DOI: 10.17469/O2104AISV000015

ROBERTO GRETTER, MAURIZIO OMOLOGO,
LUCA CRISTOFORETTI, PIERGIORGIO SVAIZER

A vocal interface to control a mobile robot

A multi-modal interface has been integrated on a moving robotic platform, which allows the user to interact at distance, through voice and gestures. The platform includes a microphone array, whose processing provides speaker localization as well as an enhanced signal acquisition. A multi-modal dialogue management is combined with a traditional HMM-based ASR technology, in order to give the user the possibility to interact with the robot in different possible ways, e.g., for platform navigation purposes. The system is always-listening, it operates in real-time, and has been tested in different environments. A corpus of dialogues was collected while using the resulting platform in an apartment. Experimental results show that performance are quite satisfactory in terms of both recognition and understanding rates, even when the user is at a distance of some meters from the robot.

Keywords: multi-microphone signal processing, multi-modal interfaces, distant-speech recognition, spoken dialogue management, human-robot interaction.

1. *Introduction*

In human-machine communication, one of the most common dreams is about talking with a robot. In spite of recent important advances in automatic speech recognition (ASR), primarily obtained thanks to a corresponding tremendous progress in deep learning (Goodfellow, Bengio & Courville, 2016; Yu, Deng, 2015), most of the existing robotic technologies do not have the capability of conducting a human-like voice interaction (Markovitz, 2015). A fundamental problem is that in most of the cases speech recognition and spoken dialogue are treated as plug-in technologies. Speech processing and related feedback mechanisms are not part of a fully integrated framework, that manages them in a coherent manner, together with other sensing (e.g., vision, touch), sensorimotor channels, knowledge and "world" representation. Moreover, this framework is often disconnected by all the cognitive processes that are needed to obtain a fully autonomous and effective solution.

This paper describes our recent activities towards the development of a robotic platform able to interact by voice thanks to an integrated framework that includes a multi-microphone input device, a Kinect device, an Arduino based GUI interface, and a low-cost moving robotic platform.

The main focus of our work was to realize a spoken dialogue management and, inherently related to it, a scheduler to real-time process in a coherent way the various information captured by the available sensing platforms, and execute different kinds of actions, such as platform movements, navigation, voice feedback, and GUI

output. As outlined in the following, the system can access and modify its internal setup (e.g. speed), can handle low- and mid-level navigation commands, can manage an agenda, learn new information.

At this moment, the platform does not include cameras and related video processing, which would further increase the possible functionalities and skill, for a deeper understanding of the surrounding scene. Nevertheless, the robotic platform performs a lot of possible actions in a highly multimodal fashion, based on the interpretation of the acoustic scene as well as of what can be deduced from Kinect, and other navigation related devices.

An important aspect to highlight is also related to hardware: the system runs on processing units available on the platform itself. It does not exploit any connection to external computers or to remote services (e.g., such as those possible through Amazon Alexa, GoogleHome, Siri, etc.). In other words, it can be classified as a "very-low" cost solution, fully relying on off-the-shelf devices and on-board computing platforms.

Finally, it is worth mentioning that our work also aims to show how effective a robot-control can be just based on a mix of voice and gestures, with few linguistic restrictions, and from a reasonable distance (typically up to 4-5 meters) of the user from the platform. Indeed, an always-listening distant-speech interface (Wolfel, McDonough, 2009) gives a lot of flexibility to the user in real-time controlling the robot. On the other hand, robust speech preprocessing (Vincent, Virtanen & Gannot, 2018) and ASR technologies are necessary, in order to tackle environmental noise, reverberation, as well as the mechanical noise produced by the platform itself, when it moves.

The remainder of this paper is organized as follows. Section 2 provides an overview of the system, including the architecture and the user interface. Section 3 describes the corpus that was collected and the related experimental set-up that was used to develop and test the system. Section 4 provides some experimental results, while Section 5 draws some conclusions and outlines the envisaged future work.

2. *System overview*

2.1 Architecture

Our robot is based on the TurtleBot2[1] platform. The basic structure is a Kobuki moving base, a Microsoft Kinect device and an entry-level laptop (eventually replaced by an Odroid XU4 in the most recent version). The Kobuki base is equipped with two motors, bumpers, encoders on the wheels and a gyroscope. To this standard setup we added eight digital MEMS microphones, some LEDs and an Arduino-based LCD screen. The robot can be seen in Figure 1.

[1] https://www.turtlebot.com/turtlebot2.

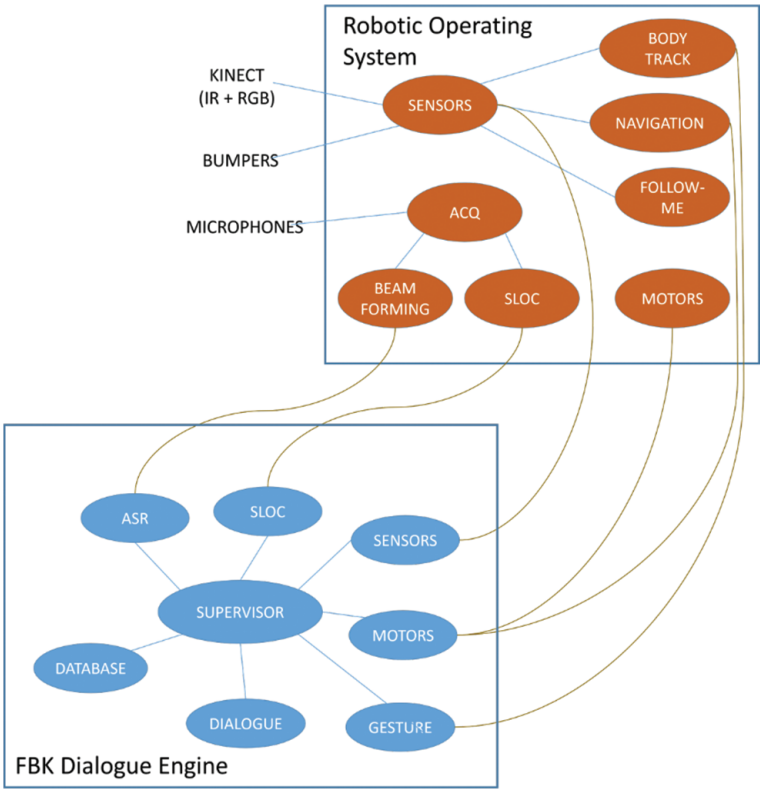
Figure 1 - *The robot equipped with all the installed devices*

The software architecture is based on ROS[2], an open source platform for Ubuntu that helps in building robot applications. The main concept is based on software nodes that collaborate, supervised by a master. Nodes can also run on different machines to obtain a distributed architecture.

We added new nodes to handle multichannel audio acquisition, beamforming, sound source localization, Arduino LCD. In addition, other specific nodes interact with the speech recognizer and the dialogue framework. Figure 2 depicts the different ROS nodes, connected to physical devices and connected to the dialogue environment modules, described later.

[2] http://www.ros.org.

Figure 2 - *ROS nodes and dialogue modules and their interaction*



2.2 User interaction and feedback

The interaction between the user and the robot is multimodal: it can be based on speech or gestures, or both. Gestures are captured by the Microsoft Kinect installed on the robot: publicly available Kinect libraries provide a skeleton representation of the user in front of the device. This representation is processed to detect the user position and arms movements, a feedback is provided by an LCD display. The display on top of the robot represents a simple face, whose eyes track the user's head position. Nose turns red when the user moves the right hand to stop the robot, while mouth corners turn red to indicate the estimation of the direction pointed by the user. Around the display, eight red LEDs turn on to indicate the direction from which the acoustic wave is impinging the microphone array.

Speech is captured by eight digital MEMS microphones installed on a plate on top of the robot. They are distributed around the borders to have a 360-degree coverage. More details are provided in the following.

2.3 Multi-microphone signal processing

In the ideal scenario, the robot should always be ready to detect, decode and understand questions or commands uttered by a potential user and addressed to it. For this reason,

sound acquisition must always be active, such that the robot can listen continuously at what is happening in its surroundings. In practice, as we did not implement barge-in capabilities yet, the robot is always listening except when it is speaking: this condition is notified by means of a LED that becomes red, while it is green when the user is allowed to speak to the robot.

The availability of multiple microphones (8 sensors distributed in a circular pattern, with higher concentration in the frontal direction) enables the processing of the acquired signals not only by means of a time/frequency analysis, but also taking into account the spatial characteristics of the arriving acoustic waves. In particular, by evaluating global coherence field (GCF)-based acoustic maps related to the surrounding space (De Mori, 1997; Knapp, Carter, 1976), the robot is able to derive sound direction and also to estimate the distance of the source.

Once the mutual delays on the eight channels are compensated and the corresponding signals are summed up (i.e. by means of a delay-and-sum beamforming operation) the robot is also able to perform a sort of spatially-selective listening (Brandstein, Ward, 2001). The main goal is to detect human speech and isolate potentially interesting utterances.

The speech/non-speech classifier currently implemented on the platform is rather basic, as it uses only simple features: signal dynamics and a periodicity index denoting the presence of pitch within the frequency range typical of human speech. In rather quiet conditions this is already sufficient to provide a satisfactory voice activity detection (VAD). Experiments in more adverse acoustic conditions show that the use additional acoustic features such as the Mel-frequency cepstral coefficients (MFCCs) and the inter-channel spatial coherence yield improved performance especially under medium-low SNR conditions (Armani, Matassoni, Omologo & Svaizer, 2003).

An exhaustive evaluation of the VAD is outside the scope of this paper. Nevertheless, the VAD is a delicate part of the whole chain: in particular, it suffers from the noise produced during movement by the electrical motors, when the robot is moving at high speed: in these cases, the system is usually able to discard false alarms due to the constraint that each valid command must begin with a keyword ("robottino"), as will be explained later.

2.4 Acoustic models

The beamformed signal represents the input to the next stage of the speech decoding process.

The acoustic features are the traditional 13-dimensional MFCCs, augmented by first and second order temporal derivatives, and mean-normalized. The front-end processing is applied on 25 ms length windows, with a 10 ms shift. A standard HMM based recognition system is built, based on the FBK's ASR technology developed during the past years, and already used in other application fields (Brutti, Coletti, Cristoforetti, Geutner, Giacomini, Maistrello, Matassoni, Omologo, Steffens & Svaizer, 2005; Sosi, Ravanelli, Matassoni, Cristoforetti, Omologo & Ramella, 2014): fifty Italian monophone models are trained using phone segmentation, obtained automatically and manually checked. Around 1000 tied-state context-dependent triphones are then derived (with ~14000

Gaussians). For state tying, a standard decision tree based on specific phonetic questions for the Italian language is employed. Acoustic models are trained on a dataset composed of Apasci clean (Angelini, Brugnara, Falavigna, Giuliani, Gretter & Omologo, 1994), Apasci reverberated (Ravanelli, Sosi, Svaizer & Omologo, 2012), and part of DIRHA database (Cristoforetti, Ravanelli, Omologo, Sosi, Abad, Hagmüller & Maragos, 2014). The total duration of the speech dataset is about 7 hours and 23 minutes. The final model is obtained after a MAP adaptation stage with a limited amount (25 minutes) of in-domain recordings.

2.5 Finite state grammars

The language model used to perform ASR is a set of Finite State Networks (FSNs), where each transition can be either a word or the link to another FSN. In this way, the resulting grammar is in fact a context free one (De Mori, 1997).

Output labels can be assigned to both words and links, so the output of the ASR results to be a parse tree of the sentence. In order to reduce the risk of recognizing generic sentences as commands, valid commands have to be preceded by the name of the robot, which is *Robottino*.

Each FSN can be designed following different criteria: some of them, representing well defined sublanguages like numbers or time expressions, were designed by hand by means of regular expressions. Some others can be lists of words or phrases, while often N-grams are used to represent for instance the surface of the sentences (Falavigna, Gretter & Orlandi, 2000).

In principle, each dialogue state could activate a specific grammar, but in practice this possibility is used only to favour expected default data, like assigning to the utterance "25" the label *degree* after the question "give me the rotation angle". In this application, a set of about 170 FSNs is used, with a global lexicon composed of about 6000 words.

2.6 Dialogue

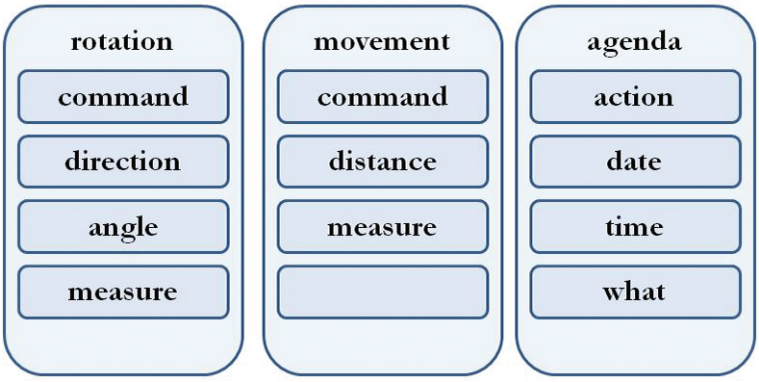
FBK built a dialogue engine in the past (Falavigna, Gretter, 1999; Giorgino, Azzini, Rognoni, Quaglini, Stefanelli, Gretter & Falavigna, 2005), which was used in several European projects (e.g., C-Oral-Rom, Homey, Vico, Dirha)[3] and commercial prototypes and systems. It is able to handle system- and mixed-initiative dialogues, and can cope with relatively complex sentences in natural language in limited domains. The dialogue component implemented in the actual version of the FBK robot is composed of the following main parts:

- engine, written in Perl (also a version implemented in C is available), which is basically an interpreter of a description, written in a Perl-like proprietary language and compiled in a Perl data structure;

[3] C-Oral-Rom (http://cordis.europa.eu/project/rcn/60720_en.html);
Homey (http://cordis.europa.eu/project/rcn/61013_en.html);
Vico (http://cordis.europa.eu/project/rcn/60714_en.html);
DIRHA (http://dirha.fbk.eu).

- description of the dialogue, which consists in:
  - a dialogue strategy – a number of structures and procedures common to many applications, that implement the philosophy of the dialogue – very briefly, there is a main loop in which the status of the dialogue is analyzed in order to decide which is the next move to do. A number of semantic concepts are organized into contexts, each one corresponding to a subdomain: the dialogue strategy has to cope with all of them, being able to change context depending on the user input;
  - a number of tools that can be easily included in the application, to handle common concepts like numbers, dates, confirmations, etc.;
  - an application-dependent part, where the semantic data necessary to model the desired domains are defined, together with their dedicated procedures. In this project, some of the contexts and concepts defined for the navigation and for the agenda are shown in Figure 3.

Figure 3 - *Some concepts related to navigation and agenda, that are normally filled by voice. They are grouped into contexts, that can be considered as subdomains. All subdomains can be activated at any time without the need for an explicit switch*

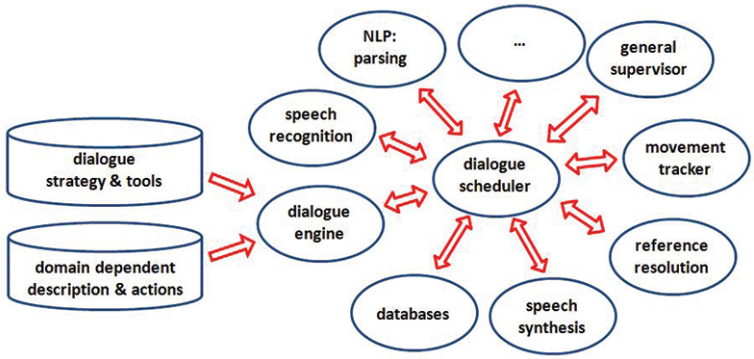


The dialogue module communicates through a scheduler – responsible only to exchange messages among processes – with several external modules, each one dedicated to some precise task. The most important processes for this application, depicted in Figure 4, are:

- speech acquisition: it gets the audio corresponding to a sentence by the user, at present in a synchronous way, i.e. the user can speak only after the prompt given by the robot. This process can be simple – just one audio channel, for instance for phone applications – or complex – in the case of the robot, 8 microphones located on the robot itself: the eight speech signals are properly processed in order to obtain a unique, enhanced, speech signal. Note that also a text input mode is possible;
- ASR + parsing: the former processes the input speech in order to get the word sequence, the latter produces a parse tree of the sentence. There must be an alignment between the labels of the parse tree and the labels of the semantic concepts defined in the dialogue description;

- speech synthesis: has to produce a synthetic speech output, starting from the word sequence produced by the dialogue;
- database: has to collect information from the "external world", for instance labels and positions recorded in past interactions, a personal agenda, etc.
- reference resolution: it keeps track of the objects that are nominated during the dialogue and in case of pronouns, tries to find out the most probable object (how much is your speed? ... augment it by 0.2 ... bring it to 0.7...), taking into account both properties of the objects (speed can be augmented but not cancelled) and distance in time (objects nominated recently are more likely to be selected).
- robot modules: there are several processes which perform speaker localization, return or set the values of some parameters (translation and rotation speed, battery level, etc.), execute navigation commands that could be simple (go forward 1 meter, turn right 30 degrees) or complex (follow me, perform autonomous navigation to reach coordinates XY), etc.
- movement tracker: based on the skeletal tracking capability of the Kinect, the robot can track the movements of the users in front of it. Some gestures with arms are recognized and used as additional directives for the dialogue manager that, besides spoken commands, is able to handle multimodal input.

Figure 4 - *Architecture of the processes connected with the dialogue. New processes can be easily added to the architecture and each component can be easily replaced with an improved version*



Newly added features to the dialogue concern the interaction with the last three modules, namely reference resolution, robot modules and movement tracker. Further, due to the characteristics of the application, the dialogue engine was updated to address the following issues:

- compound commands: the system is now able to get multiple commands (go forward 1 meter, turn left 30 degrees and then go on for 2 meters) and to properly execute them in the right order, waiting for the formers to be completed before starting the next one.
- multimodality: the system is able to get input from speech only, from gesture only (equivalent to the command "*stop*") or from a combination of the two (the command "*go there*", indicating a direction with the arm). It is also capable to detect the posi-

tion of the speaker from his voice, by exploiting the delay with which it arrives to the eight microphones. This feature is essential to be able to properly react to the command *"come here"* even if the speaker is behind the robot. At present, the two channels (speech and gesture) are completely independent and their fusion only requires a synchronization in time (i.e. the speech command and the gesture recognition must be detected within a couple of seconds, otherwise they will be ignored). Gesture processing is always active.

2.7 Language recognized by the robot

The language that the robot can understand covers some subdomains, each modeled by hand-defined grammars that allow to express in natural language the possible commands, with relatively complex sentences in a variety of ways. When a command is incomplete, a mixed-initiative dialogue helps to get the missing parameters. The most important command types are:

- basic navigation commands ("*go forward one meter and a half*"; "*turn right 90 degrees*"; *"go backward 50 centimeters"*; *"stop"*);
- multimodal commands, which merge spoken commands with information coming from various sensors (gesture detection using Kinect, speaker localization via audio processing) to be properly executed:
  - ***come here*** needs to localize the position of the speaker using his voice – he could be behind the robot;
  - ***go there*** + ***gesture*** combines speech and gesture, seen by the robot via Kinect – user must be in front of the robot;
  - ***follow me*** combines the two: based on localization via speech, first the robot will move to the speaker and then will use Kinect to follow him;
- information commands and self-awareness: the system can answer questions about the capabilities of the robot; the system can access and change some of its internal parameters (*"which is the value of your speed?"*; *"bring it to 1.0"*; *"set angular speed to 2"*; *"what can you do?"*; *"shut up"*);
- teaching commands: to provide new information in a dynamic way. The user can teach the robot a new position ("*learn, this is the entrance door"*) that will be memorized and that could be immediately used to drive the robot in that position;
- mid level navigation commands: ("*go to the entrance door"*; "*go home"*; *"go to the office of Maurizio"*) that use ROS primitives to perform autonomous navigation in order to reach the coordinate X,Y associated with the given label;
- compound commands: the system can handle a list of commands, for instance up to four navigation commands that will be executed in sequence (turn right 90 degrees, go forward one meter, turn 30 degrees to the left);
- agenda: the user can save in the agenda a new appointment, or ask for the appointments previously saved. Each appointment needs a date, a time slot, a description to be complete (*"set a new appointment for next Tuesday: phone call with John at 3PM"*; "*tell me what I have to do this afternoon"*).

The system is also able to solve pronoun resolution, useful in dialogues like:

User: *robot, how much is your speed?*
Robot: *speed is 0.4.*
User: *raise **it** by 0.2.*
Robot: *changing speed from 0.4 to 0.6.*

3. *Experimental set-up*

3.1 Corpus of dialogues

Data for the evaluation of the system were acquired in the ITEA Apartment, described below. Seven subjects performed interactions with the robot in order to execute some predefined tasks, designed in order to collect data covering most of the operations that *Robottino* could perform. Two of the subjects were involved in the design of the vocal interface of the system, the others were researchers working on other topics. Each subject first read a page describing the main features of the robot and reporting some sample commands, then performed the recordings in three sessions. He/she had to drive the robot to execute 14 tasks in different conditions (close to the robot or not, turned toward the robot or not); about 30 minutes were needed on average to perform all the tasks (min 21, max 40). Some examples of the tasks are reported in Table 1. The entire speech data sequences (total duration 3h 19m) were acquired and processed to get time markers and ASR output.

Manual correction was then performed to obtain a reliable transcription. Also a semantic representation was automatically derived and manually checked from the transcriptions. Only the pure speech commands were considered; we decided to leave evaluation of gesture and mixed commands to future works. Some samples are in Table 2.

Table 1 - *Some examples of the tasks to be executed. Each task was performed in the indicated conditions: speaker close to Robottino or not – respectively, less than 2 meters or more than 2.5 meters; talking towards Robottino or turned on the other side*

Task description	*Conditions*
Teach Robottino where the living room window is located (you must first bring it to the desired position and then tell it to memorize the position).	Robottino starts from the center of the living room. Hand-clap. Speaker close, talks towards Robottino.
Make Robottino run a 8-path between two chairs, with basic movement commands.	Speaker away, talks towards Robottino.
Ask Robottino for Friday's appointments.	Speaker close, talks to the other side.
Ask Robottino for the speed value and then decrease it by 0.1.	Speaker away, talks to the other side.
Set up a new appointment: Saturday morning at 10.00 am, trip to the museum with Paolo and Alessio.	Speaker close, talks towards Robottino.

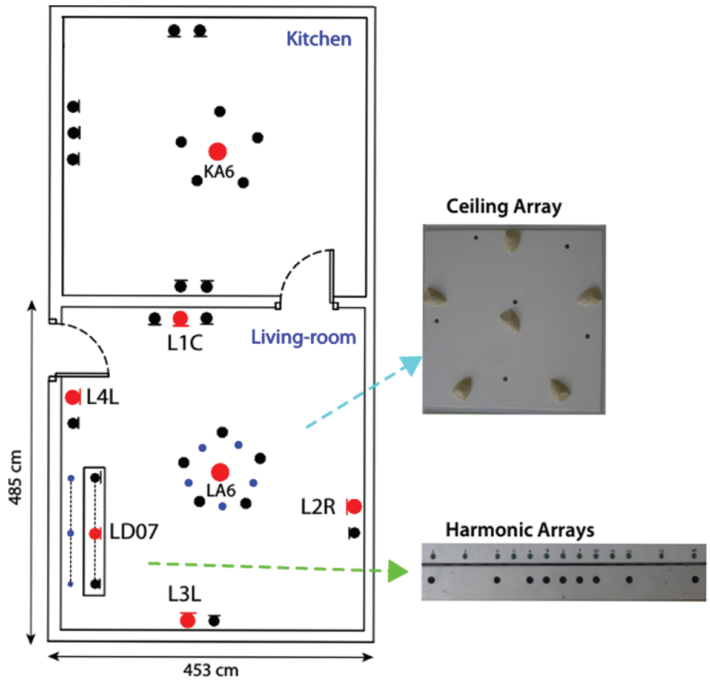
Table 2 - *Some examples of transcriptions of original sentences (in Italian) with the corresponding semantic representations*

robottino vai avanti di ottanta centimetri (FORW_CMD(vaiavanti)FORW_CMD) (FORW_DIST(ottanta)FORW_DIST) (FORW_MIS(centimetri)FORW_MIS)
robottino ruota a sinistra di novanta gradi (ROT_CMD(ruota)ROT_CMD) (ROT_DIR(asinistra)ROT_DIR) (ROT_ANG(novanta)ROT_ANG) (ROT_MIS(gradi)ROT_MIS)
aumenta= =la di zero punto due (PRN_VRB(aumenta=)PRN_VRB) (PRN_CLT(=la)PRN_CLT) (PRN_VAL(di(NUM3(zero)NUM3)punto(NUM3(due)NUM3))PRN_VAL)
robottino dimmi gli appuntamenti di domenica prossima (AG_ACTION(dimmigliappuntamenti)AG_ACTION) (AG_DATE((WEEK(domenica)WEEK)prossima)AG_DATE)

3.2 The microphone network

A data collection took place in an apartment in Trento, called ITEA Apartment. The flat comprises five rooms which are equipped with a network of several microphones. Most of them are high-quality omnidirectional microphones (Shure MX391/O), connected to multichannel clocked pre-amp and A/D boards (RME Octamic II).

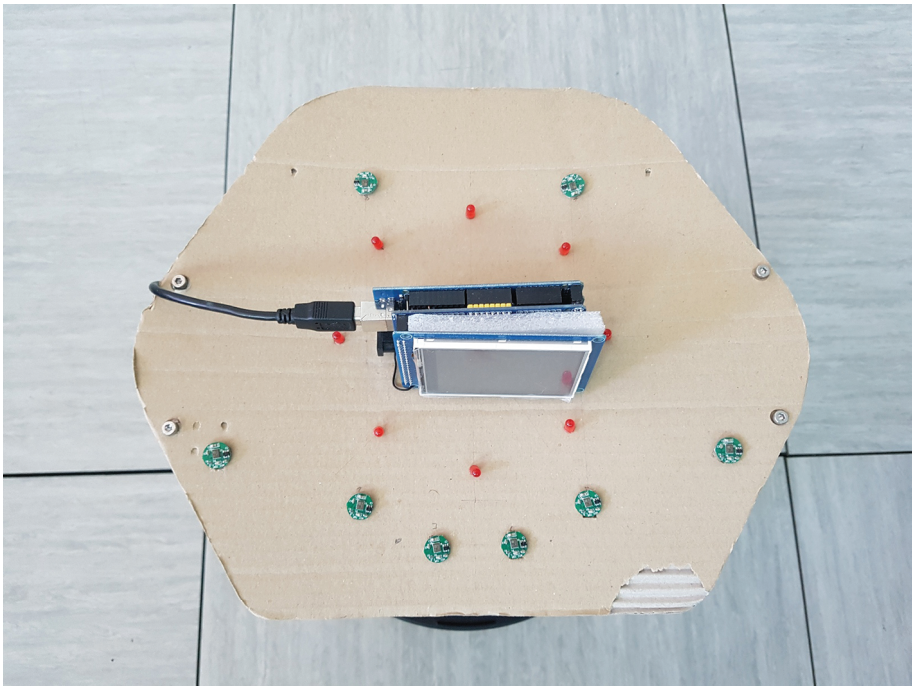
Figure 5 - *Details of the microphone network installed in the kitchen and in the living-room*



The bathroom and two other rooms were equipped with a limited number of microphone pairs and triplets (i.e., overall 12 microphones), while the living-room and the kitchen comprise the largest concentration of sensors and devices. As shown in Figure 5, the living-room includes three microphone pairs, a microphone triplet, two 6-microphone ceiling arrays (one consisting of MEMS digital microphones), two harmonic arrays (consisting of 13 electret microphones and 16 MEMS digital microphones, respectively).

Eight digital MEMS microphones are installed on the top plate of the robot (see Figure 6). Their polar pattern is omnidirectional and they are distributed on both front and back sides to acquire sound coming from all directions. Sampling rate is 16 kHz with 16 bit resolution.

Figure 6 - *Robot top plate, with MEMS microphones and LEDs*



Along with audio recordings, the interactions have been video-recorded with a digital camera. Audio-video synchronization is obtained with a hand-clap.

The experimental dataset was collected under realistic conditions, i.e. with the robotic platform moving inside this apartment. Note that it was not possible to synchronize at sample level the acquisition boards devoted to different sets of microphones, which operated at the same nominal sampling rate but with independent clocks, leading to possible time drift. Therefore, only a rough synchronization was achieved between the robot, the distributed microphones and the video recordings.

This misalignment was then compensated off-line for each utterance by means of a method based on GCC-PHAT (Knapp et al., 1976).

Beamformed signals were used to accomplish the manual segmentation and annotation of the dataset, which served then as reference in order to evaluate the whole speech interaction system.

4. *Evaluation/Results*

To evaluate the system from the ASR point of view we considered the beamformed signal derived from the acquisitions done by the robot's array in the ITEA apartment. From the whole corpus we extracted only the segments, manually checked, corresponding to sentences pronounced by the users, without considering sentences in which also *Robottino* was speaking. In total we have 950 utterances, amounting to 33:07 minutes of speech. This material was divided into development and test set, composed of 254 (09:08 minutes, development) and 696 (23:58 minutes, test) utterances, respectively.

Results are reported both in terms of Word Accuracy (WA) by considering words as units, and in terms of Semantic Accuracy (SA). In this case, and for ASR evaluation only, we consider as a semantic unit all the words and the semantic labels used: even a small error in a functional word will cause the semantic unit to be considered wrong, like the following two examples:

(AG_DATE((WEEK(domenica)WEEK)prossima)AG_DATE)
(AG_DATE((WEEK(domenica)WEEK)prossimo)AG_DATE)

(AG_WHAT(gitainmontagnaconMirco)AG_WHAT)
(AG_WHAT(montagnaconMirco)AG_WHAT)

Table 3 reports results in terms of Word and Semantic Accuracy for development and test set.

Table 3 - *Results both in terms of Word Accuracy (WA) and Semantic Accuracy (SA). Also the total number of units as well as deletions, insertions and substitutions are reported*

	Words					*Semantic Units*				
	WA	#Units	Del	Ins	Sub	SA	#Units	Del	Ins	Sub
dev	77.96%	1384	88	37	180	81.55%	542	55	9	36
test	81.73%	3366	219	97	299	85.28%	1427	71	36	103

To further understand the distribution of the errors, we joined development and test set and rearranged the whole corpus following the acquisition conditions:

- **F** vs **N**: Far (>2.5 meters) vs Near (<2 meters);
- **GO** vs **BO**: GoodOrientation (user speaks turned toward Robottino) vs BadOrientation (user speaks turned opposite);

- **NH** vs **NL**: Noise produced by *Robottino* when moving (High vs Low: the amount of noise depends on the speed of Robottino).

Table 4 reports results for the different acquisition conditions. Despite the fact that in some cases (in particular N-BO and NH) the number of utterances is too low to have statistical significance, a clear trend can be observed. The distance from the robot seems not to be a critical issue for comprehension when the orientation is good: SA drops only from 86.67% (N-GO) to 85.90% (F-GO), despite a bigger drop in WA (84.76% to 79.70%) which involves insertion and deletions of semantically irrelevant words. Much more critical seems to be the drop in orientation: 85.90% drops to 58.90% when far, 86.67% drops to 68.57% – a bit less severe – when near the robot. This is probably due to the fact that the voice impinges the microphones with less energy and there are much more reflections, which also determines lower quality of the resulting beamformed signals. Note that, for the very few sentences recorded in the NH condition, we got a really low WA (0.00%), mainly due to insertion and deletion of semantically irrelevant words; still the SA remained at an acceptable level (62.50%).

Table 4 - *Recognition results under different conditions*

	Words					*Semantic Units*				
	WA	#Units	Del	Ins	Sub	SA	#Units	Del	Ins	Sub
F-BO	57.93%	435	26	20	137	58.90%	163	52	0	15
F-GO	79.70%	2803	228	73	268	85.90%	1156	79	14	70
N-BO	71.23%	73	0	7	14	68.57%	35	6	1	4
N-GO	84.76%	1450	74	45	102	86.67%	615	16	19	47
NH	00.00%	15	0	8	7	62.50%	8	1	1	1
NL	80.05%	4746	325	133	489	84.29%	1961	143	31	134

Finally, it is interesting to report a rough indication of the performance that can be achieved by this system, in terms of speaker localization performance. Given the aforementioned on-board microphone array geometry, and the state-of-the-art techniques that were embedded in our solution, estimating the direction from which the user is speaking is relatively easy, when no other speakers or noise sources are active. In this case, we observe an average error of less than 5 degrees in the azimuth angle estimation, which normally leads to a quite accurate activation of the LED in the direction of the user.

On the other hand, a more challenging task is the depth estimation, i.e., the distance of the user from the robot, which is more difficult to evaluate due to the geometry of the array (all the microphones are concentrated in a rather restricted area on the top of the platform). The average error in depth estimation is typically of about 50 cm - 100 cm, though this strongly depends on some factors, including how

loud the command was uttered by the speaker, and how her/his head was oriented (i.e., facing the array increases the chance of obtaining a satisfactory localization).

5. *Conclusions and future work*

We described recent activities conducted by Fondazione Bruno Kessler for the development of a multi-modal robotic platform, which integrates a multi-microphone input device, a Kinect device, and a Kobuki moving base, and realizes different functionalities relying on audio and gesture input.

Spoken dialogues of variable complexity can be realized, related to robot's self-awareness information, to management of a user's agenda, to execution of commands for platform movements, as well as to navigation in the surrounding space.

A corpus has also been collected, based on user-robot interactions in a real-environment, in order to evaluate system performance.

Experimental results on this corpus and on-field sessions showed that in most of the cases the word accuracy of 70-80% is reached, which corresponds to more than 80% semantic accuracy. This performance allows the user to interact with the robot in a smooth and effective way, even when he/she is rather distant from the robot. Although some performance loss was observed when the user did not speak in the direction of the robot, this situation does not seem to be critical, since interacting with a robot always induces one, if possible, to face it (and its GUI interface).

It is also worth mentioning that the presented solution is characterized by a very low complexity, which allowed a low-cost implementation that does not require any connection with external computing platforms, clusters, or cloud-based services.

Future developments could take different directions.

A first one is based on the integration of a camera and related video processing, in order to take advantages of the complementarity between audio and video modalities, for the introduction of additional functionalities as well as for a more robust user-robot interaction performance. A first example is showed in (Qian, Xompero, Brutti, Lanz, Omologo & Cavallaro, 2018), which investigated on the joint use of audio and video input for 3D person tracking.

Concerning multimodal dialogue management, we also envisage several possible evolutions of this work.

For instance, the system is capable to handle multiple dialogues at a time: this feature was implemented for instance in the DIRHA system where in each room a different dialogue was running, being able to serve different people at the same time. This could be useful in future to allow different users to interact with the robot and, in case of audio/video person recognition, access personal information (agenda, preferences, etc.). Recently, a web interface has been implemented to be able to use the dialogue potentially from any device connected to Internet. This feature, in addition to some webcam, could allow to control the robot remotely, or in general to use the dialogue to access information from anywhere. The dialogue architecture is modular, in the sense that it is quite easy to add new processes to the

system (recently: robot modules, pronoun resolution, multimodality, etc.): this will easily allow to introduce new processes that, maybe exploring the characteristics of the environment, give suggestions to the dialogue about some move to do to improve the interaction with the user (for instance, when detecting a noise source, trying to do something to reduce it – close a window – or proposing to go near the speaker to get a better signal to noise ratio).

It is also worth noting that the collected corpus will give us the chance to explore another possible approach, which is based on integrating all the available microphone signals for speaker localization as well as for speech recognition purposes. This perspective can eventually be related to very attractive and novel application contexts, e.g., when the user is very distant from the platform, or is not facing it, a collaborative processing based on both the on-board array and the distributed microphone network will increase the overall system's robustness, i.e., robot behaviour.

Finally, the adoption of deep learning together with distributed processing will be explored for instance based on the recent paradigm introduced in (Ravanelli, Brakel, Omologo & Bengio, 2017) in order to realize an on-board small footprint deep learning based solution.

Acknowledgment

We would like to thank our colleagues Alessio Brutti, Marco Matassoni, Marco Pellin, Mirco Ravanelli and Alessandro Sosi, for various contributions provided during the realization of the described robotic platform.

Bibliography

Angelini, B., Brugnara, F., Falavigna, D., Giuliani, D., Gretter, R. & Omologo, M. (1994). Speaker Independent Continuous Speech Recognition Using an Acoustic-phonetic Italian Corpus. *Proceedings of the Third International Conference on Spoken Language Processing (ICSLP 94)*, Yokohama, Japan, September 18-22, 1994, 1391-1394.

Armani, L., Matassoni, M., Omologo, M. & Svaizer, P. (2003). Use of a CSP-Based Voice Activity Detector for Distant-Talking ASR. *Proceedings of the 8th European Conference on Speech Communication and Technology (EUROSPEECH 2003 - INTERSPEECH 2003)*, Geneva, Switzerland, September 1-4, 2003, 501-504.

Brandstein, M., Ward, D. (2001). *Microphone Arrays: Signal Processing Techniques and Applications*. Berlin: Springer.

Brutti, A., Coletti, P., Cristoforetti, L., Geutner, P., Giacomini, A., Maistrello, M., Matassoni, M., Omologo, M., Steffens, F. & Svaizer, P. (2005). Use of Multiple Speech Recognition Units in a In-car Assistance System. In Hüseyin, A. Hansen, J. & Takeda, K. (Eds.), *DSP for In-Vehicle and Mobile Systems*. Berlin: Springer, 97-111.

Cristoforetti, L., Ravanelli, M., Omologo, M., Sosi, A., Abad, A., Hagmüller, M. & Maragos, P. (2014). The DIRHA Simulated Corpus. *Ninth International*

Conference on Language Resources and Evaluation (LREC 2014), Reykjavik, Iceland, *May 26-31*, 2014, 2629-2634.

De Mori, R. (1997). *Spoken Dialogues with Computers*. Cambridge, MA, Academic Press, Inc.

Falavigna, D., Gretter, R. (1999). Flexible Mixed Initiative Dialogue over the Telephone Network. *Proceedings of the Workshop on Automatic Speech Recognition and Understanding (ASRU99)*, December 12-15, 1999, Keystone, Colorado, USA, 12-15.

Falavigna, D., Gretter, R. & Orlandi, M. (2000). A Mixed Language Model for a Dialogue System over the Telephone. *Proceedings of 6th International Conference on SpokenLanguage Processing (ICSLP 2000)*, October 16, 2000, Beijing, China, 585-588.

Giorgino, T., Azzini, I., Rognoni, C., Quaglini, S., Stefanelli, M., Gretter, R. & Falavigna, D. (2005). Automated Spoken Dialogue System for Hypertensive Patient Home Management. *International Journal of Medical Informatics*, 74(2):159-167.

Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep learning*. Cambridge, MA, MIT Press. http://www.deeplearningbook.org.

Knapp, C., Carter, G., (1976). The Generalized Correlation Method for Estimation of Time Delay. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(4), 320-327.

Markowitz, J. (2015). *Robots that Talk and Listen*. Berlin: Walter de Gruyter.

Ravanelli, M., Sosi, A., Svaizer, P. & Omologo, M. (2012). Impulse Response Estimation for Robust Speech Recognition in a Reverberant Environment. *Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, August 27-31, 2012, Bucharest, Romania.

Ravanelli, M., Brakel, P., Omologo, M. & Bengio, Y. (2017). A Network of Deep Neural networks for Distant Speech Recognition. *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2017)*, March 5-9, 2017. New Orleans, USA, 172-176.

Sosi, A., Ravanelli, M., Matassoni, M., Cristoforetti, L., Omologo, M. & Ramella, S. (2014). Interazione vocale a distanza in ambiente domestico. In Romano, A., Rivoira, M. & Meandri I. (Eds.), *Aspetti prosodici e testuali del raccontare: dalla letteratura orale al parlato dei media*. Alessandria: Edizioni dell'Orso, 2015.

Qian, X., Xompero, A., Brutti, A., Lanz, O., Omologo, M. & Cavallaro, A. (2018). 3D Mouth Tracking from a Compact Microphone Array Co-located with a Camera. *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2018)*, April 15-20, 2018. Calgary, Canada, 3071-3075.

Vincent, E., Virtanen, T. & Gannot, S., Eds. (2018). *Audio Source Separation and Speech Enhancement*. Hoboken, NJ: Wiley.

Wolfel, M., McDonough, J. (2009). *Distant Speech Recognition*. Hoboken, NJ: Wiley.

Yu, D., Deng, L. (2015). *Automatic Speech Recognition - A Deep Learning Approach*. Berlin: Springer.