiple users in an exposition space, results about the speech interface performance are not provided to support the validity of the speech recognition method. Wheelchairs have also been controlled via voice.[10,11] Nishimori[10] proposed a set of nine Japanese commands to control a wheelchair. The system uses Julian as the grammar-based recognition parser. It must be pointed out that, even with a recognition rate above 97%, these results will probably worsen since the set of commands is highly reduced, consisting of only one or two simple words. Galindo[11] controls a wheelchair with voice commands as well. However, the command recognition rate is only 83.4%, because the voice-recognizer software exhibits a high level of accuracy but it sometimes fails due to a

* Corresponding author. E-mail: apg@dte.uma.es

https://doi.org/10.1017/S0263574714000010 Published online by Cambridge University Press

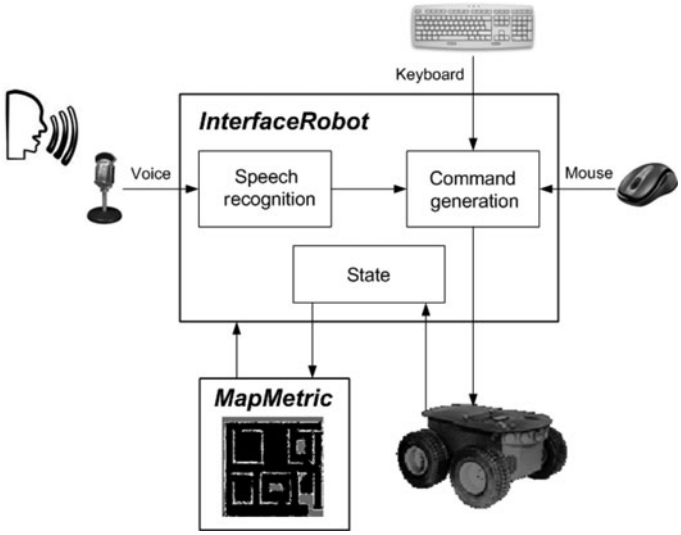


Fig. 1. Structure of the system for voice teleoperation.

noisy environment or poor training by the user. In Barkana's work,[12] a speech recognition technique has been developed to include patient's feedback into a robot-assisted rehabilitation system, with accuracy equal to or higher than 90%. As well as for Barkana, there are other works which deal with voice recognition applied to help in health services, such as the robotic scrub nurse,[13] Quirubot. It can identify up to 27 surgical instruments and more than 82 spoken instructions with a recognition rate above 87.7%. A lower rate (80.6%) is achieved in a system concerned with surveillance tasks,[14] or even it is not provided by the authors, as in the work by Glas,[15] where multiple robots are teleoperated in a social fashion. Finally, it must be pointed out that, as in this paper, examples of teleoperation of a robotic platform can also be reported.[16,17] Chatterjee[16] uses a fuzzy-neural network as a building block in a robot system, controlled by voice-based commands. Medicherla[17] describes the implementation of a voice-controllable intelligent user interface for a Pioneer 3-AT mobile robot. However, both works[16,17] do not show results about their recognition rates.

The integration of voice control into a HRI would make humans feel more comfortable during the interaction. Thus, speech is the primary form of interaction between human and robot in the HRI for teleoperation of a robotic platform presented in this paper, since natural spoken is the most user-friendly means of interacting with machines and, from the human standpoint, spoken interactions are easier than others. As well as with the user's voice, the interaction between a user and a robot can be achieved via a mouse and a keyboard.

The HRI as a whole is one of the contributions of the paper. It is integrated into a robot teleoperation system. Hence, this system is first presented in Section 2. Then, the interface is analyzed in Section 3, where the user's voice is the main input to control the robotic platform. In this work special attention has been paid to the design of a new speech recognition system for Spanish speakers, which is based on a user-dependent acoustic model to teleoperate the robot via a set of commands. This speech recognition mechanism is one of the main contributions of the paper. It is presented in Section 4. Results are described in Section 5. Finally, conclusions and future work are presented in Section 6.

2. System Architecture

The structure of the system is presented in Fig. 1. The style of the *Distributed and Layered Architecture* (DLA) is used to support this structure.[18] The *Distributed and Layered Architecture* implements a distributed shared memory scheme to let distributed processes interact in an asynchronous way. The following main features of DLA can be highlighted: (i) simple, as the user only has to exchange data using the DLA library; (ii) extendable, adding new modules and/or hardware with new functions and behaviors to the system without further changes in the rest of the modules; and (iii) portable, since the DLA library has been developed in *MATLAB*, *JAVA,* and *C* for *Linux* and *Windows*. Hence, any module using those languages and new robotic platforms could be integrated with existing modules.

https://doi.org/10.1017/S0263574714000010 Published online by Cambridge University Press

The system works on *Linux* operating system. It is made up of different modules which allow the robot to be teleoperated:

1. *InterfaceRobot*. The HRI is described in detail in Section 3. It has been integrated into the robot system (Fig. 1) to effectively achieve its teleoperation. The robot performs tasks ordered by the user of the HRI. The interface receives the user's input in the form of a voice command captured with a microphone, mouse command, or keyboard command, the voice command being the primary control input. In response to these commands, the interface communicates with the robotic platform to apply its motion commands and capture its state. Furthermore, it shows the geometrical representation of the environment, which is built in module *MapMetric*. As depicted in Fig. 1, the module *InterfaceRobot* is composed of several interconnected submodules:
   - State. It captures odometrical data and sonar readings from the robotic platform, to acquire information of the environment. These data are made available to build the geometrical model of the environment with the module *MapMetric*. Although in this work the state of the robot is only used to build the model of the environment, it would be easy to extend the system with new behaviors. With the DLA features, applying the state of the robot as the needed input sensor information to achieve a shared control strategy for navigation, the autonomy of the robotic platform will improve.
   - Speech recognition. It involves the user's voice processing entity. Its output is the command to be sent to the robotic platform. It is further described in Section 4.
   - Command generation. It receives orders from three different sources: keyboard, mouse, and the most important, the user's voice saved with a microphone. Its output is the command to be sent to the robotic mobile platform.
2. *MapMetric*. It builds a geometrical map of the environment using occupancy grids.[19] This grid is fast and easy to construct and update, efficiently combining any number of sensor readings. In grid-based representations each grid cell of the map is associated to a specific position of the environment, which is implicitly given by cell coordinates (x,y). Therefore, the robot position within the model can be determined by its position and orientation in the environment. Any cell yields an occupancy value which is equal to the probability of that cell being occupied according to current and past sensor readings. The metric map will allow the user of the HRI to progressively increase the knowledge of the robot environment while teleoperating.

3. Human-robot interface for teleoperation

Teleoperation interface design is necessary to enable interaction between the user and the robot, providing the user with situation awareness and controls for operating the robot.[15] Three aspects must be considered when designing a HRI: effectiveness, usability, and aesthetics. Effectiveness refers to the behavior of the HRI, which must perform its tasks as the user expects.[20] An effective HRI inspires confidence and sense of control in the user.[21] Usability measures how easy to learn and use a product is.[22] It must be realized that usability is not a unique attribute but has several components. It is traditionally associated with learnability, efficiency, memory, errors, and satisfaction.[23] As well as effectiveness and usability, aesthetics should also be sought, to improve the utility and global performance of the HRI,[24] since it increases the user's perception of the HRI usability.

In the literature, different works on criteria applied to the design of effective, usable, and aesthetic HRI can be found. Johnson[25] proposed eight points to consider, while Spolsky[20] established a set of advices to design a HRI based on the user's expectations. Probably, the three main lists of criteria considered when designing a HRI are Schneiderman's eight golden rules of interface design,[26] Nielsen's 10 usability heuristics,[27] and Tognazzini's basic principles for interface design.[21] All of these authors propose valid criteria to design the HRI presented in this paper. Even more, some of them appear in all of the lists, as consistency does. However, the full list of criteria of one specific author could not be directly applied to the design of our interface. Thus, a new set of criteria has been developed, combining the work of different authors, mainly Scheiderman,[26] Nielsen,[27] and Tognazzini.[21] The list of criteria to design the HRI for voice teleoperation presented in this paper is as follows:

- Consistency: any task is performed in the same way a user is used to.
- Effective use of colour.

https://doi.org/10.1017/S0263574714000010 Published online by Cambridge University Press

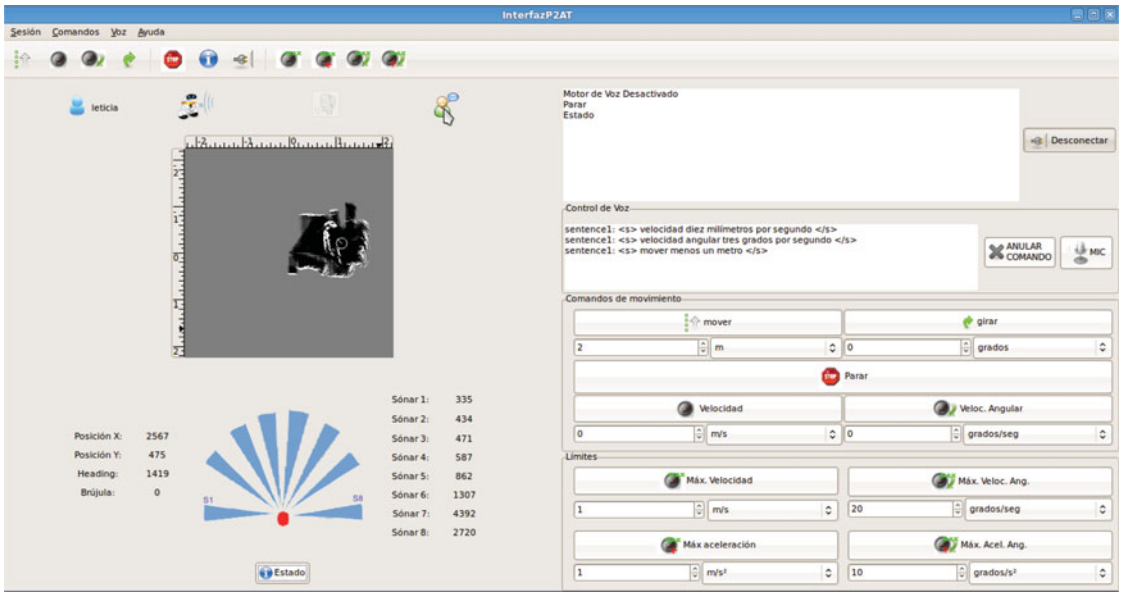


Fig. 2. (Colour online) Human-robot interface for voice teleoperation.

- Fitt's law: the time to reach a control depends on its size and distance.
- State visibility: the user will always be informed about changes in the system.
- Known language: employ simple words which can be easily understood by users.
- Recognize instead of remember: all of the elements to be used by the user will be visible and attainable.
- Minimalistic and aesthetic design: lesser the information shown, easier it is for the user to see what there is.

In this paper, voice is the main form of communication between the human and the robot in the proposed HRI, what we might call the speech interface. Specific criteria to design a speech interface have also been reported.[28,29] Nevertheless, as for the whole interface, the set of criteria of any author does not meet the needs of our application. Hence, the following criteria have been selected to design the proposed speech interface:

- Consistency regarding the dialogue and errors recovering: notify about executed commands and alert about non-recognized commands.
- Support experienced and non-experienced users.
- Simplicity: use a similar simple structure for all commands.

The HRI consists of a form where all the components are arranged, as it is shown in Fig. 2. A menu bar, a toolbar, a state area, and a control area are found in the proposed HRI. The menu bar holds an ordered list of the actions which can be run. The toolbar gives a rapid access to the set of commands which can be sent to teleoperate the robot. In the state area, information regarding the robot state (sonar readings and odometrical data), the geometrical representation of the environment, and the state of the speech engine is presented. Finally, components related to commands sending and voice control are located in the control area.

4. Speech Recognition

The proposed HRI provides a speech interface to let the user use his/her voice to teleoperate the robot. To this purpose, a new speech recognition system has been developed. It takes the user's spoken sounds and returns the corresponding command to be sent to the robot in the form of a sentence. To achieve the recognition, a voice decoder is needed. Since the system runs on *Linux* operating system, the voice decoder must be available for *Linux*. Currently, there are several voice decoder software for *Linux*. Among them *CVoiceControl*, DynaSpeak, *LumenVox Speech Engine*,

https://doi.org/10.1017/S0263574714000010 Published online by Cambridge University Press

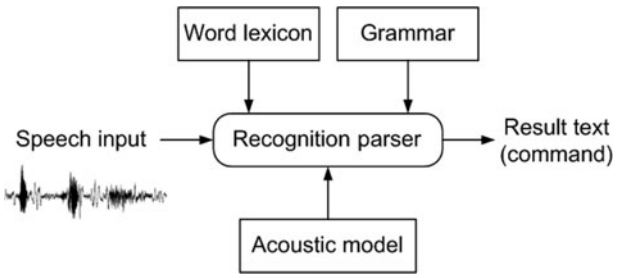


Fig. 3. Structure of the *Julius*/*Julian* grammar-based recognition parser.

Verbio ASR, *Simon*, *Sphinx,*[30] and *Julius*/*Julian*[31] may be highlighted. In this work, it was aimed to look for an open-source software, which was able to manage a Spanish language and an acoustic model. Therefore, only *Simon*, *Sphinx,* and *Julius*/*Julian* met these constraints. *Simon* was not suitable to be used in Spanish. Hence, *Sphinx* and *Julius*/*Julian* were the two major native solutions. However, *Sphinx* does not have a clear documentation to adapt the acoustic model to the Spanish needs of this work. Thus, *Julius*/*Julian* was chosen as it is the voice decoder which better suited our requirements.

Julius/*Julian* is an open-source large vocabulary continuous speech recognition (LVCSR) engine. It combines high recognition rate and high-speed speech recognition. It is based on word N-gram and context-dependent hidden Markov models (HMM), working at almost real time. Hence, it becomes suitable for human-robot tasks. *Julius*/*Julian* is used in word dictation tasks, using probabilistic models. Furthermore, it can also be used in command and control applications. In the latter case, *Julius*/*Julian* works as a grammar-based recognition parser, using a hand-designed finite-state machine grammar and being useful to build voice command systems with a small vocabulary.

This work relies on a command-based HRI to teleoperate a robot via the user's voice. Then, *Julius*/*Julian* acts as a grammar-based recognition parser, whose structure is shown in Fig. 3. *Julius*/*Julian* needs a word lexicon, a grammar, and an acoustic model. The word lexicon consists of all the words which can be recognized by the parser. The grammar contains the rules about how words can be combined in a sentence. When the recognition parser receives a speech input, it takes the acoustic model to perform a maximum likelihood search using the degrees of freedom given by the grammar, and outputs the word sequence that best matches the input utterance.

Although *Julius*/*Julian* was originally developed for Japanese, it has been successfully applied for various other languages such as English,[32] French,[33] Mandarin Chinese,[34] Thai,[35] Estonian,[36] Slovenian,[37] and Korean.[38] Since *Julius*/*Julian* computes the most likely sentence for a given speech input, the recognition accuracy largely depends on the model. Thus, it would be desirable to have a specific acoustic model for each language. In this work, a Spanish acoustic model has been developed to improve the recognition rate of a speech input from a Spanish speaker.

Next sections further describe the technique. First, the design of the word lexicon is explained in Section 4.1. The word lexicon contains all the words in Spanish that the system can recognize in the form of voice commands. Then, Section 4.2 presents one of the main contributions of the work, the process to obtain a user-dependent acoustic model. To that purpose, a set of training sentences is needed. Before training, data are prepared. Next, recorded data are processed to extract the corresponding feature vectors. Finally, the training stage is carried out.

4.1. Word lexicon and grammar

The word lexicon is based on a language model in Spanish for Sphinx. It contains all the words the system can recognize. The grammar is composed of the rules and restrictions which words in a speech input must satisfy. The interaction between the human and the robot can be achieved with 12 different commands, presented in Table I. Commands have been classified in four types:

1. Connection: to connect and disconnect from the robot.
2. Movement: to move or to stop the robot.
3. State: to receive information from the robot sensors.
4. Parameter: to configure different values of the robot navigation.

https://doi.org/10.1017/S0263574714000010 Published online by Cambridge University Press

Table I. Spanish voice commands for robot teleoperation.

Short commands (command)			
Commands	Type	English translation	Description
Conectar	Connection	Connect	Connect to the robot
Desconectar	Connection	Disconnect	Disconnect from the robot
Parar	Movement	Stop	Stop the robot
Estado	State	State	Read the robot state
Long commands (command + number + unit)			
Commands	Type	English translation	Description
Mover	Movement	Move	Move linearly a distance
Girar	Movement	Turn	Rotate a number of degrees
Velocidad	Movement	Velocity	Move linearly at a specified velocity
Velocidad angular	Movement	Rotational velocity	Rotate at a specified rotational velocity
Máxima velocidad	Parameter	Maximum velocity	Maximum translational velocity
Máxima velocidad angular	Parameter	Maximum rotational velocity	Maximum rotational velocity for the robot
Máxima aceleración	Parameter	Maximum acceleration	Maximum translational acceleration for the robot
Máxima aceleración angular	Parameter	Maximum rotational acceleration	Maximum rotational acceleration for the robot

Table II. Numbers for long commands.

Number	Recognized word(s)	Number	Recognized word(s)
0	Cero	21	Veintiuno, veintiún
1	Uno, un	22	Veintidós
2	Dos	23	Veintitrés
3	Tres	24	Veinticuatro
4	Cuatro	25	Veinticinco
5	Cinco	26	Veintiséis
6	Seis	27	Veintisiete
7	Siete	28	Veintiocho
8	Ocho	29	Veintinueve
9	Nueve	30	Treinta
10	Diez	40	Cuarenta
11	Once	50	Cincuenta
12	Doce	60	Sesenta
13	Trece	70	Setenta
14	Catorce	80	Ochenta
15	Quince	90	Noventa
16	Dieciséis	100	Cien, ciento
17	Diecisiete	200	Doscientos
18	Dieciocho	300	Trescientos
19	Diecinueve	400	Cuatrocientos
20	Veinte	500	Quinientos

From a speech recognition point of view, commands are divided into two groups, short commands and long commands. Short commands consist of a single word, its command identifier. Long commands, otherwise, consist of a sentence, a list of ordered words. For these commands, one, two, or three words are needed to identify the command, followed by a Spanish spoken number. Table II shows the Spanish word(s) for numbers which can be recognized in the HRI. Positive and negative values are allowed. Negative numbers must be preceded by the word *menos* (minus). The

https://doi.org/10.1017/S0263574714000010 Published online by Cambridge University Press

Table III. Units for each command.

Commands	Recognized word(s)	Units
Mover	Metro(s)	m
	Centímetro(s)	cm
	Milímetro(s)	mm
Girar	Grado(s)	°
Velocidad Máxima velocidad	Metro(s) por segundo	m/s
	Centímetro(s) por segundo	cm/s
	Milímetro(s) por segundo	mm/s
Velocidad angular Máxima velocidad angular	Grado(s) por segundo	°/s
Máxima aceleración	Metro(s) por segundo cuadrado	m/s^2
	Centímetro(s) por segundo cuadrado	cm/s^2
	Milímetro(s) por segundo cuadrado	mm/s^2
Máxima aceleración angular	Grado(s) por segundo cuadrado	$°/s^2$

system is also able to recognize the word *y* (and) to constitute numbers like *cuarenta y uno* (41) or *setenta y cinco* (75).

The last part of the sentence in a long command is the unit. Each command has associated a list of permitted units. This association is depicted in Table III, together with words in Spanish the system recognizes for each unit. Units can be expressed in singular or plural form, depending on the magnitude of the preceding number.

4.2. Acoustic model

The acoustic model contains a statistical representation of phonemes which make up each of the words of the grammar. Currently, *Julius*/*Julian* offers acoustic models for Japanese and English, but there is not available a Spanish acoustic model. Thus, in this work, it has been developed with the Hidden Markov Model Toolkit (HTK)[39] for the set of commands in Table I. Hidden Markov Model Toolkit is a toolkit for building signal processing tools based on HMM. Although HTK has been primarily used for speech recognition, it is also useful in other applications, including speech synthesis or character recognition.

To build the acoustic model a set of training sentences is needed to obtain the model after a training process. In this work the proposed acoustic model is user-dependent, using training data from a single user. Therefore, each user has to train his/her own acoustic model with his/her data. Eight hundred and two words grouped in 72 sentences have been considered for the training stage, containing words present and not present in the grammar. Three aspects were borne in mind when designing these sentences: (i) consider data with good phonetic balance and coverage; (ii) include from three to five repetitions (in different triphthongs) of each of the phonemes of the Spanish language; and (iii) include two or three repetitions of each of the triphthongs used in the grammar.

Figure 4 shows the procedure to obtain, for each user, the acoustic model. The same procedure would be followed to build a speaker independent system. Nevertheless, this would require large amounts of training data from a large number of speakers and there would be a consequential increase in the training time and in model complexity.

Before training, data must be prepared. Training sentences have to be recorded with a microphone. It must be pointed out that it has not been necessary to formulate any explicit consideration about the quality of the microphone used to perform both the training and recognition processes. If a good recognition rate is desired, it is just mandatory to use the same microphone for training and for recognition. Next, recorded data are parameterized to convert the raw speech waveforms into a sequence of feature vectors. Likely, linear prediction coefficients (LPC) and Mel frequency cepstral coefficients (MFCC) are the most frequently used parameters. Among the two techniques, MFCC is the more widely used in speech recognition because with this method the characteristics of the vocal tract are taken into account.[40] Thus, in this paper, MFCC have been used to parameterize the speech input.

Mel frequency cepstral coefficients are derived from Fast Fourier Transform (FFT)-based log spectra. In speech recognition applications, 12 MFCC are commonly used.[12] It is also typical to

https://doi.org/10.1017/S0263574714000010 Published online by Cambridge University Press

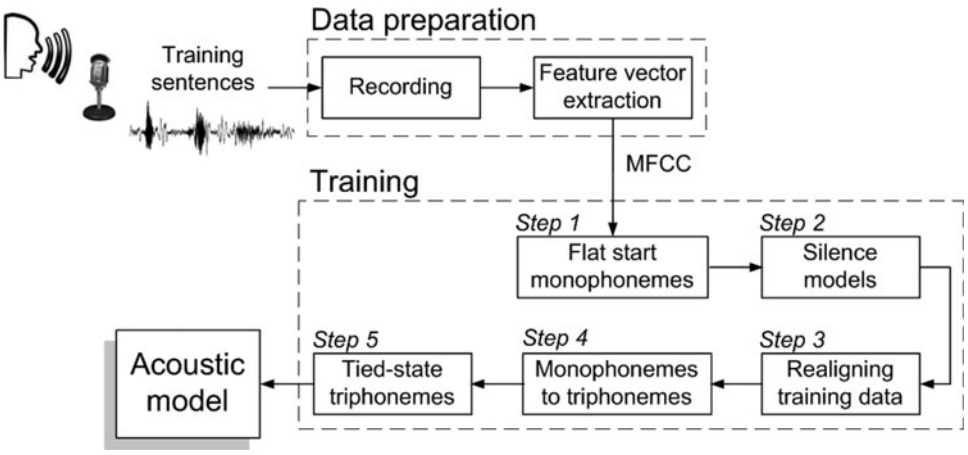


Fig. 4. Procedure to obtain the acoustic model.

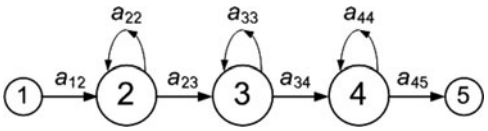


Fig. 5. HMM: three-state left-right topology.

compute delta coefficients[41–43] and acceleration coefficients.[44] Delta and acceleration coefficients are usually calculated because the performance of the speech recognition system can be greatly enhanced by adding time derivatives, with a large amount of phonetic information, to the basic static parameters. Delta coefficients are computed using the following regression expression:

$$d_t = \frac{\sum_{\theta=1}^{\Theta} \theta(c_{t+\theta} - c_{t-\theta})}{2\sum_{\theta=1}^{\Theta} \theta^2}, \tag{1}$$

where d_t is the delta coefficient in time t, obtained from cepstral coefficients $c_{t-\theta}$ and $c_{t+\theta}$, and $\Theta = 2$ defines a two-frame window. Expression (1) is also applied to delta coefficients to obtain acceleration ones. Log-energy of the speech input is sometimes appended to cepstral coefficients.44 Besides, other approaches carry out a mean normalisation43 to compensate for long-term spectral effects. In this work, tests have been performed with different sets of features. These tests are described in Section 5.

When speech data are parameterized, the training stage starts. The procedure consists of the five steps depicted in Fig. 4.

Step 1. Creating flat start monophonemes.

First, the prototype model of the HMM has to be defined. It is assumed that the pronunciation of a phoneme goes through three sequential phases. In the first phase, the vocal tract is changing its shape, to accomplish the transition from the previous phoneme. In the second stage, the sound is considered to be pure and, in the third, the phoneme finishes and the transition to the next phoneme starts. Thus, a three-state left-right topology has been chosen, which is commonly used in speech recognition applications. The three-state topology is shown in Fig. 5, where a_{ij} is the transition probability from state i to state j, and states 1 and 5 are just used to concatenate the models.

At this point, each flat start phoneme will be defined by a HMM. Then, phonemes are re-estimated according to MFCC feature vectors, obtaining a final set of initialized monophoneme HMMs.

Step 2. Fixing the silence models.

The previous step generates a three-state left-to-right HMM for each phoneme and also a HMM for the silence model *sil*, which models long pauses at the end of a sentence. However, a silence

https://doi.org/10.1017/S0263574714000010 Published online by Cambridge University Press

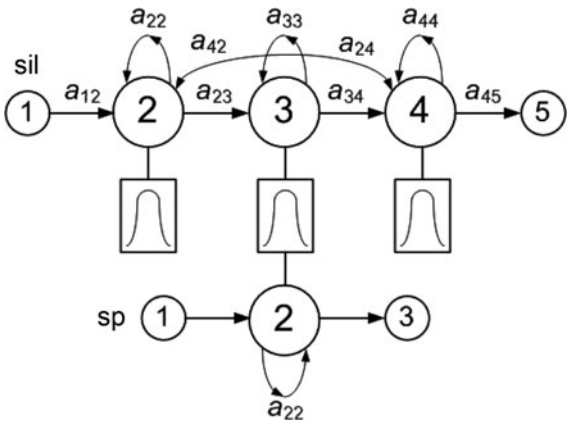


Fig. 6. Silence models.

model for short pauses, *sp*, has not been created yet. These pauses occur between words. Thus, *Step 2* modifies the silence model *sil* to consider the model *sp*.

The silence model *sil* has the same topology as each phoneme does. Nevertheless, this model works with silent periods of variable lengths. Hence, extra transitions from states 2 to 4 and from states 4 to 2 in the silence model are added (Fig. 6), making the model more robust by allowing individual states to absorb the various impulsive noises in the training data. The backward skip allows this to happen without committing the model to transit to the following word.

Next, the model for short pauses, *sp*, is created. This model ensures that short silences between words are not assigned to other models during training. It consists of a one-state model, a *tee-model*, which has a direct transition from entry to exit node. The emitting state is tied to the centre state of the modified silence model *sil*. At the end of this step, a set of phoneme HMMs including *sp* is available.

Step 3. Realigning the training data.

The word lexicon contains words which, in some cases, can have multiple pronunciations. However, this characteristic was not considered when preparing data before training. Phoneme models created after *Steps 1* and *2* could be used to realign the training data. In this step, created HMMs are recomputed to consider all pronunciations for each word, taking into account the pronunciation that best matches the acoustic data. The output of this step is the final monophoneme HMM set.

Step 4. Making triphonemes from monophonemes.

Given a set of monophoneme HMMs, the final stage of model building is to create context-dependent triphoneme HMMs. This task is accomplished in two steps. First, triphoneme models are created by copying monophonemes and re-estimating. Phoneme HMMs are cloned, tying all the transition matrices in each triphoneme set to achieve an initial set of context-dependent models. Then, as for the case of monophonemes in *Step 1*, triphonemes models are re-estimated.

Step 5. Making tied-state triphonemes.

The outcome of the previous step is a set of triphoneme HMMs with all triphonemes in a phoneme set sharing the same transition matrix. When estimating these models, many of the variances in the output distributions would have been floored since there would be insufficient data associated with many of the states. The last step in the model building process is to tie states within triphoneme sets in order to share data and, thus, be able to make robust parameter estimates. A clustering algorithm, based on decision trees, has been used to determine which states to tie.

Decision trees are based on asking questions about the left and right contexts of each triphoneme. Questions are related to the phonetics of phonemes present in the language. Phonetic features are specific of each language. Thus, available questions for Japanese or English are not suitable to develop a Spanish acoustic model. Hence, a set of questions has been defined to cover any possible context.

https://doi.org/10.1017/S0263574714000010 Published online by Cambridge University Press

Table IV. Number of appearances of commands in the testing set.

Command	Number of appearances
Conectar	2
Desconectar	1
Parar	1
Estado	1
Mover	12
Girar	11
Velocidad	8
Velocidad angular	9
Máxima velocidad	11
Máxima velocidad angular	12
Máxima aceleración	16
Máxima aceleración angular	16

The HRI provides a Training Assistant to generate the acoustic model. This tool allows each user to record his/her training data (72 sentences). This recording task may last about half an hour. The rest of the stages of the procedure shown in Fig. 4 to calculate the acoustic model are accomplished by the HRI itself. Due to the user-dependent training, the acoustic model adapts to the voice and characteristic pronunciation of the user. Thus, when teleoperating the robot, the user would not have to speak in a different manner as he/she usually does to ensure that voice commands are recognized, making the human–robot communication more natural.

5. Experiments and Results

In order to test the performance of the proposed HRI and speech interface, several tests have been performed. Sections 5.1 and 5.2 describe these results. First, the results of the speech recognition tests are presented, in terms of the recognition rate. Then, navigation tests with the robotic platform are shown. *Linux Ubuntu* 9.10 was used as the operating system.

5.1. Speech recognition tests

Julius/*Julian*, the voice decoder, uses the word lexicon, the grammar, and the user-dependent acoustic model to determine from an utterance the corresponding teleoperation command. To evaluate the speech recognition interface, it has been tested with two users in four cases of MFCC feature vectors:

Case 1. 39-dimensional vector: log-energy (1), static MFCC (12), delta coefficients (13), and acceleration coefficients (13).
Case 2. 25-dimensional vector: log-energy (1), static MFCC (12), delta coefficients without their absolute energy (12), and cepstral mean normalization.
Case 3. 26-dimensional vector: log-energy (1), static MFCC (12), delta coefficients (13), and cepstral mean normalization.
Case 4. 26-dimensional vector: log-energy (1), static MFCC (12), and delta coefficients (13).

Each user trained the system to obtain a personal acoustic model with a set of 802 words grouped in 72 sentences. Then, they tested the generated acoustic model with 678 words grouped in 100 test command sentences, including all the possible commands in Table I. The number of each teleoperation command included in the testing set is detailed in Table IV.

Long commands were purposefully made up with different arbitrary numeric values and different units, as can be appreciated in Tables V and VI, where the 100 teleoperation commands to test the speech interface are presented.

The performance of the speech recognition system has been evaluated with the word recognition rate (*WRR*), a common metric to evaluate its behaviour.[31] It is commonly used because it is a direct and objective measure of speech recognition performance.

https://doi.org/10.1017/S0263574714000010 Published online by Cambridge University Press

Table V. Test commands 1–50.

No.	Teleoperation command	Description
1	Conectar	Connect
2	Mover veintiséis metros	Move 26 m
3	Máxima aceleración quince metros por segundo cuadrado	Maximum acceleration 15 m/s^2
4	Máxima aceleración angular once grados por segundo cuadrado	Maximum rotational acceleration 11°/s^2
5	Velocidad diez metros por segundo	Velocity 10 m/s
6	Girar once grados	Turn 11°
7	Máxima velocidad treinta metros por segundo	Maximum velocity 30 m/s
8	Parar	Stop
9	Máxima velocidad angular cincuenta grados por segundo	Maximum rotational velocity 50°/s
10	Desconectar	Disconnect
11	Máxima aceleración angular cuarenta grados por segundo cuadrado	Maximum rotational acceleration 40°/s^2
12	Máxima velocidad angular veintisiete grados por segundo	Maximum rotational velocity 27°/s
13	Girar nueve grados	Turn 9°
14	Máxima aceleración veintinueve metros por segundo cuadrado	Maximum acceleration 29 m/s^2
15	Máxima aceleración tres metros por segundo cuadrado	Maximum acceleration 3 m/s^2
16	Máxima velocidad treinta y dos metros por segundo	Maximum velocity 32 m/s
17	Velocidad angular veinticuatro grados por segundo	Rotational velocity 24°/s
18	Estado	State
19	Mover setenta y seis centímetros	Move 76 cm
20	Máxima aceleración angular veintidós grados por segundo cuadrado	Maximum rotational acceleration 22°/s^2
21	Máxima aceleración setenta y siete metros por segundo cuadrado	Maximum acceleration 77 m/s^2
22	Girar ochenta y ocho grados	Turn 88°
23	Máxima aceleración angular cuarenta grados por segundo cuadrado	Maximum rotational acceleration 40°/s^2
24	Máxima velocidad angular cuarenta y cinco grados por segundo	Maximum rotational velocity 45°/s
25	Máxima velocidad cincuenta y dos metros por segundo	Maximum velocity 52 m/s
26	Velocidad seis metros por segundo	Velocity 6 m/s
27	Velocidad angular veinticinco grados por segundo	Rotational velocity 25°/s
28	Máxima aceleración angular sesenta y dos grados por segundo cuadrado	Maximum rotational acceleration 62°/s^2
29	Mover noventa y cuatro metros	Move 94 m
30	Girar veintitrés grados	Turn 23°
31	Máxima aceleracion angular veintidós grados por segundo cuadrado	Maximum rotational acceleration 22°/s^2
32	Máxima aceleración angular sesenta y ocho metros por segundo cuadrado	Maximum rotational acceleration 68°/s^2
33	Girar dieciocho grados	Turn 18°
34	Velocidad diecisiete metros por segundo	Velocity 17 m/s
35	Máxima velocidad un metro por segundo	Maximum velocity 1 m/s
36	Máxima velocidad angular ocho grados por segundo	Maximum rotational velocity 8°/s
37	Máxima velocidad angular cinco grados por segundo	Maximum rotational velocity 5°/s
38	Máxima aceleración angular 2 grados por segundo cuadrado	Maximum rotational acceleration 2°/s^2
39	Mover treinta y siete centímetros	Move 37 cm
40	Mover siete metros	Move 7 m
41	Máxima velocidad setenta metros por segundo	Maximum velocity 70 m/s

https://doi.org/10.1017/S0263574714000010 Published online by Cambridge University Press

Table V. Continued.

No.	Teleoperation command	Description
42	Move trece metros	Move 13 m
43	Girar cuarenta grados	Turn 40°
44	Máxima aceleración angular noventa y seis grados por segundo cuadrado	Maximum rotational acceleration 96°/s^2
45	Máxima velocidad angular cincuenta grados por segundo	Maximum rotational velocity 50°/s
46	Conectar	Connect
47	Máxima velocidad angular cuatro grados por segundo	Maximum rotational velocity 4°/s
48	Máxima aceleración angular nueve grados por segundo cuadrado	Maximum rotational acceleration 9°/s^2
49	Velocidad angular diez grados por segundo	Rotational velocity 10°/s
50	Máxima aceleración angular dieciséis grados por segundo cuadrado	Maximum rotational acceleration 16°/s^2

Let N_w be the number of words. Let S, D, and I also be, respectively, the number of substituted words, deleted words, and inserted words. Then, *WRR* is defined as

$$WRR = 1 - \frac{S + D + I}{N_w}. \tag{2}$$

Table VII shows the results for the *WRR* in the four test cases of MFCC feature vectors. The word recognition rate has been calculated, for each user, for the 100 test command sentences (see Tables V and VI). The *WRR* is above 98.8% in all cases. It is believed that such successful results have been obtained due to the user-dependent acoustic model for the set of commands.

The best *WRR* is obtained in *Case 2*, the test with the shortest MFCC feature vector and, hence, with the least time consumption. The recognition rate in *Case 2* has been perfect, with no substitutions, deletions, or inserted words. This result does not mean that it can be asserted that the speech interface will never make any mistake. However, it may be considered that the system achieves a high recognition rate, provided each user teleoperates the robot with his/her trained and personal acoustic model. It must also be pointed out that a higher number of features does not imply a better performance and a higher recognition rate. If there are several correlated features which contribute with redundant information, the recognition rate will decrease.

Although *WRR* is a direct and objective measure of speech recognition performance, the proposed recognition is based on commands rather than on words. Thus, we propose a new recognition rate, the command success rate (*CSR*). It is defined as the number of correct recognized sentences (*RS*) among the total number of sentences (*TS*):

$$CSR = \frac{RS}{TS} \tag{3}$$

Table VIII shows the results for the *CSR* in the four test cases of MFCC feature vectors. The command success rate has also been calculated, for each user, for the 100 test command sentences (see Tables V and VI). As well as for the *WRR*, the *CSR* has yielded a very high recognition rate, above 95% in all cases. This situation occurs even when many on the test sentences, the speech interface must recognize, are a combination of up to 10 words. It must be pointed out that, for each user, both the *WRR* (Table VII) and the *CSR* (Table VIII) are different for the four cases, since each case considers a MFCC feature vector.

The recognition rates are higher than the results obtained in other command-based robotic systems.[11,45,46] The rate only reaches 87.5% in Marín's approach.[45] Galindo[11] reported a rate of what we have defined as the *CSR* of only 83.4%, noticing that their system sometimes does not work properly due to poor training by the user. The recognition of commands is also worst when recognizing among a set of eight Spanish commands,[46] a rate of only 91.9%. However, this. result

https://doi.org/10.1017/S0263574714000010 Published online by Cambridge University Press

Table VI. Test commands 51–100.

No.	Teleoperation command	Description
51	Máxima aceleración doce metros por segundo cuadrado	Maximum acceleration 12 m/s^2
52	Máxima aceleración catorce metros por segundo cuadrado	Maximum acceleration 14 m/s^2
53	Máxima aceleración diecinueve centímetros por segundo cuadrado	Maximum acceleration 19 cm/s^2
54	Máxima aceleración veinte milímetros por segundo cuadrado	Maximum acceleration 20 mm/s^2
55	Máxima aceleración veintiún milímetros por segundo cuadrado	Maximum acceleration 21 mm/s^2
56	Máxima aceleración veintiocho centímetros por segundo cuadrado	Maximum acceleration 28 cm/s^2
57	Máxima aceleración treinta y un milímetros por segundo cuadrado	Maximum acceleration 31 mm/s^2
58	Máxima aceleración quinientos treinta y tres milímetros por segundo cuadrado	Maximum acceleration 533 mm/s^2
59	Máxima aceleración ciento treinta y seis centímetros por segundo cuadrado	Maximum acceleration 136 cm/s^2
60	Máxima aceleración treinta y ocho milímetros por segundo cuadrado	Maximum acceleration 38 mm/s^2
61	Máxima velocidad ciento cuarenta y un centímetros por segundo	Maximum velocity 141 cm/s
62	Máxima velocidad trescientos cuarenta y seis milímetros por segundo	Maximum velocity 346 mm/s
63	Máxima velocidad doscientos cuarenta y siete centímetros por segundo	Maximum velocity 247 cm/s
64	Máxima velocidad cuarenta y nueve milímetros por segundo	Maximum velocity 49 mm/s
65	Máxima velocidad ciento cincuenta y un centímetros por segundo	Maximum velocity 151 cm/s
66	Velocidad cincuenta y tres milímetros por segundo	Velocity 53 mm/s
67	Máxima aceleración angular doscientos cincuenta y cuatro grados por segundo	Maximum rotational acceleration 254°/s^2
68	Máxima aceleración angular cincuenta y cinco grados por segundo cuadrado	Maximum rotational acceleration 55°/s^2
69	Máxima aceleración angular trescientos cincuenta y ocho grados por segundo cuadrado	Maximum rotational acceleration 358°/s^2
70	Máxima aceleración angular sesenta y un grados por segundo cuadrado	Maximum rotational acceleration 61°/s^2
71	Máxima aceleración angular doscientos sesenta y dos grados por segundo cuadrado	Maximum rotational acceleration 262°/s^2
72	Máxima velocidad angular sesenta y tres grados por segundo	Maximum rotational velocity 63°/s
73	Máxima velocidad angular sesenta y cuatro grados por segundo	Maximum rotational velocity 64°/s
74	Máxima velocidad angular trescientos sesenta y cinco grados por segundo	Maximum rotational velocity 365°/s
75	Velocidad angular cuatrocientos sesenta y nueve grados por segundo	Rotational velocity 469°/s
76	Velocidad angular cuatrocientos setenta y dos grados por segundo	Rotational velocity 472°/s
77	Mover menos trescientos setenta y tres centímetros	Move –373 cm
78	Mover menos setenta y cinco milímetros	Move –75 mm
79	Move menos doscientos setenta y nueve centímetros	Move –279 cm
80	Move menos ochenta y un milímetros	Move –81 mm
81	Girar menos ochenta y tres grados	Turn –83°
82	Girar menos cuatrocientos ochenta y cuatro grados	Turn –484°

https://doi.org/10.1017/S0263574714000010 Published online by Cambridge University Press

Table VI. Continued.

No.	Teleoperation command	Description
83	Girar menos ochenta y seis grados	Turn –86°
84	Girar menos quinientos ochenta y siete grados	Turn –587°
85	Velocidad angular menos cuatrocientos ochenta y nueve grados por segundo	Rotational velocity –489°/s
86	Velocidad angular menos quinientos noventa grados por segundo	Rotational velocity –590°/s
87	Velocidad menos noventa y un milímetros por segundo	Velocity –91 mm/s
88	Velocidad menos quinientos noventa y cinco centímetros por segundo	Velocity –595 cm/s
89	Velocidad menos quinientos noventa y siete milímetros por segundo	Velocity –597 mm/s
90	Velocidad angular menos ciento noventa y ocho grados por segundo	Rotational velocity –198°/s
91	Máxima velocidad angular cien grados por segundo	Maximum rotational velocity 100°/s
92	Máxima velocidad doscientos milímetros por segundo	Maximum velocity 200 mm/s
93	Máxima aceleración angular trescientos grados por segundo cuadrado	Maximum rotational acceleration 300°/s^2
94	Máxima aceleración cuatrocientos centímetros por segundo cuadrado	Maximum acceleration 400 cm/s^2
95	Mover menos quinientos milímetros	Move –500 mm
96	Girar menos cien grados	Turn –100°
97	Velocidad angular menos cero grados por segundo	Rotational velocity –0°/s
98	Velocidad menos un centímetro por segundo	Velocity –1 cm/s
99	Mover menos un milímetro	Move –1 mm
100	Máxima aceleración un centímetro por segundo cuadrado	Maximum acceleration 1 cm/s^2

Table VII. *WRR* for test teleoperation commands.

	MFCC case (vector length)	N_w	S	D	I	$N_w - (S + D + I)$	*WRR* (%)
User 1	*Case 1* (39)	678	3	0	4	671	98.97
	Case 2 (25)	678	0	0	0	678	100
	Case 3 (26)	678	1	1	0	676	99.71
	Case 4 (26)	678	5	0	2	671	98.97
User 2	*Case 1* (39)	678	4	0	4	670	98.82
	Case 2 (25)	678	2	0	0	676	99.71
	Case 3 (26)	678	0	2	0	676	99.71
	Case 4 (26)	678	4	0	2	672	99.11

Table VIII. *CSR* for test teleoperation commands.

	MFCC case (vector length)	*TS*	*RS*	*CSR* (%)
User 1	*Case 1* (39)	100	97	97
	Case 2 (25)	100	100	100
	Case 3 (26)	100	98	98
	Case 4 (26)	100	95	95
User 2	*Case 1* (39)	100	96	96
	Case 2 (25)	100	98	98
	Case 3 (26)	100	98	98
	Case 4 (26)	100	96	96

https://doi.org/10.1017/S0263574714000010 Published online by Cambridge University Press

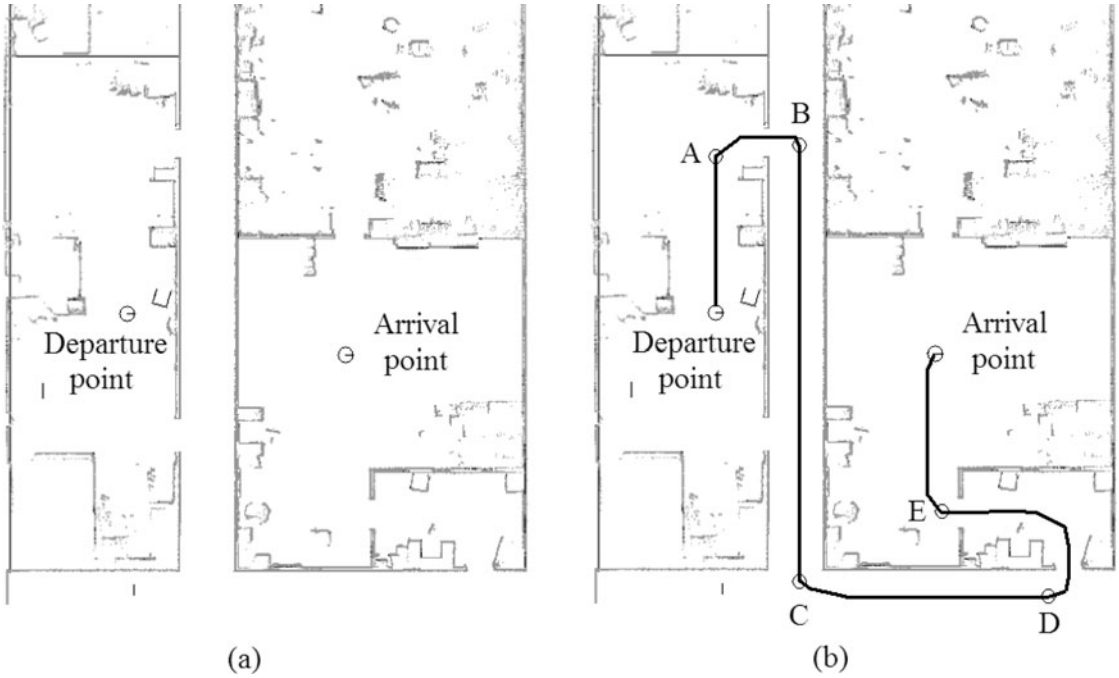


Fig. 7. (a) Environment to teleoperate the robot; (b) route followed by the robot.

is exactly the *WRR*, since commands consists of a unique word, without the possibility of adding numbers or units to commands, conforming a very simple set of commands

5.2. Robot navigation tests

The system has been tested with a Pioneer P2AT robot from ActivMedia equipped with eight frontal sonar sensors in simulated environments, with MobileSim, a software for simulating mobile robots and their environments. Figure 7(a) shows an environment yielding an area of 437 m^2 (19 m $\times$ 23 m). In this environment, a user was requested to perform the teleoperation of the robot from the Departure Point to the Arrival Point using exclusively voice commands. The user was familiar with ICT (Information and Communications Technology), but was not involved in robotics. Thus, prior to performing the task, the user was informed about the HRI, to acquire basic knowledge about its performance and the information about the state of the robot and the environment. The user was first trained for using the speech interface to obtain the personal acoustic model, and was also taught on voice commands.

The robot was supposed to be already connected and heading east at the departure point. The route followed by the robot while being teleoperated with the user's voice is depicted in Fig. 7(b). Voice commands sent to the robot by the user are detailed in Table IX. Commands in Table IX are ordered from the first sent command to the last sent command. The user was recommended to use as many types of commands as possible. Nevertheless, parameter commands were not recommended to be used since the user was not involved in robotics. Parameters, maximum velocities, and accelerations were previously configured to proper values to have a suitable navigation. Furthermore, the user was also requested to use as many commands as wanted. In the test environment, the user teleoperated the robot from the departure point to the arrival point with 39 commands.

Initially, the robot was heading east at the departure point. Therefore, the user sent a command to position the robot heading north (*Girar noventa grados*) to reach the upper door on the left side of the environment. To cross that door, a negative turn of –45° was executed at point A. Then, the robot was again turned to head south along the long corridor between both sides of the environment. At point B, the robot was commanded with the movement command *Mover quince metros*, until it reached point C. The objective at this point was to go towards the lower door of the right side of the environment. To that purpose, the user sent a sequence of commands (commands 17–25) from point C to point D (*Girar cincuenta grados*), to have the robot heading north and prepared to cross the door.

https://doi.org/10.1017/S0263574714000010 Published online by Cambridge University Press

Table IX. Sequence of commands from the departure point to the arrival point.

No.	Teleoperation command	Description
1	Girar noventa grados	Turn 90° (departure point)
2	Mover dos metros	Move 2 m
3	Velocidad veinte centímetros por segundo	Velocity 20 cm/s
4	Parar	Stop
5	Estado	State
6	Velocidad angular quince grados por segundo	Rotational velocity 15°/s
7	Mover ochenta y cinco centímetros	Move 85 cm
8	Mover cien milímetros	Move 100 mm
9	Girar menos cuarenta y cinco grados	Turn –45° (point A)
10	Velocidad un metro por segundo	Velocity 1 m/s
11	Parar	Stop
12	Estado	State
13	Velocidad angular menos diez grados por segundo	Rotational velocity -10°/s
14	Parar	Stop
15	Mover quince metros	Move 15 m (point B)
16	Parar	Stop
17	Velocidad angular diez grados por segundo	Rotational velocity 10°/s (point C)
18	Parar	Stop
19	Estado	State
20	Velocidad cincuenta milímetros por segundo	Velocity 50 mm/s
21	Parar	Stop
22	Velocidad menos ochenta milímetros por segundo	Velocity –80 mm/s
23	Parar	Stop
24	Velocidad diez centímetros por segundo	Velocity 10 cm/s
25	Girar cincuenta grados	Turn 50° (point D)
26	Mover veinte centímetros	Move 20 cm
27	Mover cuarenta centímetros	Move 40 cm
28	Mover menos quince centímetros	Move –15 cm
29	Velocidad angular diez grados por segundo	Rotational velocity 10°/s
30	Parar	Stop
31	Estado	State
32	Velocidad diez centímetros por segundo	Velocity 10 cm/s
33	Parar	Stop
34	Velocidad angular menos quince grados por segundo	Rotational velocity –15°/s (point E)
35	Parar	Stop
36	Estado	State
37	Velocidad quince centímetros por segundo	Velocity 15 cm/s
38	Parar	Stop
39	Girar noventa grados	Turn 90° (arrival point)

When the robot crossed that door, the user found a small room with another door on the left side of the room, which the user was also able to cross sending commands 26–33, arriving at point E. At this point, the task was almost completed. The user sent another turning command (*Velocidad angular menos quince grados por segundo*) to situate the robot heading north. With command 37 the robot started to move towards the north of the environment. When it was at the level of the arrival point, the last command was executed to have the robot heading east as well as for the starting point. Thus, the robot was successfully teleoperated from the departure point to the arrival point, even when the environment was initially unknown. It must be realized that the user performed the task autonomously and without any supervision. Although it seems that the user might have had difficulty to estimate the values of the movements the robot requires, a sequence of commands was used to overcome this situation. Nevertheless, to facilitate the task an alternative is suggested, the use of speed commands instead of position commands.

Otherwise, the user was not specifically requested to send commands to receive information about the robot sensors. However, five state commands out of 39 were executed. Hence, it is assumed that the user was continuously interested in having maximum information about the environment and

https://doi.org/10.1017/S0263574714000010 Published online by Cambridge University Press

the sonar readings of the robot, to be updated about the environment, and to know which action or command execute at each time instant. Besides, if the user performed so many updates of the robot state, it means that the information presented in the HRI is useful to enhance the experience when teleoperating the robot and that it can help the user to perform the task.

6. Conclusions

An approach to teleoperate a robot via a command-based HRI has been presented in this paper. The HRI has been integrated into a robot teleoperation system. Due to the features of the DLA, the system can be easily adapted to another platform, e.g., a wheelchair, even with new sensors as a laser range finder. One of the main contributions is the user-dependent acoustic model for Spanish speakers to increase the recognition rate of a set of voice teleoperation commands. The speech interface has been successfully tested with different users. It has been evaluated with the *WRR* and a new proposed metric, the *CSR*, more suitable for command-bases systems. They have yielded very high recognition rates, above the 98.8% for the *WRR* and above the 95% for the *CSR*. Furthermore, a Pioneer P2AT robot has been successfully teleoperated with the user's voice in navigation tasks.

Although the proposed set of commands to teleoperate the robot is fixed, one of the advantages of the presented system is its easy adaptation to new grammars and new platforms. Future work will focus on exploring the voice interaction with new grammars, the combination of speech recognition, and text-to-speech to achieve a most natural human–robot interaction and background noise. Also, the work will be oriented to increase the autonomy of the robot in a shared control strategy while it navigates through unknown environments.

Acknowledgements

This work has been partially supported by the Spanish Junta de Andalucía, under Project SIAD, No. P08-TIC-03991.

References

1. H. Hexmoor, C. Castelfranchi and R. Falcone, "Multiagent Systems, Artificial Societies, and Simulated organizations," **In**: *Agent Autonomy*, vol. 7 (Springer, Germany, 2003).
2. T. H. Sheridan and R. Parasuraman, "Human-automation interaction," *Rev. Hum. Factors Ergon.* **1**, 89–129 (2006).
3. F. Courreges, A. Edkie, G. Poisson and P. Vieyres, "Ergonomic Mouse Based Interface for 3D Orientation Control of a Tele-Sonography Robot," *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, USA (2009) pp. 61–66.
4. L. Zhang, Q. Huang, Q. Liu, T. Liu, D. Li and Y. Lu, "A Teleoperation System for a Humanoid Robot with Multiple Information Feedback and Operational Modes," *Proceedings of the 2005 IEEE International Conference on Robotics and Biomimetics*, Hong Kong, China (2005) pp. 290–294.
5. R. Mahony, F. Schill, P. Corke and Y. S. Oh, "A New Framework for Force Feedback Teleoperation of Robotic Vehicles Based on Optical Flow," *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, Kobe, Japan (2009) pp. 1079–1085.
6. T. Harada, T. Sato and T. Mori, "Human Motion Tracking System Based on Skeleton and Surface Integration Model Using Pressure Sensors Distribution Bed," *Proceedings of the Workshop of Human Motion*, Austin, USA (2000) pp. 99–106.
7. O. Fukuda, T. Tsuji, M. Kaneko and A. Otsuka, "A human-assisting manipulator teleoperated by EMG signals and arm motions," *IEEE Trans. Robot. Autom.* **19**(2), 210–222 (2003).
8. M. Ferre, J. Macias-Guarasa, R. Aracil and A. Barrientos, "Voice Command Generation for Teleoperated Robot Systems," *Proceedings of the IEEE ROMAN 1998*, Takamatsu, Japan (1998) pp. 679–685.
9. A. Drygajlo, P. J. Prodanov, G. Ramel, M. Meisser and R. Siegwart, "On developing a voice-enabled interface for interactive tour-guide robots," *Adv. Robot.* **17**(7), 599–616 (2003).
10. M. Nishimori, T. Saitoh and R. Konishi, "Voice Controlled Intelligent Wheelchair," *Proceedings of the 2007 SICE Annual Conference*, Takamatsu, Japan (2007) pp. 336–340.
11. C. Galindo, J. González and J. A. Fernández-Madrigal, "Control architecture for human-robot integration application to a robotic wheelchair," *IEEE Trans. Syst. Man Cybern.* **36**(5), 1053–1067 (2006).
12. D. E. Barkana, J. Das, F. Wang, T. E. Groomes and N. Sarkar, "Incorporating verbal feedback into a robot-assisted rehabilitation system," *Robotica* **29**(3), 433–443 (2011).
13. C. Pérez-Vidal, E. Carpintero, N. García-Aracil, J. M. Sabater-Navarro, J. M. Azorín, A. Candela and E. Fernández, "Steps in the development of a robotic scrub nurse," *Robot. Auton. Syst.* **60**(6), 901–911 (2012).
14. H. Lee and H. Ko, "Competing models-based text-prompted speaker independent verification algorithm," *Speech commun.* **48**(1), 28–44 (2006).

https://doi.org/10.1017/S0263574714000010 Published online by Cambridge University Press

15. D. F. Glas, T. Kanda, H. Ishiguro and N. Hagita, "Teleoperation of multiple social robots," *IEEE Trans. Syst. Man Cybern. Part A: Systems and Humans* **42**(3), 530–544 (2012).
16. A. Chatterjee, K. Pulasinghe, K. Watanabe and K. Izumi, "A particle swarm optimized fuzzy neural network for voice controlled robot systems," *IEEE Trans. Ind. Electron.* **52**(6), 1478–1489 (2005).
17. H. Medicherla and A. Sekmen, "Human-robot interaction via voice-controllable intelligent user interface," *Robotica* **25**(5), 521–527 (2007).
18. C. Urdiales, A. Bandera, E. J. Pérez, A. Poncela and F. Sandoval, "Hierarchical Planning in a Mobile Robot for Map Learning and Navigation," **In**: *Autonomous Robotic Systems* (Physica-Verlag, Heidelberg, Germany, 2003) pp. 2165–2188.
19. A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE J. Robot. Autom.* **3**, 249–265 (1987).
20. J. Spolsky, *User Interface Design for Programmers* (Apress Springer-Verlag, USA, 2001).
21. B. Tognazzini, *Tog on Interface* (Addison-Wesley, USA, 1992).
22. J. Nielsen, *Designing Web Usability. The Practice of Simplicity* (New Riders Publishing, Indiana, 2000).
23. J. Nielsen, *Usability Engineering* (Morgan-Kaufmann Publishers, San Francisco, 1993).
24. D. Norman, *Emotional Design: Why We Love (or Hate) Everyday Things* (Basic Books, New York, 2004).
25. J. Johnson, *GUI bloopers: Don'ts and Do's for Software Developers and Web Designers* (Morgan-Kaufmann Publishers, San Francisco, 2000).
26. B. Schneiderman, *Designing the User Interface* (Addison-Wesley, USA, 1997).
27. J. Nielsen, "Heuristic Evaluation," **In**: *Usability Inspection Methods* (J. Nielsen and R. Mack, eds.) (John Wiley & Sons, New York, 1994).
28. K. Severinson-Eklundh, A. Green and H. Hüttenrauch, "Social and collaborative aspects of interaction with a service robot," *Robot. Auton. Syst.* **42**(3–4), 223–234 (2003).
29. S. Hura, "Voice User Interfaces," **In**: *HCI Beyond the GUI* (Elsevier, UK, 2003) pp. 197–228.
30. K. F. Lee, H. W. Hon, M. Y. Hwang, S. Mahajan and R. Reddy, "The Sphinx Speech Recognition System," *Proceedings of the 1989 International Conference on Acoustic, Speech and Signal Processing*, Glasgow, UK (1989) pp. 445–448.
31. A. Lee, T. Kawahara and K. Shikano, "Julius - An Open-Source Real-Time Large Vocabulary Recognition System," *Proceedings of the Interspeech 2001*, Aalborg, Denmank (2001) pp. 1691–1694.
32. A. Lee and T. Kawahara, "Recent Development of Open-Source Speech Recognition Engine Julius," *Proceedings of the 2009 APSIPA Annual Summit and Conference*, Sapporo, Japan (2009) pp. 131–137.
33. D. Fohr, O. Mella, C. Cerisara and I. Illina, "The automatic news transcription system: ANTS, some real time experiments," *Proceedings of Interspeech 2004*, Jeju Island, Korea (2004) pp. 377–380.
34. D. Yang, K. Iwano and S. Furui, "Accent Analysis for Mandarin Large Vocabulary Continuous Speech Recognition," *Asian Workshop on Speech Science and Technology* (Tokyo, Japan, 2008) pp. 87–91.
35. M. Jongtaveesataporn, C. Wutiwiwatchai, K. Iwano and S. Furui, "Development of a Thai broadcast news corpus and an LVCSR system," *ASJ Annual Meeting* (Okayama, Japan, 2008).
36. T. Alumäe, "Large Vocabulary Continuous Speech Recognition for Estonian using Morphemes and Classes," *Proceedings of Interspeech 2006*, Pittsburgh, USA (2004) pp. 389–392.
37. T. Rotovnik, M. S. Maucec, B. Horvat and Z. Kacic, "A Comparison of HTK, ISIP and Julius in Slovenian Large Vocabulary Continuous Speech Recognition," *Proceedings of Interspeech 2002*, Denver, USA (2002) pp. 671–684.
38. J. G. Kim, H. Y. Jung and H. Y. Chung, "A Keyword Spotting Approach Based on Pseudo N-gram Language Model," *Proceedings of SPECOM, St. Petersburg*, Russia (2004) pp. 256–259.
39. S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason and D. Povey, *The HTK Book*, version 3.4 (Engineering Department, Cambridge University, Cambridge, UK,2006).
40. W. Yutai, L. Bo, J. Xiaoqing, L. Feng and W. Lihao, "Speaker Recognition Based on Dynamic MFCC Parameters," *Proceedings of the Conference on Image Analysis and Signal Processing IASP*, Ningbo, China (2009) pp. 406–409.
41. Y. Jeong, "Robust speaker adaptation based on parallel factor analysis of training models," *Electron. Lett.* **47**(7), 465–467 (2011).
42. B. Nasersharif and A. Akbari, "Sub-band weighted projection measure for sub-band speech recognition in noise," *Electron. Lett.* **42**(14), 829–831 (2006).
43. C. Cerisara, "Automatic discovery of topics and acoustic morphemes from speech," *Comput. Speech Lang.* **23**(2), 220–239 (2009).
44. L. M. Lee, "Adaptation of hidden Markov models for half frame rate observations," *Electron. Lett.* **46**(10), 723–724 (2010).
45. R. Marín, P. Vila, P. J. Sanz and A. Marzal, "Automatic Speech Recognition for Teleoperate a Robot Via Web," *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems EPFL*, Laussane, Switzerland (2002) pp. 1278–1283.
46. P. Mayorga, J. Martín, A. M. Hernández and J. Flores, "Gaussian Components Optimization for a Robot Controlled by Speech Commands in Mexican Spanish," *4th Congress of Electronics, Robotics and Automotive Mechanics (CERMA 2007)*, Cuernavaca, Mexico (2007) pp. 124–129.

https://doi.org/10.1017/S0263574714000010 Published online by Cambridge University Press