Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.
Digital Object Identifier XX.XXXX/YYY.20XX.ZZZZZZZ

Design and Evaluation of a Voice-Controlled Elevator System to Improve Safety and Accessibility

ANDER GONZÁLEZ-DOCASAL[1, 5], JON ALONSO[2], JON OLAIZOLA[3], MIKEL MENDICUTE[3] (Senior Member, IEEE), MARÍA PATRICIA FRANCO[4], ARANTZA DEL POZO[1], DANIEL AGUINAGA[2], AITOR ÁLVAREZ[1], EDUARDO LLEIDA[5]

[1]Fundación Vicomtech, Basque Research and Technology Alliance (BRTA), Mikeletegi 57, 20009 Donostia-San Sebastián, Spain
[2]IKOR, Sistemas Electrónicos S.L., Parque Empresarial Zuatzu, Francisco Grandmontagne Kalea, 4, 20018 Donostia-San Sebastian, Spain
[3]Electronics & Computer Science Department, Mondragon University, 20500 Arrasate-Mondragon, Spain
[4]Orona Corporación, Orona Ideo, Jauregi Bidea s/n, 20120 Hernani, Spain
[5]Aragon Institute for Engineering Research, University of Zaragoza, 50009 Zaragoza, Spain

Corresponding author: Ander González-Docasal (e-mail: agonzalezd@vicomtech.org).

This work was supported by the Basque Government's Elkartek research and innovation program, through the iVOZ (KK-2021/00038) project.

ABSTRACT

This work introduces the design and assessment of a voice-controlled elevator system aimed at facilitating touchless interaction between users and hardware, thereby minimising contact and improving accessibility for individuals with disabilities. The research distinguishes three distinct deployment scenarios – on cloud, on edge and embedded – with the ultimate goal of integrating the entire system into a low-resource environment on a custom carrier board. An objective evaluation measured acoustic conditions rigorously using a dataset of 2900 audio files recorded inside a laboratory elevator cabin featuring two internal coatings, five audio input devices, and under four distinct noise conditions. The study evaluated the performance of two Automatic Speech Recognition systems: Google's Speech-to-Text API and a Kaldi model adapted for this task, deployed using Vosk. Additionally, latency times for these transcribers and two communication protocols were measured to enhance efficiency. Finally, two subjective evaluations on clean and noisy conditions were conducted simulating a real world scenario. The results, yielding 84.7 and 77.2 points respectively in a System Usability Scale questionnaire, affirm the reliability of the presented prototype for industrial deployment.

INDEX TERMS Automatic speech recognition, embedded systems, human-machine interaction

I. INTRODUCTION

In an era marked by a growing emphasis on touchless interaction, the development of voice-controlled Human Machine Interaction (HMI) systems has become pivotal [1]. This study delves into the design and experimental evaluation of a voice-controlled elevator system, exploring its significance in mitigating contact concerns, and enhancing accessibility for individuals with disabilities. The adoption of voice interaction serves a dual purpose. First and foremost, it addresses the pressing need for touchless control, crucial in environments where hygiene is paramount, such as hospitals; and desirable, in the context of recent pandemics, in public spaces such as elevators. Additionally, a voice-controlled elevator system contributes to inclusiveness by simplifying access for individuals with disabilities, ensuring a more seamless and equitable vertical mobility experience.

However, several challenges arise during the design and development process. Opting for online voice recognition would demand Internet connectivity in every elevator, a complex and impractical endeavour. In contrast, an offline solution would require embedding the speech processing technology in a low-resource environment, a task that poses considerable challenges and demands careful consideration of efficiency and overall system performance. Furthermore, the acoustics within the elevator cabin pose a significant hurdle, where external noise and reverberation can potentially affect the accuracy of the voice recognition software. All these factors add each an extra layer of complexity to the system's design and performance.

A succinct yet effective solution has been implemented in this work to address these challenges. The system employs offline voice interaction within the elevator cabin, eliminating dependence on external connectivity. An array of microphones is also incorporated to mitigate the adverse effects of cabin reverberation and ambient noise on command detection. Central to its functionality, a voice recognition model has been meticulously trained, specifically tailored for elevator commands, ensuring a seamless and responsive user experience. The presented study investigates the viability and efficiency of the proposed voice-controlled elevator system, considering its implications for public health and accessibility, together with the practical challenges encountered in its implementation. The focus is placed on overcoming acoustic challenges found within the elevator environment, with the ultimate goal of developing a customised carrier board incorporating all essential embedded technologies.

The subsequent sections of this work are organised as follows: Section II delves into the relevant literature in the domain of HMI deployed on low-resource environments. Section III provides a comprehensive description of the proposed voice-controlled elevator system. In Sections IV and V two conducted evaluations, one objective and one subjective, are rigorously detailed. Lastly, Section VI presents the primary conclusions drawn and outlines future avenues for research.

II. RELATED WORK

The inclusion of voice commands as a type of HMI in the elevation sector has already been discussed and explored in several works. Regarding simulated environments, in [2] a small set of voice commands was integrated in a 3 floor elevator mock-up controlled by an Arduino Nano board using a Voice Recognition Module V3[1]. As another example, Meenatchi *et al.* [3] implement a voice command detection system based in CMUSphinx [4] and a single condenser microphone on a software simulating a real elevator. In the research conducted by [5], a set of touch-less sensors which activate the elevator controls are presented as substitutes to conventional buttons. Even though this approach presents a feasible alternative for ensuring safety in hazardous environments such as hospitals, it is not sufficient for enhancing accessibility for users with motor disabilities since a physical interaction is needed for their activation. In the case of [6], the speech recognition system that includes wake-up word detection and intent classification enables an external hardware for activating the corresponding floor button on the original panel. This approach would address the two main issues of safety and accessibility, but requiring an *ad hoc* implementation adapted to each button panel design is not viable for commercialisation on cabins that are already in use.

In a broader sense, voice interaction has been used for controlling different types of machines in many industrial sectors. For example, the authors' previous work [7] explores the use of voice commands alongside a predefined ontology for controlling an industrial collaborative robot. Other studies evaluate the effectiveness of a multi-modal approach for HMI systems, complementing the use of voice commands with other inputs such as computer vision or augmented reality [8]–[11]. In addition, some studies evolve voice interaction by means of a Natural Language Understanding (NLU) component, which gives the user the capability of communicating with the system via natural language instead of using predefined commands [12].

Nevertheless, such systems still encounter problems linked to the acoustic conditions of the environment. Industrial or by any means noisy environments may decrease the accuracy of voice interaction systems [13]. In the case of elevators, apart from the possible noise reaching the main cabin, reverberation due to the closed space can also play an important role in the quality of the recognition [14], prompting the design of novel techniques to address this issue in terms of feature extraction [15] or machine learning models [16].

Moreover, the implementation of voice recognition software within a low-resource environment presents another challenge. Various methodologies have been explored in the literature regarding this matter. These approaches span from employing compact, lightweight neural networks exclusively trained on specific commands [17], to integrating large vocabulary recognition engines within the primary system [18]. Even one of the most promising recognition models developed recently such as OpenAI's Whisper [19] has also embraced quantisation as a strategic approach for its deployment in said challenging environment, as seen in its C++ implementation, whisper.cpp [20].

In this work, a voice-controlled elevation system is implemented across three deployment scenarios: a dedicated server in the cloud, a system on the edge, and ultimately embedded on the main board alongside the complete system. With the objective of enabling a more natural interaction than with voice commands, the speech processing unit was built using the Vosk Automatic Speech Recognition (ASR) module [21], an offline speech recognition toolkit based on Kaldi [22] suited for deployment on low resource environments, while the Google Speech-to-Text API was employed as a contrastive recogniser. In order to evaluate the effect of external noise, the audio acquisition hardware, comprising five different input devices (single or multi-microphone), underwent performance assessment based on Signal-to-Noise Ratio (SNR) and Word Error Rate (WER) metrics, by means of a dataset recorded for the specific purpose of analysing the impact of noise and reverberation across various audio capturing devices. The performance of both ASR systems has also been measured in terms of transcription accuracy and inference speed across various acoustic conditions. Concluding the study, a subjective evaluation involving end-users was conducted in a simulated environment featuring a real elevator cabin in order to gauge the overall usability of the final system.

Hence, the contributions of this work with respect to the state of the art are the following:

- Instead of voice commands, a more natural interaction with the system was pursued.
- The system has been developed in a non-English language, i.e. in Spanish; being easily extendable to other languages.

[1] https://www.elechouse.com/elechouse/images/product/VR3/VR3_manual.pdf

- It featured a more realistic scenario in a real elevator cabin with two different coatings.
- The acoustic environment was analysed using a sound level meter and multiple microphones.
- Two different ASR systems deployed in different environments were compared in terms of performance and latency.
- The whole system was evaluated with real users in a near-real situation featuring clean and noisy conditions.

III. PROPOSED SYSTEM DESCRIPTION

Our proposed system will be placed inside the cabin of an elevator. Ideally, it will occupy a small enough space to be integrated inside the button panel of the elevator. This piece of hardware will be in charge of both the audio acquisition and the communication with an Automatic Speech Recognition (ASR) software that will process the contents of the captured signal. Finally, the obtained textual response will be processed using a rule based approach to detect the intention of the speaker and send the order to the elevator.

Regarding the placement of the ASR module, three different scenarios were studied:

1) Deploying the ASR in cloud
2) Deploying the ASR on edge
3) Embedding the ASR in the HMI component

Scenario 1) adds the possibility of having a more powerful dedicated server that could process the requests of one or more elevator systems more easily, or even handle the Speech-to-Text task to a third-party operator such as Google. However, an open Internet connection is mandatory, adding more failure modes to the system, such as in the transmission of the data.

In Scenario 2), a less powerful dedicated machine could be placed near the cabin, e.g. in the engine room, without the need for an open Internet connection. Nonetheless, the audio signal could not be sent to a third-party operator for processing, requiring an on-premise ASR module for the task.

Ideally, the targeted deployment of the entire system is that of Scenario 3). Embedding the ASR in the same hardware where the audio acquisition component resides would solve all connectivity issues and reduce the costs of the whole setup. Still, the hardware should also be powerful enough for supporting a Speech-to-Text module while keeping the construction and maintenance costs low.

With these objectives in mind, the hardware selected for embedded deployment is a VAR-SOM-MX8M-PLUS System on Module (SOM), which includes a 1.8 GHz Quad Cortex-A53 i.MX 8M Plus processor and 4 GB of RAM. In order to minimise hardware and operating system set-up time, an Evaluation Kit was used for the proof of concept. The final goal, however, is to build a custom carrier board that only integrates the SOM and the necessary components.

A diagram of the whole system and the three scenarios is shown in Figure 1.

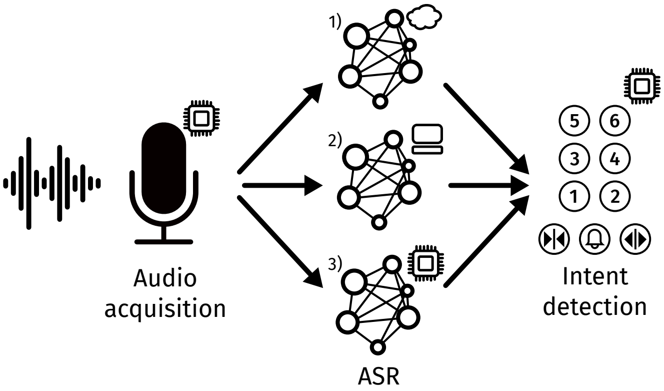

Figure 1. Diagram of the proposed system including the main three validation scenarios: Scenario 1) with the ASR deployed on cloud, Scenario 2) with the ASR on edge, and Scenario 3) with the ASR embedded on board.

A. AUDIO ACQUISITION HARDWARE

In the case of audio acquisition hardware, different sensors were explored. First of all, the VAR-SOM-MX8M-PLUS SOM includes an integrated omnidirectional digital stereo microphone, named *internal*, which could be easily integrated in the final carrier board due to its small footprint and reduced number of digital signals required to connect to the processor.

Next, two different ReSpeaker microphone arrays were also considered: a 4-microphone circular array[2], named *circular*; and a 4-microphone linear array for Raspberry Pi[3], named *linear*. The circular disposition of microphones incorporates an XMOS XVF-3000 processor that integrates multiple advanced Digital Signal Processing (DSP) algorithms. The linear array, on the other hand, uses a Raspberry Pi for the implementation of similar procedures. In this context, a delay-sum beamforming algorithm was implemented in C programming language in order to reduce the noise captured by the sensor and improve the incoming speech signal.

Two more microphones were also used for contrasting the chosen audio acquisition hardware. First, a headset equipped with a unidirectional short-distance microphone, named *headset*. This microphone is not physically suitable for this task since users would need to wear it in order to interact with the elevator. However, a unidirectional short-distance sensor would reduce the undesired effects of reverberation and external noise and can therefore serve as a valuable contrast for determining their influence on the whole system. And finally, a commercial speakerphone Jabra Speak 510[4], named *jabra*, consisting of an omnidirectional microphone specifically designed for communication in office meetings that integrates built-in DSP algorithms.

[2] https://wiki.seeedstudio.com/ReSpeaker_Mic_Array_v2.0/

[3] https://wiki.seeedstudio.com/ReSpeaker_4-Mic_Linear_Array_Kit_for_Raspberry_Pi/

[4] https://www.jabra.es/business/speakerphones/jabra-speak-series/jabra-speak-510

This article has been accepted for publication in IEEE Open Journal of the Industrial Electronics Society. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJIES.2024.3483552

B. SPEECH RECOGNITION MODULE

Vosk [21] was chosen as the main ASR architecture implemented for this system, which uses Kaldi [22] models for recognition. In this work, an Acoustic Model (AM) was trained using the *nnet3* DNN configuration on a *chain* acoustic model based on a factorised time-delay neural network (TDNN-F) [23]. This model consisted of 16 TDNN-F layers with an internal cell-dimension of 1536, a bottleneck-dimension of 160 and a dropout schedule of `0,0@0.2,0.5@0.5,0`. It was trained for 4 epochs with a learning rate of $1.5 \cdot 10^{-4}$ and a mini-batch size of 64. As input, it received a concatenation of 40 high-resolution Mel Frequency Cepstral Coefficients (MFCC), augmented using speed perturbation with factors of 0.9 and 1.1 [24] and volume perturbation on a random factor between 0.125 and 2 [25], concatenated with a 100-dimensional i-vector. It was trained on the datasets of *Albayzín* [26], *Multext* [27] and version 5.0 of the Spanish corpus of *Common Voice* [28]. The corresponding number of files and duration of each corpus can be found in Table 1.

Table 1. Number of audio files and duration of the main corpora used to train the acoustic model of the adapted Kaldi model.

	Audio files	Duration
Common Voice	205 869	287:09:07.95
Albayzín	6200	5:33:05.01
Multext	669	47:39.46
Total	212 738	293:29:52.42

In the case of the Language Model (LM), the characteristics of the final system allow us to focus on the desired input for controlling the elevator instead of using a large vocabulary approach. Therefore, a collection of voice commands was chosen to align with the primary functions that this technology is expected to perform. These include:

- Going to a specific destination
- Elevator control commands:
  - Opening of doors
  - Closure of doors
  - Maintaining the doors open
  - Request of assistance

Different versions of these commands were introduced in order to give the user the sensation of naturalness in the interaction with the system. For example, "*vete al primer piso*" (go to the first floor) or "*llévame a la planta uno*" (carry me to the storey one[5]) were both added to the language model as the same intent of going to the first floor. In addition to floor numbers, facilities such as "*recepción*" (reception), "*garaje*" (garage) or "*restaurante*" (restaurant) were also added as destinations.

A total of 1122 voice commands were mapped to 27 destinations (13 floors, from -6 to 6, and 14 facilities) and the 4 control commands. The minimum and maximum number of words per command is 2 and 8, respectively; with a mean of $\mu = 5.18$ and a standard deviation of $\sigma = 1.28$ words. The lexicon is composed out of 67 different words.

Once the interactions were defined, a 7-gram model with modified Kneser-Ney smoothing was built using the KenLM toolkit [29], which also included two special words that mark the beginning and the end of the sentence.

As a contrastive ASR system, Google's Speech-to-Text API[6] was also integrated in the on-cloud scenario.

IV. OBJECTIVE EVALUATION

In order to test the efficacy of the proposed system, a series of objective tests were executed on a set of audios specifically recorded for this task.

A. CONSTRUCTED DATABASE

A subset of 15 commands chosen from the texts used for training the LM were recorded by 5 speakers inside an elevator cabin used for evaluation purposes. Said cabin was placed in a laboratory that presented some background noise due to working personnel. The 5 microphones described in Section III-A were all placed near the button panel of the cabin parallel to the wall, and all recorded simultaneously the utterances of each speaker. The *linear* ReSpeaker was placed horizontally. A small table was set inside the cabin for placing the additional equipment needed. The cabin doors remained closed during the recordings. For better visualisation purposes, a diagram of the disposition of the cabin is shown in Figure 2.

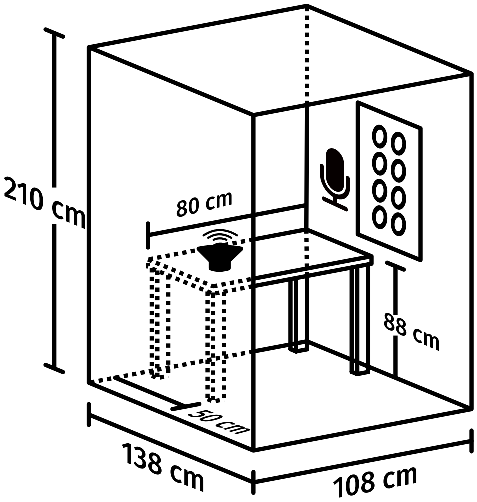

Figure 2. Diagram of the elevator cabin (not to scale).

In order to check the impact of external noise to the system, a second speakerphone Jabra 510 set on the table was used for emitting predefined sounds in loop during the recording sessions. A total of 4 different noise conditions were scheduled: no injected noise (*base*), the noise of an

[5] Odd expression in English, but completely normal for a native Spanish speaker to utter.

[6] https://cloud.google.com/speech-to-text

elevator (*noise*), a conversation recorded inside an elevator in high volume (*conv*), and the same conversation in lower volume (*c soft*). The name *c soft* should not imply that the injected noise is a soft conversation, but a softer version of the injected noise *conv*. The elevator noise (*noise*) began and ended with a bell, an audio clue normally used for indicative purposes. The conversation (*conv* and *c soft*) also featured a quieter version of machinery noise since it was recorded inside a lift cabin. The spectrograms of these injected noises are shown in Figure 3.

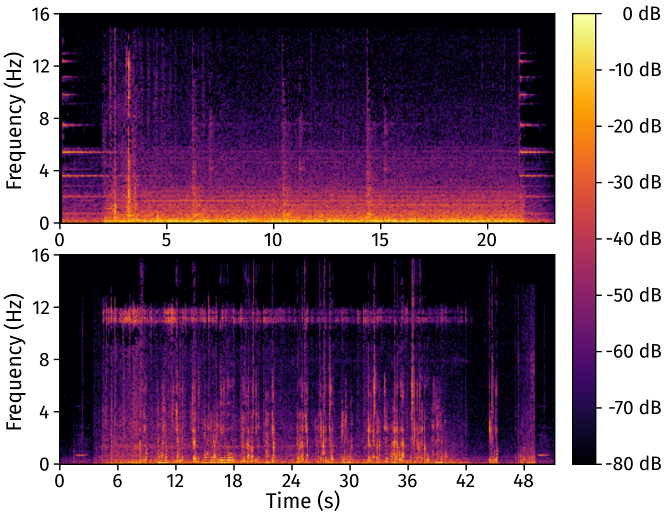

Figure 3. Spectrograms of the two injected noises (above: *noise*, below: *conv* and *c soft*) played in the background during the recording session.

The elevator cabin featured two interchangeable internal coatings: wood and metal. Two recording sessions were performed using each of these coatings.

All these scenarios resulted in a total of 2900 recorded audio commands.

B. ACOUSTIC ANALYSIS

To begin with the evaluation, an analysis at the acoustic level was performed on the recorded database.

Given that the conditions in which the database was recorded were far from being a clean environment, the noise levels while recording the database were characterised by means of a sonometry using a PCE-MSM 4 Sound level meter[7] configured to measure C-weighted sound pressure levels in dB (dBC) on a slow range (1 measurement per second). The device was placed on the table next to the microphones, and a person was inside the cabin during the recording. The obtained results of these measurements can be seen in Figure 4.

The first conclusion that can be obtained from this data is that, even with no external noise added (*base*), the sound pressure levels obtained are quite elevated due to the background noise of the working personnel in the laboratory where the cabin is placed. Moreover, the lowest and average values obtained from the sonometries with the metallic coating are higher than those with the wooden interior, which indicates that the effect of the reverberation inside the cabin with the metallic coating is indeed more relevant than for its wooden counterpart.

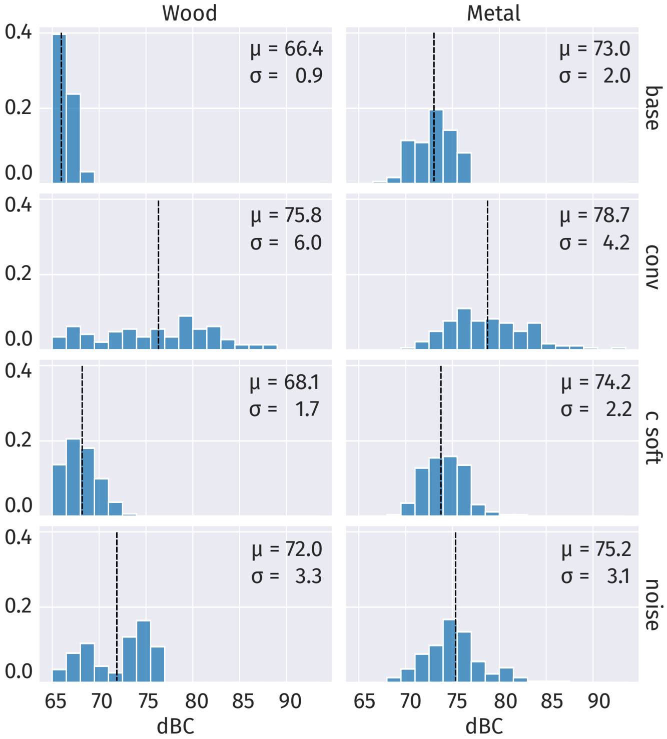

Figure 4. Density histogram of measured sound pressure levels for the four different injected noise conditions on the two internal coatings of the evaluation cabin (left: wood, right: metal). The average value is marked with a dashed line. The average and standard deviation values are indicated with the symbols μ and σ respectively.

In addition, the Signal-to-Noise Ratio (SNR) of the audio signals captured by the microphones was estimated by comparing the power of the signal when the speaker uttered a command (X) with the trailing noise until the next command (N) using the following formula:

$$\mathrm{SNR} = 10\log_{10}\frac{\max\left(P(X)-P(N),10^{-6}\right)}{P(N)}$$

where $P(A)$ is the estimated power of the signal A. The signal X was estimated during speech activity, and the noise N using the pauses between commands. The results are shown in Figure 5. The mean value of the obtained SNR on each recording configuration is shown in Table 2. Please note that the reported SNR values are not directly comparable to the noise level measurements shown in Figure 4. This discrepancy is due to the differing positions of the sound level meter and the microphones, as well as the influence of signal enhancement algorithms used by the recording devices.

It can be seen that the SNR drops in presence of the conversation at a higher volume (*conv* noise) when compared with the rest of noise setups, which correlates with the higher

[7] https://www.pce-instruments.com/eu/measuring-instruments/test-meters/pce-instruments-sound-level-indicator-pce-msm-4-det_5971763.htm

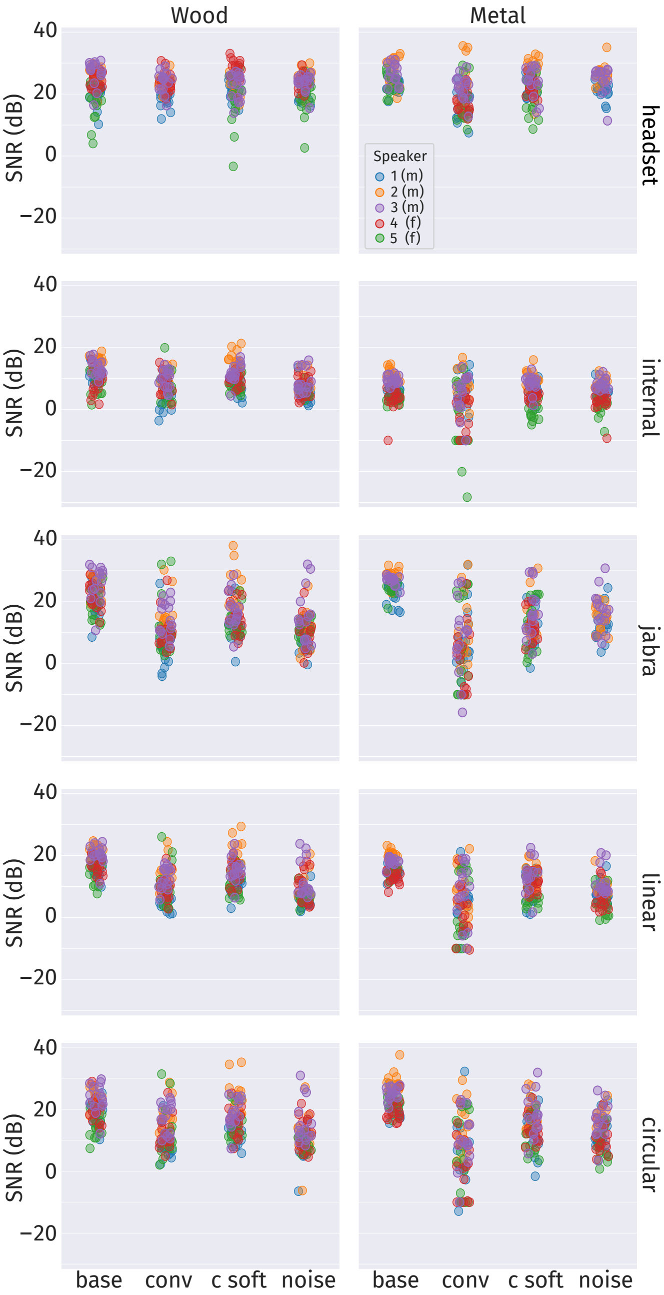

Figure 5. SNR (dB) values of the audio signals constituting the recorded dataset recorded by the different microphones divided by coating (left: wood, right: metal), microphone (per row) and speaker (marker colour).

sound pressure levels obtained in the sonometries of this particular noise injection.

Regarding the microphones, it can be easily deduced that the best and worst sensors in terms of SNR are the *headset* and the internal microphone embedded on the evaluation board (*internal*), respectively. These two microphones are discarded for the final prototype due to the impracticality of the former and the low performance of the latter.

The average SNR values obtained by the 4-microphone linear array (*linear*) are lower than those from its circular counterpart (*circular*). However, the sparsity of the data is slightly lower in the case of the *linear* microphone as seen in Figure 5. Curiously, the point clouds obtained by the speakerphone *jabra* and the *circular* ReSpeaker give almost equivalent SNR values in the column Total of Table 2.

Table 2. Mean SNR values for each microphone, injected noise and coating (wood: W, metal: M). The total SNR value for each microphone and coating is included in the last column.

Setup		base	conv	c soft	noise	Total
headset	W	23.2	22.6	22.7	22.3	22.7
	M	25.3	19.6	23.3	24.1	22.8
internal	W	11.2	7.8	10.5	7.1	9.2
	M	6.5	2.1	5.2	5.2	4.9
jabra	W	22.3	11.3	16.1	11.2	15.2
	M	25.6	7.0	13.7	14.7	14.7
linear	W	17.8	10.6	14.3	8.8	12.9
	M	16.1	4.8	11.0	8.0	10.0
circular	W	20.0	13.3	17.0	11.0	15.3
	M	23.0	7.6	15.0	12.9	14.6

Also, excepting the *headset* that reduces the impact of the reverberation, the audios recorded on a metallic coating setup present in general a higher noise than their wooden counterparts.

In addition, impulse response measurements of the cabin were conducted for both coatings with a person inside, the acoustic conditions of the recorded database, estimated by a T20 measurement, resulting in similar RT60 values to be approximately 420 ± 40 ms.

C. ASR EVALUATION

The second evaluation conducted on the proposed system tried to characterise the performance of both used ASR engines: our adapted model deployed on Vosk in comparison with Google Speech-to-Text's generic Spanish model. In order to do so, the WER obtained by both ASR engines on the previously recorded dataset was measured. The results are shown in Figure 6.

As these results show, the WER values obtained by Google's Speech-to-Text are higher than those obtained by the Vosk model for all cases. This is mainly due to the fact the former is a generic large-vocabulary model whilst the latter was specifically adapted for the voice commands present on the database. Nevertheless, despite the WER values for Vosk are significantly low, the ASR system still suffers notably in presence of a conversation played on high volume (*conv* noise injection) due to the appearance of other words in the audio signal.

In most of the cases, the WER values obtained on the voice commands recorded with the metallic coating are higher than their wooden counterparts, which aligns with the SNR and sound pressure measurements obtained in the acoustic analysis of the audio files.

Thanks to the capability of the microphone *headset* to reduce the surrounding noise, the WER values for said sensor are practically zero for all configurations. This indicates that

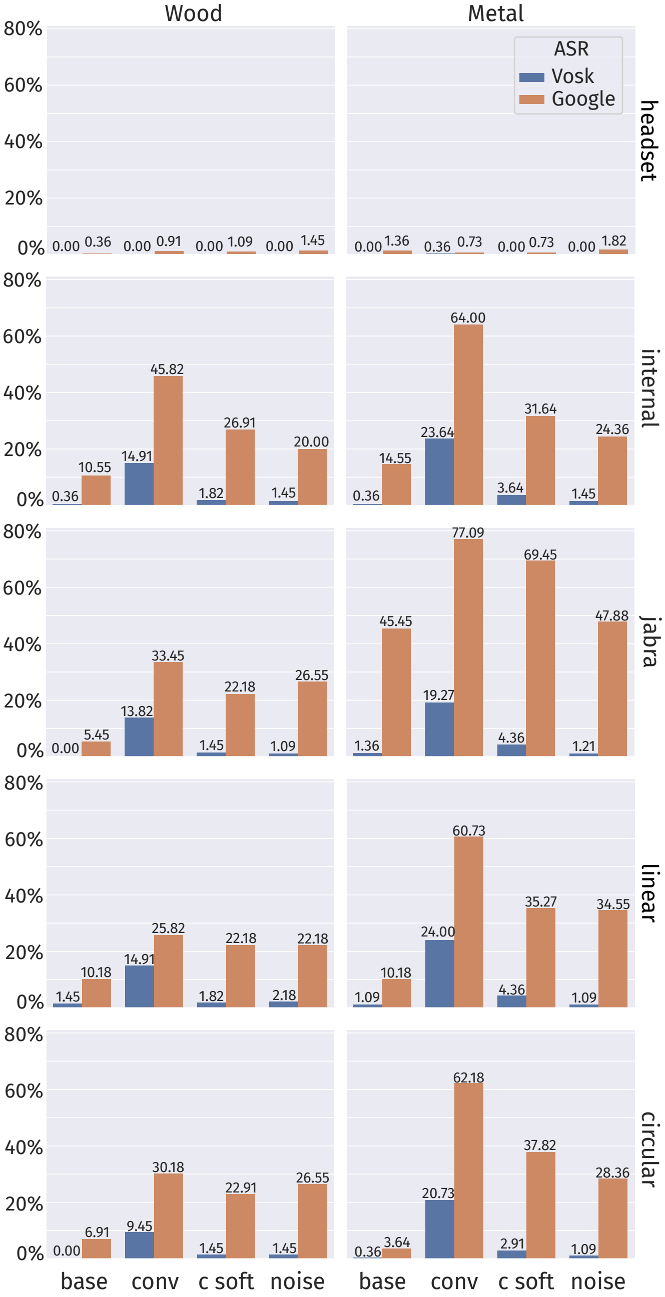

Figure 6. WER values obtained by both ASR engines (blue: Vosk, orange: Google's Speech-to-Text) on the audio files constituting the recorded dataset, separated by coating (left: wood, right: metal) and microphone (per row).

the higher error values on the rest of the microphones are indeed due to the quality of the signal.

D. INFERENCE TIME EVALUATION

Latency is a key aspect to consider in order to decide which of the three proposed scenarios (on cloud, on edge and embedded) is the most suitable for the targeted task. This is why another set of measurements was conducted in order to characterise the Speech-to-Text systems' inference speed on the different deployment scenarios.

In the case of the on-cloud scenario, Google's Speech-to-Text API and two instances of Vosk were deployed: on an Intel Core i9-13900K up to 5.8 GHz and on an Intel Xeon CPU E5-2683 v4 up to 2.1 GHz. Regarding the on-edge scenario, Vosk was deployed on an Intel Core i5-7500 up to 3.4 GHz. And finally, for the embedded scenario, on the i.MX 8M Quad Cortex-A53 up to 1.8 GHz.

In order to present a faithful measurement, 1500 of the recorded audio files were sent to the tested environments a total of 10 times each and their response times were averaged. Since the goal of this evaluation is to observe the performance of the deployed ASR on each CPU, this test was conducted without taking data transmission into account, except for Google's Speech-to-Text since this information was not available to the user. The obtained latency values as a function of the duration of the tested audio commands are presented in Figure 7. A dashed black line representing a Real Time Factor (RTF) of 1 is also marked in the figure.

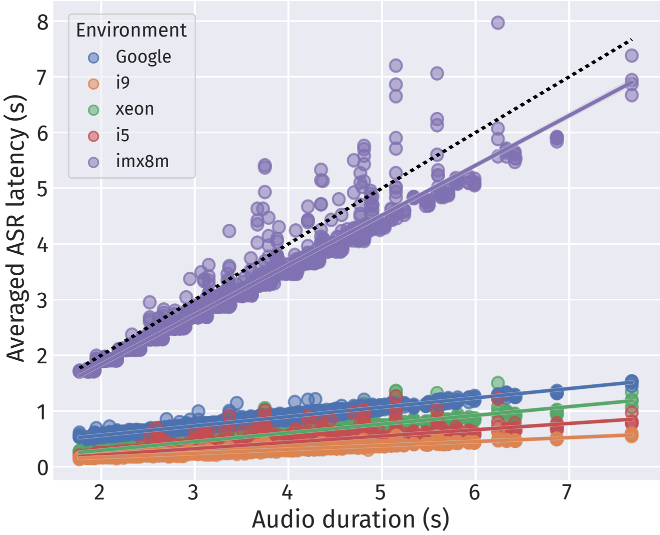

Figure 7. Averaged latency values obtained by Google and Vosk deployed on 4 different architectures when transcribing 1500 of the voice commands recorded in the dataset as a function of their duration. A dashed black line represents a RTF of 1.

The most notable remark that can be drawn from these data is that Vosk deployed on the i.MX 8M processor requires significantly more time than the rest of the environments, even sometimes reaching values higher than the duration of the transcribed audio, i.e. a RTF greater than 1. However, thanks to the capability of Vosk to transcribe the audio in streaming, these latency times lower than a RTF of 1 will produce responses in almost real time as perceived by the final user.

Regarding the rest of the environments, no significant difference between them can be drawn since the purpose of the whole system is to perform real-time recognition. Obviously, a faster CPU such as the Intel Core i9 provides a response in less time than a slower one such as the Intel Xeon, but since all presented an averaged RTF value less than 1, these results conclude that any of the measured environments could be used as the main ASR system.

E. COMMUNICATION PROTOCOL EVALUATION

In scenario 2) described in Section III, where the ASR is deployed on the Edge, a test environment was created in a private network in order to evaluate two different communication protocols.

On the one hand, the Message Queuing Telemetry Transport (MQTT) protocol was utilised, whereas on the other, a REST API over HTTP was developed. In both scenarios, the Transport Layer Security layer was not considered, allowing network traffic to be captured using the Wireshark application to log timestamps for each network event and the exchanged messages between client and server.

The test environment consisted of a client and a server connected to the same private Ethernet network and with controlled traffic. The client was in charge of capturing the audio and sending it to the server (via MQTT broker or HTTP Web server), where the ASR was executed and the result was sent back to the client. The server operates on Ubuntu Linux 20.04 LTS on an Intel Core i9 13900K and had other services running in background. In the case of the communication using MQTT, a Mosquitto[8] broker and an Apache Web server was deployed in separate Docker containers; whereas in the case of the HTTP protocol, Python's FastAPI[9] web framework including a WSGI module was chosen.

In order to simplify the data capture and processing associated with evaluating the protocols, it was considered to use a PC as the client instead of the Evaluation Kit based on the VAR-SOM-MX8M-PLUS. Since the focus was placed on the latency times inherent in the protocol itself, the processing power of the client was a secondary concern.

The evaluation was performed using the same 1500 audio files previously used for ASR latency evaluation, which were sent 10 times each to the server for processing and averaged. The results are shown in Figure 8.

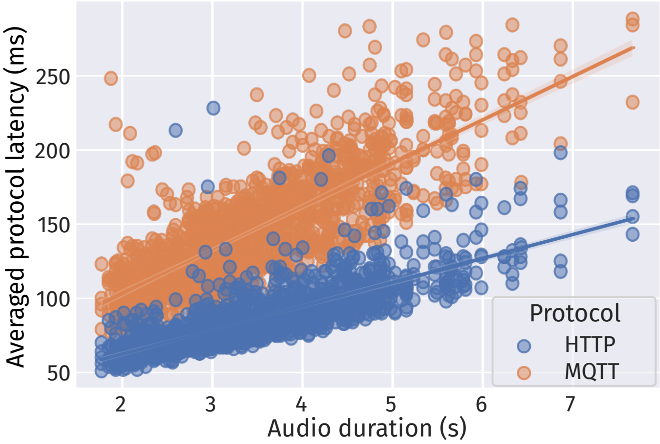

Figure 8. Averaged latency times obtained by the protocols HTTP (blue) and MQTT (orange) when sending and receiving the content and respective transcriptions of 1500 of the voice commands recorded in the dataset as a function of their duration.

According to the obtained results, the use of a REST API is considered more efficient than the use of the MQTT protocol in terms of latency. This could be due to the fact that the MQTT protocol requires the use of an external broker that manages the communication between client and server, whereas the messages are sent without an intermediary program directly to the recipient on the REST API.

Nevertheless, to ascertain the potential impact of processing power on the latency times, a new test was executed locally using the i9 processor and the VAR-SOM-MX8M-PLUS SOM, deploying both the client and the server concurrently on the same machine. The assessment used the identical set of 1500 audio files employed in the aforementioned evaluation. The corresponding outcomes are illustrated in Figure 9.

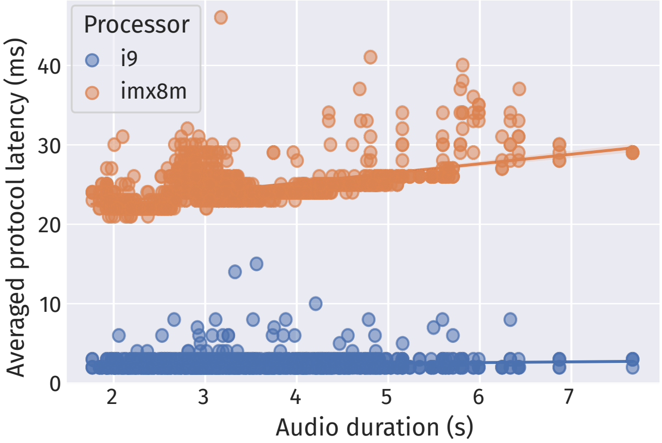

Figure 9. Averaged latency times obtained by the HTTP protocol client-server API deployed on an Intel Core i9 13900K (blue) and on the VAR-SOM-MX8M-PLUS Evaluation Kit (orange) when sending and receiving the content and respective transcriptions of 1500 of the voice commands recorded in the dataset as a function of their duration.

As it can be seen in the obtained results, the latency times achieved by the Intel Core i9 processor exhibit a relatively consistent performance across varying lengths of input audio. In contrast, due to the significantly lower computational capability of the i.MX 8M processor, the communication time with the REST API increases with the length of the input audio file.

V. SUBJECTIVE EVALUATION

Additionally to the objective evaluation presented in the previous section, in order to test the usability of our system, two subjective evaluations with end users were executed as well, one featuring no external noise, and another injecting the external noise labelled as *conv*.

A. EVALUATION SETUP

With the productisation of the final system in consideration, the chosen layout for this evaluation does not correspond with the most optimal configuration of microphone, coating, ASR system and CPU.

[8] https://mosquitto.org/
[9] https://fastapi.tiangolo.com/

First, the metallic coating was chosen for this evaluation even though it had more adverse acoustic conditions since many elevators feature this material on their inside walls. Forcing a wooden coating would exclude a remarkable percentage of already deployed machines from integrating the proposed system.

Concerning the ASR system, the trained Vosk model was chosen to be deployed in the i.MX 8M processor. Despite this CPU requiring a higher computing time than the rest of tested environments, this integration allows the whole system to be a single SOM independent from external more powerful devices and network connection protocols, reducing the costs and maintenance required for the other scenarios. Moreover, the use of Vosk instead of Google's Speech-to-Text API enables a higher degree of freedom for adjusting the AM or the LM to specific deployments, as well as no additional fee for the usage of the transcription pipeline.

In terms of the input sensor, the ReSpeaker 4-microphone linear array including the delay-sum algorithm (*linear*) was chosen. The difficulty of integrating this microphone on a custom SOM is lower. This is due to the fact that the sensors are the only required hardware since the beamforming algorithm can be implemented on the same CPU as the ASR system is deployed. The WER obtained in the evaluation for this configuration supposing no external noise was of 1.09 %, and of a 24.00 % when adding the external noise labelled as *conv*.

In order to avoid false positives and make the implemented voice-controlled system responsive only to requested commands, the wake-up word *"Oye Orona"* was implemented using Picovoice's Porcupine toolkit[10]. The ASR only recognised audio once the wake-up word was detected. Thanks to the Spanish sentence structure, the intent of the user extracted from the final part of the transcription, checking if this result matched any of the available destinations or control commands, emitting a cached audio response.

The two evaluations were conducted within the laboratory and cabin where the database was recorded. This assessment simulated a practical scenario of a voice-controlled elevator system situated in a multi-storey hotel with various facilities. Participants were provided with instructions, including a set of 12 predefined interaction scenarios. The instructions provided to the end users can be found in Appendix A. Finally, evaluators were presented with a questionnaire featuring the System Usability Scale (SUS) [30], comprising 10 questions (Appendix B) related to user experience, rated on a scale from 1 to 5. The final score, ranging from 0 (worst case) to 100 (perfect score), is derived by considering the inverse polarity of even and odd-numbered questions. Thus, a perfect SUS score of 100 is achieved with a rating of 5 for odd-numbered questions and a rating of 1 for even-numbered questions. Additionally, an optional comments section for gathering user feedback was also provided.

B. RESULTS

The described system was evaluated two times: one evaluation featured no added noise, whilst the other was conducted while injecting the noise labelled as *conv* inside the cabin.

The evaluation with no additional noise was conducted by a total of 22 end users – 14 male and 8 female – with maximum, minimum and average ages of 40, 19 and 27, respectively. Regarding the noisy evaluation, a total of 16 end users – 9 male and 7 female – volunteered for the test, with maximum, minimum and average ages of 50, 25 and 34, respectively. The authors acknowledge that these age distributions are not the most representative, since a higher representation of older users would have been preferable. However, this range is contingent on the final set of volunteers who agreed to participate. The evaluators exhibited a wide range of Spanish accents, including Northern, Andalusian, Canarian, and Latin American, as well as bilingual users of other Spanish languages such as Catalan or Basque. Additionally, some participants were non-native Spanish speakers. The obtained values for the SUS questionnaire are displayed in Figure 10.

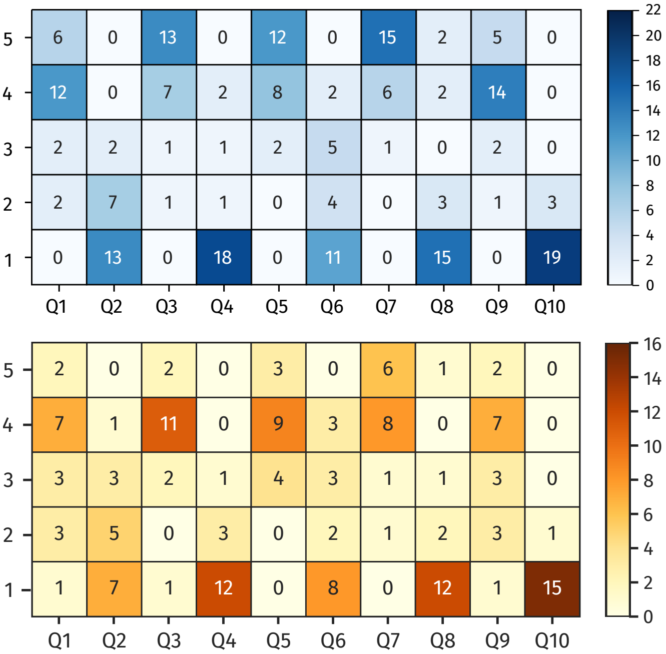

Figure 10. Colormap of the obtained values on the 10 questions of the SUS questionnaire in clean (blue) and noisy (orange) conditions.

The result of the evaluation with no injected noise yielded an average SUS value of 84.7 and a standard deviation of 10.9, with minimum and maximum values of 55 and 97.5 respectively. In the case of the evaluation with added noise, the SUS value obtained by the questionnaires was of 77.2 and a standard deviation of 15.7, with minimum and maximum values of 42.5 and 95 respectively.

In the experiment conducted by Bangor *et al.* [31], where SUS values were associated with a 7-word adjective scale, the average SUS value corresponding to the second and third best adjectives "Excellent" and "Good" were found to be

[10] https://picovoice.ai/platform/porcupine/

85.5 and 71.4 respectively. According to their findings, the obtained results of 84.7 and 77.2 fall into the range of these two categories. As stated by Lewis and Sauro [32], a SUS score exceeding 84.1 is indicative of an A+ grade, surpassing over 96 % of the industrial usability studies and surveys analysed by the authors. Furthermore, as highlighted in that study, a minimum SUS score of 80 is considered desirable for labelling a product as providing an "above-average user experience" in industrial applications. Therefore, the main conclusion derived from the evaluation with no added noise indicates a satisfactory performance of the proposed system, even though the evaluation was conducted with a microphone, coating, and ASR system configuration proven not to be the most optimal among those tested. Regarding the evaluation on a noisy environment, the obtained results suggest a B grade according to the same study, which indicates that the proposed system performs acceptably even on hard acoustic conditions.

In addition to the numerical data, the most valuable insights obtained from the questionnaire is the feedback offered by the evaluators. Numerous users complained that some of the command structures they used were not registered on the system. Fortunately, addressing these concerns is manageable, as implementing new rules or modifying the LM of the ASR does not require significant effort. A notable number of comments highlighted difficulties with the wake-up word module, including instances of poor detection, specially remarkable on the noisy evaluation. This was the case for the lowest given SUS score of 42.5, since this user stated that the functioning of the system was correct once the wake-up word was detected. It is important to mention that this user was not a native Spanish speaker. Additionally, users expressed dissatisfaction with the lag between the wake-up word detection and the initiation of recognition. Therefore, the improvement on the wake-up word engine could lead to higher SUS values in future evaluations, even for the more adverse conditions when injecting the *conv* noise. Unfortunately, the participant with an associated SUS score of 55 in the evaluation with no injected noise did not contribute suggestions for system improvement. Notwithstanding, this user's responses indicate that the perceived problem stems from the system's excessive complexity and difficulty.

Concerning the answers to the clean evaluation, Q6 (*I thought there was too much inconsistency in this system*) received the lowest score, marked by a notably high frequency of responses greater than 2. This observation is likely associated with the earlier mentioned feedback concerning transcription and wake-up detection errors. Conversely, Q10 (*I needed to learn a lot of things before I could get going with this system*) received the highest ratings, indicating that the system was perceived as user-friendly and intuitive. This positive perception is potentially attributed to the resemblance of the system's behaviour to other commercially available voice assistants. Additionally, questions Q1 (*I think that I would like to use this system frequently*) and Q9 (*I felt very confident using the system*) stand out as the only ones where the majority of responses do not align with the values associated with highest SUS scores. This suggests that evaluators may find themselves using the proposed system to interact with the elevator, but notable inclination towards alternative channels exists. Additionally, despite the system's high reliability, it may not be perceived as optimal. However, this perception could change once the system is evaluated on target users such as elderly or disabled people.

Finally, when comparing the differences between the two evaluations, the number of questions not receiving the highest score increases to five, specifically in the "positive" odd-numbered questions. Since Lewis and Sauro do not asses a significant difference when altering the polarity of the "negative" questions [33], this result can be considered coincidental rather than a consequence of the questionnaire formulations. Nonetheless, the two questions that yielded a lower SUS score in the evaluation with added noise, Q1 and Q9, coincide with those in the cleaner evaluation.

VI. CONCLUSIONS

In this work, the viability and effectiveness of a voice controlled elevator system has been evaluated, both objectively and subjectively, focusing on the suitability of the multiple elements that compose the system in various acoustic conditions.

In order to objectively evaluate the proposed system, a database consisting of a total of 2900 audio files was created. It is constituted by a selection of voice commands recorded on an elevator cabin by multiple speakers on various microphones and different acoustic environments: two interior coatings (metal and wood) and four noise conditions.

The differences between the two coatings of the evaluation cabin were characterised my means of a sonometry, concluding that the effect of the reverberation and therefore the perceived noise is more prominent when using a metallic interior. In terms of SNR, the same conclusion was observed, this time using the contents of the recorded dataset. Moreover, the performance of the 5 different microphones used to record the voice commands was gauged, concluding that the ReSpeaker 4-microphone circular array obtained an overall cleaner signal than its linear counterpart that implemented a delay-sum beamforming algorithm. However, the latter presented the advantage of being embeddable in the final SOM with less hardware requirements.

The recorded dataset was transcribed using two different ASR systems: a Vosk model whose LM was adapted to this specific task and Google's Speech-to-Text API. In terms of WER, the error obtained by the former was lower than the latter in all cases, thanks to the LM suited for the voice commands. Moreover, the effect of a background conversation and the reverberation of the cabin were found more relevant than noise in terms of transcription errors.

For the purpose of deciding if the three proposed scenarios (on cloud, on edge and embedded) are suitable for this task, latency times on the ASR inference and on the communication protocol were measured using 1500 voice commands from the recorded dataset. When deployed on powerful CPUs, Vosk has no issue on transcribing audios with small RTF values, as well as Google's Speech-to-Text API. In the case of the i.MX 8M processor, however, the inference time increases drastically almost reaching a RTF of 1, or even greater in a few cases. Regardless of this limitation, Vosk is able to process an audio in streaming and is still perceived as real time by end users. In addition to the ASR system's inference time, the latency of the MQTT and HTTP communication protocols has also been compared. Since MQTT requires the use of an external broker for managing the packages sent between client and server, a slightly greater latency has been observed when compared with a REST API over HTTP.

Alongside the quantitative analysis, two subjective evaluation in clean and noisy environments have been undertaken in order to gauge the usability of the system by means of a SUS questionnaire. The setup used for this test was not the configuration that scored the best results in the previous measurements, but the most optimal one in terms of productisation. Despite of this, the average SUS scores of 84.7 and 77.2 given by the participants in the clean and noisy environments respectively reflect the satisfactory experience when testing the system. The feedback provided by the evaluators reflected the following highlights. First, the need of adding a more diverse set of commands, specifically those regarding the request of assistance from an external operator. This is an issue easy to overcome due to the nature of the ASR system and its adapted LM. Additionally, regarding the used wake-up word engine and the unexpected noisy laboratory acoustic conditions, participants considered the system to be too prone to false negatives. This suggests that another implementation more robust to this particular external noise could significantly increase the usability scores in future evaluations.

Regarding future work, a more complex interaction system could be tested, including intent classification for more elaborated or less direct queries. This would also require the implementation of a low resource Text-to-Speech algorithm in order to synthesise non-cached responses. The integration of more complex transcribers in embedded environments, such as OpenAI's Whisper's C++ implementation whisper.cpp, could also improve the capabilities of the overall system including, for example, multilingual support. The performance of these ASR systems in terms of WER and inference time on embedded systems should also be compared with those used in this study. Moreover, a more exhaustive research focused on reducing these models for even less powerful hardware could decrease the price of the final product. Regarding the subjective evaluation, as it was previously discussed, the lack of target end-users such as elderly or disabled people should encourage a future research on the viability of the presented system focused specifically on these vulnerable collectives. Besides, the use of an actual elevator rather than a simulated environment in eventual tests will be a primary objective for the ongoing work. Lastly, following the main goal of this project, a final prototype integrated on a single Main Carrier featuring the whole pipeline will be assembled and tested, first in laboratory conditions, and finally integrated in real deployed elevators.

.

APPENDIX A SUBJECTIVE EVALUATION SCENARIO

Orona has equipped a hotel elevator with voice interaction capabilities.

To start communication with the elevator, it is necessary to use the activation word "*Oye Orona*", after which the elevator will begin to listen.

The voice assistant may help you with:

- Movements between floors (e.g. "go to the first floor"; "go down to the garage", "I would like to go to the reception")
- Opening and closure of elevator doors (e.g. "open the doors", "close the doors")
- Contacting an operator (e.g. "I would like to contact an operator")

This is the distribution of the hotel floors where the elevator is located:

Floor	Services
Attic	Swimming pool
6	Hair saloon
5	
4	
3	
2	
1	Café / Restaurant
Ground floor	Hall / Reception
-1	Garage / Parking
-2	Basement

You may refer to them both by the name of the floor or by the name of each service.

Your task is to interact with the elevator using your voice in the following scenarios:

1) You have arrived at the hotel by car and parked in the garage. Now you have to do the check-in.
2) You go to your room 404 to unpack.
3) You are hungry and decide to eat something at the restaurant.
4) You go to your room to put on your swimsuit.
5) You take a dip in the swimming pool.
6) You go to shower and change in your room.
7) A taxi is waiting for you at reception to take you to a work meeting.
8) You return to the hotel for dinner.
9) The elevator has stopped and you need help from an operator.
10) You sleep in your room.
11) The next day, you check out.
12) You take the car to leave.

APPENDIX B SYSTEM USABILITY SCALE

1) I think that I would like to use this system frequently.

2) I found the system unnecessarily complex.
3) I thought the system was easy to use.
4) I think that I would need the support of a technical person to be able to use this system.
5) I found the various functions in this system were well integrated.
6) I thought there was too much inconsistency in this system.
7) I would imagine that most people would learn to use this system very quickly.
8) I found the system very cumbersome to use.
9) I felt very confident using the system.
10) I needed to learn a lot of things before I could get going with this system.

References

[1] M. Z. Iqbal and A. G. Campbell, "From luxury to necessity: Progress of touchless interaction technology," Technology in Society, vol. 67, p. 101796, 2021.

[2] N. A. Pande, J. Chikhalkar, V. Jadhav, S. Guha, and R. Waghmare, "Voice-controlled elevator with solar backup and SOS messaging," in Soft Computing for Security Applications, pp. 469–483. Singapore: Springer Nature Singapore, 2023.

[3] D. Meenatchi, R. Aishwarya, and A. Shahina, "A voice recognizing elevator system," in Proceedings of the International Conference on Soft Computing Systems, pp. 179–187. New Delhi: Springer India, 2016.

[4] "CMU Sphinx," https://cmusphinx.github.io/, accessed: 2023-12-05.

[5] A. S. Shinde, A. S. Jamdar, K. D. Joshi, and S. T. Sarode, "A CNN based speech recognition approach for voice controlled elevator," in 2021 5th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT), DOI 10.1109/ICEECCOT52851.2021.9707928, pp. 728–733, 2021.

[6] Y. Liu, W. Wang, and Y. Li, "Realization of contactless elevator control panel system based on voice interaction technology," in 2021 3rd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA), DOI 10.1109/SUMMA53307.2021.9632098, pp. 591–594, 2021.

[7] A. González-Docasal, C. Aceta, H. Arzelus, A. Alvarez, I. Fernández, and J. Kildal, "Towards a natural human-robot interaction in an industrial environment," Conversational Dialogue Systems for the Next Decade, pp. 243–255, 2021.

[8] M. Majewski and W. Kacalak, "Human-machine speech-based interfaces with augmented reality and interactive systems for controlling mobile cranes," in Interactive Collaborative Robotics, pp. 89–98. Cham: Springer International Publishing, 2016.

[9] D. Yongda, L. Fang, and X. Huang, "Research on multimodal human-robot interaction based on speech and gesture," Computers & Electrical Engineering, vol. 72, DOI https://doi.org/10.1016/j.compeleceng.2018.09.014, pp. 443–454, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790618315222

[10] W. Kaczmarek, J. Panasiuk, S. Borys, and P. Banach, "Industrial robot control by means of gestures and voice commands in off-line and on-line mode," Sensors, vol. 20, DOI 10.3390/s20216358, no. 21, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/21/6358

[11] A. Rogowski, "Scenario-based programming of voice-controlled medical robotic systems," Sensors, vol. 22, DOI 10.3390/s22239520, no. 23, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/23/9520

[12] T. Desot, F. Portet, and M. Vacher, "End-to-end spoken language understanding: Performance analyses of a voice command task in a low resource setting," Computer Speech & Language, vol. 75, DOI https://doi.org/10.1016/j.csl.2022.101369, p. 101369, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0885230822000134

[13] S. Li, M. O. Yerebakan, Y. Luo, B. Amaba, W. Swope, and B. Hu, "The Effect of Different Occupational Background Noises on Voice Recognition Accuracy," Journal of Computing and Information Science in Engineering, vol. 22, DOI 10.1115/1.4053521, no. 5, p. 050905, 03 2022. [Online]. Available: https://doi.org/10.1115/1.4053521

[14] H. Nam and Y.-H. Park, "Effect of reverberation on phonemes in automatic speech recognition under reverberant environment," in INTER-NOISE and NOISE-CON Congress and Conference Proceedings, vol. 261, no. 1, pp. 5719–5731. Institute of Noise Control Engineering, 2020.

[15] H. F. Pardede, V. Zilvan, D. Krisnandi, A. Heryana, and R. B. S. Kusumo, "Generalized filter-bank features for robust speech recognition against reverberation," in 2019 International Conference on Computer, Control, Informatics and its Applications (IC3INA), pp. 19–24. IEEE, 2019.

[16] Y.-L. Liao, C.-H. Lin, R.-Y. Lyu, and J.-S. R. Jang, "Improving asr in reverberant environments," in 2022 13th International Symposium on Chinese Spoken Language Processing (ISCSLP), pp. 165–169. IEEE, 2022.

[17] H. Younis and J. H. Hansen, "Challenges in real-time-embedded iot command recognition," in 2021 IEEE 7th World Forum on Internet of Things (WF-IoT), DOI 10.1109/WF-IoT51360.2021.9595903, pp. 848–851, 2021.

[18] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, Q. Liang, D. Bhatia, Y. Shangguan, B. Li, G. Pundak, K. C. Sim, T. Bagby, S.-y. Chang, K. Rao, and A. Gruenstein, "Streaming end-to-end speech recognition for mobile devices," in ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), DOI 10.1109/ICASSP.2019.8682336, pp. 6381–6385, 2019.

[19] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in International Conference on Machine Learning, pp. 28 492–28 518. PMLR, 2023.

[20] "Whisper.cpp," https://github.com/ggerganov/whisper.cpp, accessed: 2023-12-05.

[21] N. V. Shmyrev, "Vosk offline speech recognition toolkit," https://alphacephei.com/vosk/.

[22] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[23] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, "Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks," in Proc. Interspeech 2018, DOI 10.21437/Interspeech.2018-1417, pp. 3743–3747, 2018.

[24] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in Proc. Interspeech 2015, DOI 10.21437/Interspeech.2015-711, pp. 3586–3589, 2015.

[25] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in Proc. Interspeech 2015, DOI 10.21437/Interspeech.2015-647, pp. 3214–3218, 2015.

[26] F. Casacuberta, R. García, J. Llisterri, C. Nadeu Camprubí, J. Pardo, and A. Rubio, "Desarrollo de corpus para investigación en tecnologías del habla (Albayzín) [Development of corpora for research on speech technologies]," Procesamiento del lenguaje natural, no. 12, pp. 35–42, 1992.

[27] E. Campione and J. Véronis, "A multilingual prosodic database." in ICSLP, vol. 98, pp. 3163–3166, 1998.

[28] R. Ardila, M. Branson, K. Davis, M. Kohler, J. Meyer, M. Henretty, R. Morais, L. Saunders, F. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," in Proceedings of the Twelfth Language Resources and Evaluation Conference, pp. 4218–4222. Marseille, France: European Language Resources Association, May. 2020. [Online]. Available: https://aclanthology.org/2020.lrec-1.520

[29] K. Heafield, "Kenlm: Faster and smaller language model queries," in Proceedings of the sixth workshop on statistical machine translation, pp. 187–197, 2011.

[30] J. Brooke, "SUS: a quick and dirty usability scale," Usability evaluation in industry, vol. 189, no. 3, pp. 189–194, 1996.

[31] A. Bangor, P. Kortum, and J. Miller, "Determining what individual SUS scores mean: Adding an adjective rating scale," J. Usability Studies, vol. 4, no. 3, p. 114–123, May. 2009.

[32] J. Lewis and J. Sauro, "Item benchmarks for the system usability scale," vol. 13, pp. 158–167, 05 2018.

[33] J. Sauro and J. R. Lewis, "When designing usability questionnaires, does it hurt to be positive?" in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ser. CHI '11, DOI

10.1145/1978942.1979266, p. 2215–2224. New York, NY, USA: Association for Computing Machinery, 2011. [Online]. Available: https://doi.org/10.1145/1978942.1979266