ARIS: Autonomous Real-Time Interactive Social Robot

Submitted: 21st June 2025; accepted: 29th October 2025

Cesar Minaya-Andino, David Minango, Marcelo Zambrano

DOI: 10.14313/jamris-2026-016

Abstract:
The purpose of this study is to develop and evaluate ARIS (A Real-time Interactive Social Robot), based on a Turtle 4 platform aimed at improving human–robot interactions (HRI) on university campuses. ARIS combines a 3D printed social robot structure with a software architecture based on 2D LiDAR, odometry, and IMU sensors for navigation and mapping, in addition to a voice assistant structured in 3 stages: audio recording, transcriptional processing, reaction, and reproductive synthesis. Experimental results show that the success rate of ARIS is greater than 86.5%, maintaining high accuracy in navigation and obstacle avoidance. The system also offers performance consisting of voice interaction with a total reaction latency of 1500 to 2200 ms. Access to low-cost robotic platforms allows students and researchers access to practical training, customization, and validation in the development of new technologies in social robotics.

Keywords: *ARIS, social Robot, interactions, turtlebot*

1. Introduction

In recent decades, human-robot interaction (HRI) has been one of the main research topics in the field of social robotics. Studying how humans and robots interact in social settings provides key insights for creating robots that interact with people in a natural and empathetic way. In collaborative environments such as hospitals, schools, hotels, offices, and others, these robots assist with daily tasks by engaging in intelligent and responsive conversations. A social robot needs to capture emotions, body language, and language cues, thus becoming a reliable collaborator for both practical work and social relationships to build trust and maintain interest.

Adaptability is also a crucial aspect, as it allows them to integrate into various environments—supporting students, managing hotel guests, collaborating with employees in their daily tasks, or simply serving as a technological attraction.

Several studies have found a variety of environments to incorporate robots as social agents. It is the example of BRIGHTNESS, a bartender robot that gives you a personalized service and is dynamic, adapting to the interactions of customers [1]. Another study has explored the role of the teacher in math activities versus a social robot, tutee, where it was revealed that, the main limitations are the verbal interactions between teacher–student and robot–student, in addition to this feedback, a very important factor in education is that it is best provided by the teacher [2]. In the same way other authors have determined, that the emotional participation and the compression reading cognitive storytelling is higher in children of 5-7 years with the help of a social robot in a comparison to a human [3]. Even to understand the adaptation of the robots socially with children at an early age of 8-9 years, several authors have studied the anthropomorphism of children and their latent growth [4].

A social robot has the characteristic of being stationary or mobile. To be steady-you can reduce a feeling of connection to personal interaction and decrease the experience of social dynamics. By contrast, to be mobile you must have a level of autonomy to adapt in the environment. To integrate into society, these machines must function in complex environments even without direct human intervention. Recent studies have focused on cognition and reasoning, so that CASPER via the development of an architecture for cognitive artificial enables you to observe the actions of humans, understand, and collaborate on tasks in progress [5]. In another study, the authors have developed an architecture hybrid that uses a metaphor of the brain, and a formalist approach [6–8].

Apart from perception, localization and mapping are two key evaluative capabilities that allow a mobile robot to understand and act autonomously in its environment. Since the early 1990s, techniques have been developed to enable a robot to localize itself within an environment. SLAM (Simultaneous, Localization, and Mapping) became a very popular technique starting in the 2000s [9], where, with the help of a LiDAR optical distance sensor, the cross-section of the environment's geometric structure can be measured in order to create a map. With the same goal of improving the accuracy, robustness, computational efficiency, and speed of the SLAM technique, numerous researchers have developed and implemented various methods, such as LOAM, which provides LiDAR odometry computation [10]; ORB-SLAM2, which builds a more accurate map of the environment using a monocular, stereo, or RGB-D camera [11]; RTAB-Map, based on real-time appearance-based mapping with loop closure detection [12,13]; and S-PTAM, which uses stereo cameras and combines the accuracy of the parallel tracking and mapping approach from ORB-SLAM [14].

A key feature of mobile assistant robots is voice recognition, which enables them to understand and respond to human voice commands, making

human-robot communication more intuitive. This technology involves capturing audio through microphones, followed by noise reduction and converting speech to text using automatic speech recognition (ASR) systems [15, 16]. In previous research, various models have been proposed and evaluated to improve the robustness of ASR systems in noisy environments [17, 18]. Other authors have proposed combinations of bidirectional gated recurrent units (Bi-GRU) with convolutional neural networks (CNN) to operate in offline mode [19].

In summary, many interesting results have been reported that highlight the potential of social robotics and the advancements developed over the years. Social robotics brings together various technologies such as artificial intelligence, perception, human–robot interaction, automatic speech recognition, self-localization, computer vision, among others.

The objective of this work is to develop and implement, in a first stage, a social robot named ARIS (Autonomous Real-Time Interactive Social Robot) based on the TurtleBot 4 platform, carrying out human-robot interaction applications and integrating perception, localization, and mapping within the ROS environment, in order to perform tasks on a university campus.

While advanced social robots like Pepper and NAO have demonstrated effectiveness in HRI applications, their high cost ($25,000-$50,000) and closed architectures limit accessibility for educational institutions. Existing low-cost platforms like TurtleBot typically lack integrated social features (expressive structure, natural voice interaction). ARIS bridges this gap by demonstrating that effective social HRI can be achieved using affordable, open-source components while explicitly documenting the trade-offs and limitations.

The manuscript is divided into two parts: The first part includes the title, abstract, and keywords. The second part is the paper's main body including the conclusion section.

2. Related work

2.1. Evaluation of Mobile Robots Designed for Lab-Scale Applications

In the field of mobile social robots, robotics engineers usually choose between two paths: simulations or real hardware. Functional platforms allow testing with sensors in real fields, providing solid and reliable results, although they are more expensive. On the other hand, simulations only allow for quick, inexpensive, and risk-free experimentation with algorithms. The drawback arises when you want to transfer the simulation to hardware, as it is not possible to replicate all the details of the physical world. For this reason, many researchers and engineers have developed affordable educational robotics kits to streamline the creation and implementation of innovative functions.

Examples include MobileCharger, an innovative robot equipped with sensors, actuators, and a combined perception system [20]; Robotis, a humanoid robot that integrates sophisticated sensors and dynamic movement capabilities, ideal for academic research, locomotion algorithm development, human–robot interaction, and experimentation in autonomous robotics [21]; NAO, an autonomous, programmable bipedal robot known for its friendly, expressive design and equipped with 25 degrees of freedom, allowing it to perform natural and complex movements [22]; and Pepper, another humanoid robot built on a holonomic base, equipped with three omnidirectional wheels, developed by Aldebaran to welcome customers in retail environments [23, 24].

Numerous studies have employed the aforementioned platforms, as well as others, with the aim of conducting academic research. In the case of Pepper, methods have been proposed to improve its limited 3D perception capabilities [25]; in Industry 4.0, digital twins have been developed for interaction in smart homes [26]. With NAO, several studies have been presented on gesture recognition using videos recorded solely by the robot's built-in camera while performing clinical procedures [27]. For arm movements, algorithms based on Bayesian Networks (BN) have been proposed and developed to select the appropriate motion model and enhance precision to levels similar to that of humans [28].

On other platforms such as Robear, Ubtech Wassi, and Freivera—robots designed for healthcare, specifically for assisting and caring for the elderly—several studies have proposed improvements in localization and mapping using methods to obtain semantic information and dynamic selection strategies [29].

In terms of personal assistants, DARWIN-OP2 is also noteworthy; it uses reinforcement learning algorithms to detect poor posture in a group of students [30].

The use of robotic platforms can accelerate research and lower the entry barriers for new research groups. Several platforms have been mentioned previously, but there are numerous affordable options in different sizes. However, the more advanced and recently developed platforms are often out of reach for many research groups due to their high cost or closed-source nature. For this reason, there is a need to develop open-source platforms integrated with sufficient sensors and actuators to support research groups in various fields.

Among open-source platforms, we have IGUS, whose robot structure is entirely 3D printed highlighting a lightweight and visually appealing design [31]; NimbRo-OP2X which integrates computer vision, IMU sensors and an Intel processor with GPU, making it ideal for locomotion, perception, and automatic control areas [32]; iCub, standing 104 cm tall, specifically designed to support research in embedded artificial intelligence, with hands engineered to support sophisticated manipulation skills [33–35]; and finally, InMoov, a humanoid robot approximately 180 cm tall that uses accessible components such as

Arduino microcontrollers and servomotors. Its modular design allows it to be built in stages and expanded according to the resources and needs of research groups [36, 37].

2.2. Approach to Real-Time Mapping and Position Tracking

Simultaneous localization and mapping (SLAM) is an indispensable technology which allows wheeled robots to navigate and determine their position accurately. The SLAM algorithm procedure is to collect information from the mains sensors such as cameras, and lidar, along with data from the inertial measurement unit (IMU). With this information, a map is constructed and the robot locates itself within it. A popular choice for mobile robots, Hector-SLAM relies on laser scans and fast estimation methods to deliver high accuracy in real time. Unlike other SLAM approaches, it operates perfectly even without wheel odometry [38].

A SLAM system based on multi-sensor fusion is a robust solution for achieving more accurate and stable localization and mapping in dynamic environments. This system compensates for the individual limitations of each sensor to achieve greater robustness with better estimates. By integrating all the data using fusion algorithms such as extended Kalman filters and optimization graphs, the system operates without disturbances [39].

Visual SLAM is another technique used to construct a map of the environment while a mobile robotic system moves through the environment. Visual information comes from one or more cameras. Unlike traditional SLAM methods that work with sensors such as LIDAR or inertial data, Visual SLAM uses exclusively image data (stereo, RGB-D, or monocular). From this visual information, the system's trajectory is estimated. In environments where GPS sensor data are lost or lacking [40], this technique improves global positioning [41].

Lidar sensors have become a primary tool for data acquisition for map construction, aiding autonomous navigation. Data acquisition from a lidar sensor produces a point cloud containing spatial information (x, y, z), which is invaluable in applications such as object detection, path tracking, and scene reconstruction [42]. In the context of SLAM systems, the point cloud represents robust and accurate data that can be integrated with data from other sensors to improve autolocalization and map construction, even in complex dynamic environments [43].

Low-cost small mobile platforms have also been developed to test algorithms in robotics such as SLAM, autonomous navigation, computer vision, and more. Create is a good example, where many research groups have implemented social functionalities on this mobile base, such as a real-time human–robot interaction system using hand gestures [44], as an office assistant with voice recognition to interact with users [45], and also by proposing a VLM-Social-Nav system with real-time decision-making [46].

Figure 1. Structure of ARIS

3. General architecture of the robot system

The general architecture of ARIS consists of a user-friendly physical robotic structure. The programmable TurtleBot 4 platform was used as the base, with a 3D-printed frame designed to give the appearance of a social robot. A touchscreen is included for human-robot interaction, along with a conference speaker and microphone for automatic speech recognition. The structure is shown in Figure 1.

3.1. Software development environment

TurtleBot 4 is a mobile robotic platform designed to provide an accessible research system for prototyping and robotics training. The mobile base, IROBOT Create 3, is integrated with motors that provide mobility. It is also equipped with proximity sensors, an IMU sensor, a stereo camera, and a 2D LiDAR sensor that provides navigation data. The model uses a Raspberry Pi 4 Model B (4GB RAM) as the main computing unit. The system runs Ubuntu 22.04 LTS with ROS 2 Humble preinstalled.

TurtleBot 4 operates with two main computers: A Raspberry Pi 4B and an integrated Create 3 processor. To visualize sensor data, configure, and control the system, among other functions, we connect another Raspberry Pi 4 Model B (4GB RAM) running the same Ubuntu and ROS 2 as TurtleBot 4. In ROS 2, DDS is integrated as the default communication layer, offering advantages in scalability and performance. Real-time data exchange, sensor reading or control commands are managed using the Simple Discovery configuration, as shown in Figure 2.

3.2. ARIS structure

The structure was manufactured using 3D printing with Fused Deposition Modeling (FDM) and polylactic acid (PLA) filament. With the help of 3D design and Creality Slicer 4.8 software, the printing parameters were adjusted with a 25% internal infill, taking into account the strength of the structure. Figure 3 shows the robot's structure and main components.

The 3D-printed structure was produced using the Creality 3DPrintMill, which offers an infinite Z-axis

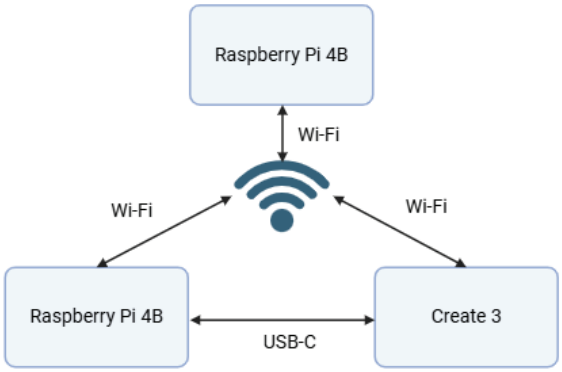

Figure 2. TurtleBot 4 Simple Discovery configuration for ROS 2 Humble

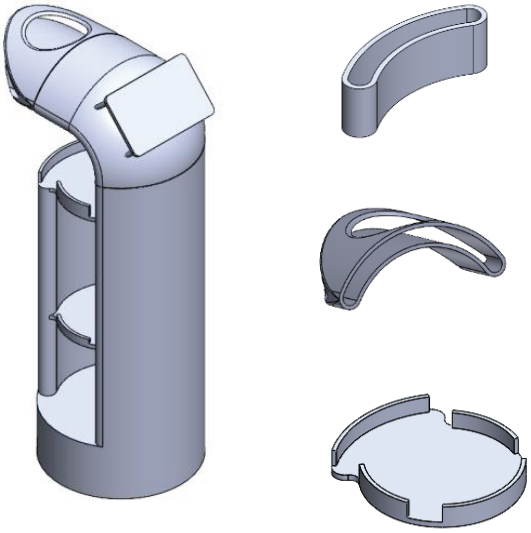

Figure 3. Design structure of the robot in 3D

and a 300 x 300 mm XY plane area. Due to the geometry and varying sizes of the robot components, it was designed and printed as a set of connectable parts rather than a single body. Each piece was oriented to maximize stability and adhesion to the surface of the adjoining piece. The complete assembly reached a weight of 3215 g.

3.3. Architecture of ROS Nodes

The ARIS architecture in ROS 2 is designed for natural and contextual human interaction. The system initiates interaction through the audio input unit, which captures the user's voice commands. This audio stream is processed by a transcription module that converts speech into text and interacts with a large language model (Gemini-1.5-pro) to enhance comprehension and generate harmonized interpretations. The core of the system lies in a central brain node that integrates decoded inputs and generative capabilities. ARIS can also respond using a speech synthesis tool that provides real-time feedback to the user. Additionally, the navigation module provides spatial awareness and mobility, allowing ARIS to physically

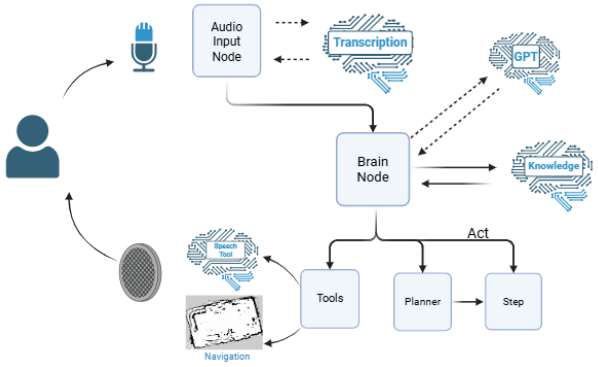

Figure 4. How ROS Nodes Work

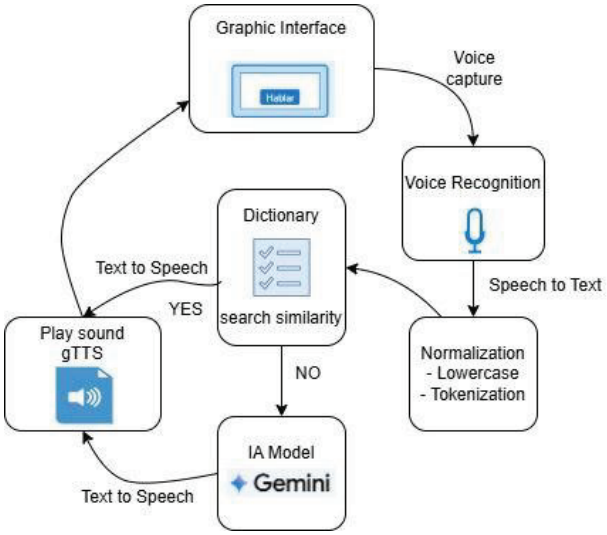

Figure 5. Block diagram assistant voice

interact with its environment. This architecture supports dynamic two-way interaction and enables ARIS to act as a helpful and conversational agent in real-world environments. The ARIS architecture in ROS is illustrated in Figure 4.

3.4. Voice assistant

The voice assistant implemented in ARIS, is organized into five modules as displayed in Figure 5, which are the capture of audio, transcription, processing of the input, the synthesis of the response and the playback of the audio, which is integrated to the graphical interface by clicking a button on the touch screen for human–robot interaction.

The voice assistant implemented in ARIS is organized into five modules that are capturing the audio, transcription, processing of the input, the synthesis of the response and the playback of the audio, which is integrated to the graphical interface by clicking a button on the touch screen for human–robot interaction.

Initialized ARIS, plays an audio file with a welcome message configured manually every 20 seconds, the audio capture was performed using the activation button in the screen called "Talk". When you press the button, it suspends playback of the first message and proceeds to the execution of the method of acquisition of audio. Using the library speech_recognition

automatically adjusts the detection threshold of environmental noise, to improve the quality of the signal in noisy environments. The capture of the audio is done by using the built-in microphone and is stored in an audio file temporarily.

The audio stored is sent to the service of Google Speech Recognition which makes the transcription from voice to text. The service returns a string in the format of a text, then the text is normalized by placing the text to all lowercase letters and using the technique of tokenization to break down the text into fragments of words or tokens. With tokens obtained is carried out a lexical analysis comparative look for the similarity of the tokens with the dictionary of words manually configured in a repository grouped by categories of topics such as (greeting, queries, institutional, control commands and diverse questions).

The detection of similarity between any of the tokens of the input and some of the words stored in the repository of dictionaries, generates the automatic playback of the answers that were previously configured by ensuring an efficient response with respect to queries that correspond to information in the university space. In the absence of match suggestions with the dictionary, the text of the question is sent in the format of a prompt to be answered by the generative model LLM employee who is the Gemini-1.5-pro.

The generative model uses a API_KEY generated from the services of Google AI Studio, additional function of the model is configured with instructions, in which it is established that the duties of a virtual assistant focused on giving answers related to the context of the education system in Ecuador, and in the case of the question requested was not appropriate in this context, is answered in a general way with the base model of a clear and precise way.

The response is generated to be pre-configured or coming from the generative model, converts text to audio format using the library Google Text-to-Speech gTTS, the which creates an MP3 file that is stored in a temporary directory, and is reproduced through the built-in speaker in ARIS using the player mpg123.

At the end of the audio playback, reset the interaction with the activation of the button, and the status message at the interface, indicating that it is ready to listen. This approach allows to ARIS to provide human–robot interaction using the format of a virtual assistant focused in giving answers that were previously configured on information relevant to the university space, combined with the flexibility of a language model to the generative for open consultation.

3.5. Functionality of ARIS

The algorithm described below controls the mobile robot's movement environment with predetermined locations, such as "Point A," "Point A," "Point A," etc. The robot waits for the user to press the touch button that allows it to select its destination. Once selected, the robot begins moving toward this destination. During the route, it visualizes potential obstacles and avoids them if necessary. If there are no obstacles, it will constantly update its movement.

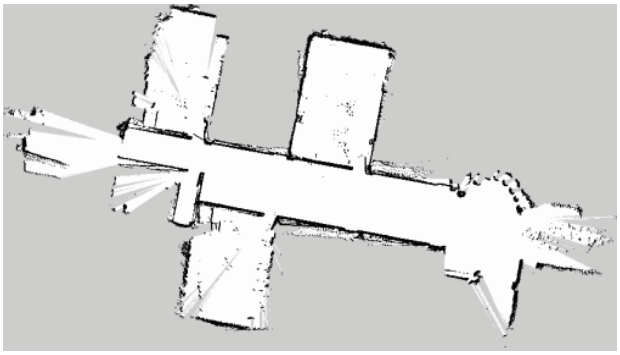

Figure 6. Local map

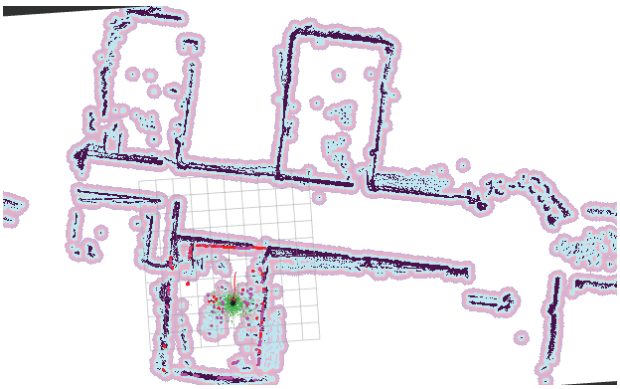

Figure 7. Aris performance with obstacles

When the robot reaches its destination, it interrupts its movement and displays a message indicating that it has arrived. This process is repeated indefinitely so that the robot can obtain new route requests at any time. At the same time, it checks if the battery percentage is greater than 20%. If not, it interrupts its route and heads to the charging point until it reaches an optimal percentage for its operation.

4. EXPERIMENTAL APPROACH

The platform ARIS was tested several times during its development. Firstly, it was tested at laboratory scale and, finally, in a university environment, including offices and corridors. You could measure the accuracy in the achievement of the objectives of the system ARIS.

The platform ARIS was tested with multiple users located in different points within the environment university, as illustrated in Figure 6.

Table 1 presents the default Cartesian coordinates for a set of five offices and the designated ARIS loading point. The X and Y values indicate the respective position of each location within the workspace.

ARIS autonomously determines efficient routes between designated points while performing tasks. The data in Table 1 serve as base positions for navigation.

ARIS is located at the entrance, which is part of the university campus. Users can approach and interact with it or request information about university authorities, academic programs, or simply general

```
start
waypoints := [“Point A”, ” Point B ”, ” Point C ”, ” Point D ”, ” Point H ”]
current_position := ” Point H ”
destination := ””
is_moving := false
battery_level := 100
previous_destination := ””
while true do
   if battery_level < 25 then
      previous_destination := destination
      destination := ”charging station (h)”
      is_moving := true
end if
if battery_level < 70 and current_position = ”charging station (h)” then
      charge_battery()
      continue
end if
   if touch_button_pressed() then
      destination := select_destination(waypoints)
      is_moving := true
end if
   if is_moving then
   if detect_obstacle() then
      avoid_obstacle()
   else
      move_toward(destination)
      update_position()
      update_battery()
      if current_position = destination then
      is_moving := false
      if destination = ”charging station (h)” and battery_level < 70 then
       // stay and keep charging
       continue
      end if
      if destination = ”charging station (h)” and battery_level ≥ 70 then
       destination := previous_destination
       is_moving := true
   else
      display(”arrived at ” + destination)
    end if
  end if
 end if
end if
end while
end
```

Table 1. Default coordinates of the offices and location of ARIS

No.	Offices	Coordinate (X)	Coordinate (Y)
1	Office 1	-0.25	-0.22
2	Office 2	-3.59	-0.41
3	Office 3	6.35	-7.09
4	Office 4	8.72	4.30
5	Office 5	0.59	-14.39
6	Default place of the ARIS to charge	0.0	0.0

information. At the same time, ARIS can guide users to the office they wish to reach.

During the route, ARIS is capable of detecting obstacles in its frontal path. When it detects a distance below a predefined threshold, ARIS stops its movement and performs a turning maneuver to avoid the obstacle, then resumes its trajectory toward the destination as illustrated in Figure 7.

Figure 7 shows the map created by SLAM, in which the dark areas represent permanent barriers such as walls, while the blue sections reflect the live zones that have been explored by the robot’s sensors. The purple color refers to the costmap inflation areas, which alert about the proximity of obstacles. The red band

Table 2. ARIS Performance of ARIS

User	Attempts	Questions Answered Correctly	Reached the Correct Coordinates	Avoided Obstacles (Yes/No)	Travel Time (s)	Success Rate
U1	2	Yes	Yes	Yes	52	100%
U2	2	Yes	Yes	IF	60	100%
U3	3	No	No	No	78	0%
U4	4	Yes	Yes	Yes	49	100%
U5	2	Yes	Yes	Yes	55	100%
U6	2	Yes	Yes	Yes	63	100%
U7	3	Yes	Yes	Yes	50	100%
U8	4	No	Yes	No	70	33%
U9	3	Yes	Yes	Yes	62	100%
U10	4	No	Yes	No	43	33%

Table 3. Performance of ARIS with inflation radius 0.5 m

User	Attempts	Questions Answered Correctly	Reached the Correct Coordinates	Avoided Obstacles (Yes/No)	Travel Time (s)	Success Rate
U1	2	Yes	Yes	Yes	52	100%
U2	2	Yes	Yes	Yes	59	100%
U3	3	No	No	Yes	63	66%
U4	4	Yes	Yes	Yes	49	100%
U5	2	Yes	Yes	Yes	55	100%
U6	2	Yes	Yes	Yes	61	100%
U7	3	Yes	Yes	Yes	50	100%
U8	4	No	Yes	Yes	64	66%
U9	3	Yes	Yes	Yes	60	100%
U10	4	No	Yes	Yes	53	66%

indicates the programmed route the robot will follow to reach its goal.

Autonomous navigation serves as a functional foundation in unknown environments which also allows the system to demonstrate adaptability in dynamic social settings. If ARIS detects a battery level below a predefined threshold, it decides to head toward a predetermined charging station and wait until it reaches a sufficient energy level to resume its activities.

To evaluate the acceptance of ARIS's functionalities, the system is considered successful if it correctly responds to users' questions and reaches the predetermined coordinates while avoiding both dynamic and static obstacles along the way. The travel time during autonomous navigation between the different coordinates was also evaluated to identify any errors.

The success rate of each event was monitored upon completing the described process, conducting 10 random experiments, with each user making a maximum of four attempts.

Table 2 summarizes the performance of ten users in a task-based assessment involving interaction with ARIS. The dataset in Table 2 also provides information on user-robot interaction under task constraints and shows performance as areas requiring improvement in obstacle avoidance and precise positioning in cases of poor performance. To evaluate the overall performance of each user during the navigation task, the level of success was defined based on three key criteria:

- Good, correct coordinates: whether ARIS navigated correctly to the destination.
- Avoid obstacles: whether ARIS avoided any obstacles along the route.
- Answer questions correctly: whether ARIS answered at least one question correctly during the test.

Each criterion received a binary result: 1 if the condition was met and 0 otherwise. The success level was calculated as the percentage

$$Sucess\ rate = \left(\frac{Criteria\ met}{3}\right) \times 100 \qquad (1)$$

Due to the limited processing power of the Raspberry Pi 4, which leads to occasional mapping errors or localization drift, conservative navigation heuristics and obstacle buffer zones were employed in the navigation system to improve the functional performance of ARIS. In the cost map, the inflation radius parameters were set to 0.5 m, providing a buffer zone around all detected obstacles. This increases safety margins.

In conservative heuristics for route planning and control logic, the movement speed of ARIS was reduced. This allowed for early stopping in large environments. Modifying these parameters minimized the likelihood of collisions, mapping errors, and, above all, computational overload on the Raspberry Pi 4 platform. The data obtained are shown in Table 3.

Table 4. Average latency

Stage	Component	Average Latency (ms)	Descriptions
A	ASR (Google Speech-to-Text)	480–650	Varies with noise and internet conditions
B	NLP (Gemini-1.5-pro)	600–900	Depends on prompt complexity and server load
C	TTS (gTTS)	300–450	Measured for responses < 50 words
Total Interaction	**End-to-end delay**	**1500–2200**	End-to-end delay per user query-response c

Table 5. Comparison of interaction latency between standard processing pipeline and FAQ dictionary-based optimization for common queries

Stage	Component	Standard Pipeline Latency (ms)	FAQ Dictionary Pipeline Latency (ms)
A	ASR (Google Speech-to-Text)	480–650	120–180
B	NLP (Gemini-1.5-pro)	600–900	240–360
C	TTS (gTTS)	300–450	180–270
Total Interaction	**End-to-end delay**	**1500–2200**	**600-910**

5. Results and discussion

The experimental phase was conducted to determine the functionality of the entire ARIS system. Functionality was obtained in a dynamic environment of a university campus, with and without external obstacles. The experimental data are shown in Table 2 and Table 3. The accuracy of the ARIS system exceeds 86.5% on average. In particular, an accuracy of 86.5% was achieved in the completion of each action, and ARIS failed only four times out of a total of 93 test cases during the experimental phase. Furthermore, ARIS completed the target point localization process with an accuracy of 90%. The failures were due to various reasons, such as improper operation of the robot's self-localization system and external noise. However, the ARIS did not fail when overcoming static and dynamic external obstacles. Therefore, the ARIS showed no difficulty in autonomously guiding itself while avoiding static and dynamic external obstacles. Overall, the ARIS system achieved its goal, based on user commands, not only in the absence of external obstacles, but also when there are static external obstacles in the university campus environment.

Figure 7 shows the relationship between battery level and system performance. This shows that as the battery level drops, system performance also decreases, but not linearly. When the battery level exceeds 60%, performance remains high with a slight drop. However, the 40% drop becomes much steeper and reaches a critical level below 20% battery level. This trend indicates that the system significantly reduces its performance to save power when the battery is low.

The Table 4 shows the average latency of each component involved in voice-based human–robot interaction which is divided into three stages: speech recognition, natural language processing, and speech synthesis. In Stage A, the speech-to-text conversion system (Google ASR) operates with an average latency of 480 to 650 ms, depending on background noise. Stage B processes natural language using Gemini-1.5-Pro, with a latency of 600 to 900 ms depending on the complexity of the request and server demand. In Stage C, the text-to-speech (GTT) engine converts responses of

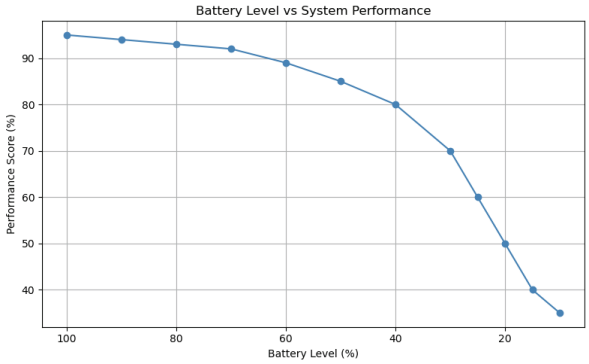

Figure 8. Battery Level vs System Performance

less than 50 words with a latency of 300 to 450 ms. Each user query has a latency in the entire interaction of between 1500 and 2200 ms which represents the total time elapsed from when the user speaks until they receive the response.

To mitigate the latency issues observed in Table 4, where total interaction delays ranged from 1500 to 2200 ms per query-response cycle, a lightweight caching system was implemented that stores frequently asked questions alongside their pre-computed responses. This optimization strategy creates a local repository of the most common user interactions, enabling rapid response retrieval without requiring cloud-based NLP processing or real-time text-to-speech synthesis.

As shown in Table 5, for cached queries, Stage B performs simplified NLP processing using pre-indexed semantic patterns, and Stage C retrieves pre-generated audio files from local storage, reducing synthesis time. ASR processes only the initial trigger phrase for pattern matching.

To measure positioning accuracy as the distance error in the path planning system, the Euclidean distance between the final estimated robot position and the true target position was used. The results in Table 6 show that increasing the inflation radius by 0.5 m significantly improved the accuracy and consistency of navigation. Almost all attempts reduced the calculated condition failure (e.g., from 0.08 m to

Table 6. Position error

Default Inflation Radius				Inflation Radius 0.5 m		
User	**Attempts**	**Reached the Correct Coordinates**	**Error Position (m)**	**Reached the Correct Coordinates**	**Reached the Correct Coordinates**	**Error Position (m)**
U1	2	Yes	0.08	Yes	Yes	0.06
U2	2	Yes	0.09	Yes	Yes	0.07
U3	3	No	0.50	No	No	0.40
U4	4	Yes	0.07	Yes	Yes	0.05
U5	2	Yes	0.08	Yes	Yes	0.06
U6	2	Yes	0.09	Yes	Yes	0.07
U7	3	Yes	0.08	Yes	Yes	0.06
U8	4	Yes	0.10	Yes	Yes	0.08
U9	3	Yes	0.09	Yes	Yes	0.07
U10	4	Yes	0.10	Yes	Yes	0.08

0.06 m U1 and 0.10 m to 0.08 m), while the average error dropped from 0.088 m (default) to 0.072 m with the inflation radius. Furthermore, the number of attempts to achieve high navigation accuracy (error $\leq$ 0.08 m) increased from five to eight. These improvements indicate that the additional buffer space around obstacles provided by the inflation radius improves route planning, reduces localization drift, and enhances overall system reliability, especially in an indoor environment with limited computing resources.

6. Conclusions

The errors occurred for various reasons, one of the main reasons was the background noise, unclear pronunciation, accent variety or variation in the speed of the speech, ASR incorrectly converts the audio into text, which provided misspelled words, substituted or omitted by others with similar sounds. These judgments directly affect the quality of the voice input for later misunderstanding. Another error is the understanding of the transcription of the audio; the model of the natural language processing failed due to limitations in their training.

The limit of processing of the Raspberry Pi 4 is another important factor in the development of the platform, ARIS. There are significant restrictions on intensive tasks simultaneously, such as localization and mapping simultaneously, the voice recognition and the algorithms of social interaction. While the Raspberry Pi 4 is equipped with a Cortex-A processor-72 of four cores with a clock frequency of 1.5 GHz, this does not comply with the requirements of the applications of robotics and high-performance. The algorithms SLAM Gmapping, require the processing of data from multiple sensors in real-time. These tasks require a high memory bandwidth, and a flow of the CPU/GPU of the Raspberry Pi 4 is constantly unable to provide, which causes latency and affects the accuracy of the mapping. The ASR also requires patterns of deep learning and fast processing. These models are located locally on the Raspberry Pi 4 and can result in a delayed or inaccurate recognition.

During the testing of the functioning of ARIS, it was observed that the Raspberry Pi 4 is subjected to prolonged computational loads and is prone to the thermal limitation. This not only reduces performance further, but also leads to unpredictable behavior during longer sessions of SLAM or iteration.

To tackle these issues and boost ARIS's reliability, we adopted cautious navigation strategies. A 0.5 m buffer zone was set around obstacles and the robot's speed was slowed in open spaces to allow quicker reactions. These changes reduce crashes, mapping mistakes and CPU strain.

The research combines proven robotic tools: SLAM, voice recognition, and language processing. The ARIS platform offers a clear methodology for building functional social robots on a tight budget, bridging the gap between high-end laboratory robotic platforms and the real needs of schools and small research laboratories. In addition, the performance of the Raspberry Pi was optimized by adjusting the requirements of the SLAM algorithms and using pre-recorded voice files to reduce latency, enabling smooth real-time operation even with modest hardware. These improvements allow ARIS to operate reliably with low-cost equipment that meets the needs of teaching and research.

AUTHORS

Cesar Minaya-Andino* – Departamento de Investigación, Instituto Tecnológico Superior Rumiñahui, Sangolquí, 171103, Ecuador, e-mail: cesar.minaya@ister.edu.ec.

David Minango – Departamento de Investigación, Instituto Tecnológico Superior Rumiñahui, Sangolquí, 171103, Ecuador, e-mail: david.minango@ister.edu.ec.

Marcelo Zambrano – Departamento de Investigación, Instituto Tecnológico Superior Rumiñahui, Sangolquí, 171103, Ecuador, e-mail: marcelo.zambrano@ister.edu.ec.

*Corresponding author

ACKNOWLEDGEMENTS

This work was supported by Instituto Tecnológico Superior Rumiñahui through its 2023 research project funding program, ISTER-INV-PRO-D-049.

Data Availability Statement : The data and source code supporting the findings of this study are openly available in the GitHub repository at: https://github.com/cesarandresma/ARIS.

References

[1] A. Rossi et al., "BRILLO: Personalised HRI with a Bartender Robot," *Int. J. Soc. Robot.*, vol. 17, 2025; doi: 10.1007/s12369-025-01228-3.

[2] S. Ekström, L. Pareto, and Y.S. Ljungblad, "Teaching in a Collaborative Mathematic Learning Activity with and without a Social Robot," *Educ. Inf. Technol.*, vol. 30, no. 1, 2025, pp. 1301–1328; doi: 10.1007/s10639-024-12926-2.

[3] S.S. Yeung, M. Ma, and T.T. Law, "Let's listen and Tell a Story Together: Social Robot and Multidimensional Learning Engagement among Young Learners," *AI Brain Child*, vol. 1, no. 1, 2025, p. 5; doi: 10.1007/s44436-025-00006-2.

[4] R. Kühne et al., "How Does Children's Anthropomorphism of a Social Robot Develop Over Time? A Six-Wave Panel Study," *Int. J. Soc. Robot.*, vol. 16, no. 7, 2024, pp. 1665–1679; doi: 10.1007/s12369-024-01155-9.

[5] S. Vinanzi and A. Cangelosi, "CASPER: Cognitive Architecture for Social Perception and Engagement in Robots," *Int. J. Soc. Robot.*, 2024; doi: 10.1007/s12369-024-01116-2.

[6] M.Á. González-Santamarta et al., "A Hybrid Cognitive Architecture to Generate, Control, Plan, and Monitor Behaviors for Interactive Autonomous Robots," *Int. J. Soc. Robot.*, 2024; doi: 10.1007/s12369-024-01192-4.

[7] M.Á. González-Santmarta et al., "MERLIN2: MachinEd Ros 2 pLanINg," *Softw. Impacts*, vol. 15, 2023, p. 100477; doi: 10.1016/j.simpa.2023.100477.

[8] M.Á. González-Santamarta et al., "MERLIN a Cognitive Architecture for Service Robots," *Appl. Sci.*, vol. 10, no. 17, 2020, p. 5989; doi: 10.3390/app10175989.

[9] F. Dellaert and A.W. Stroupe, "Linear 2D Localization and Mapping for Single and Multiple Robot Scenarios," *Proceedings 2002 IEEE International Conference on Robotics and Automation* (Cat. No.02CH37292), 2002, pp. 688–694; doi: 10.1109/ROBOT.2002.1013438.

[10] C. Gonzalez and M. Adams, "Curvature Scale Space LiDAR Odometry and Mapping (LOAM)," *J. Intell. Robot. Syst.*, vol. 110, no. 2, 2024, p. 67; doi: 10.1007/s10846-024-02096-1.

[11] R. Mur-Artal and J.D. Tardos, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, 2017, pp. 1255–1262; doi: 10.1109/TRO.2017.2705103.

[12] M. Labbe and F. Michaud, "Online Global Loop Closure Detection for Large-Scale Multi-Session Graph-based SLAM," *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 2661–2666; doi: 10.1109/IROS.2014.6942926.

[13] M. Filipenko and I. Afanasyev, "Comparison of Various SLAM Systems for Mobile Robot in an Indoor Environment," *2018 International Conference on Intelligent Systems (IS)*, 2018, pp. 400–407; doi: 10.1109/IS.2018.8710464.

[14] T. Pire et al., "S-PTAM: Stereo Parallel Tracking and Mapping," *Robot. Auton. Syst.*, vol. 93, 2017, pp. 27–42; doi: 10.1016/j.robot.2017.03.019.

[15] J.-M. Valin et al., "Robust Recognition of Simultaneous Speech by a Mobile Robot," *IEEE Trans. Robot.*, vol. 23, no. 4, 2007, pp. 742–752; doi: 10.1109/TRO.2007.900612.

[16] S. Alharbi et al., "Automatic Speech Recognition: Systematic Literature Review," *IEEE Access*, vol. 9, 2021, pp. 131858–131876; doi: 10.1109/ACCESS.2021.3112535.

[17] C.T. Ishi et al., "A Robust Speech Recognition System for Communication Robots in Noisy Environments," *IEEE Trans. Robot.*, vol. 24, no. 3, 2008, pp. 759–763; doi: 10.1109/TRO.2008.919305.

[18] J. Davila-Chacon, J. Liu, and S. Wermter, "Enhanced Robot Speech Recognition Using Biomimetic Binaural Sound Source Localization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 1, 2019, pp. 138–150; doi: 10.1109/TNNLS.2018.2830119.

[19] S. Girirajan and A. Pandian, "Offline Automatic Speech Recognition System Based on Bidirectional Gated Recurrent Unit (Bi-GRU) with Convolution Neural Network," *J. Mob. Multimed.*, 2022; doi: 10.13052/jmm1550-4646.1869.

[20] I. Okunevich et al., "MobileCharger: An Autonomous Mobile Robot with Inverted Delta Actuator for Robust and Safe Robot Charging," *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation* (ETFA), 2021, pp. 1–8; doi: 10.1109/ETFA45728.2021.9613366.

[21] G. Vasilyev et al., "Walking Algorithm for Robotis Op3 Humanoid Robot with Force Sensors," *2019 12th International Conference on Developments in eSystems Engineering* (DeSE), 2019, pp. 20–23; doi: 10.1109/DeSE.2019.00014.

[22] A.D. Ames, "Human-Inspired Control of Bipedal Walking Robots," *IEEE Trans. Autom. Control*, vol. 59, no. 5, 2014, pp. 1115–1130; doi: 10.1109/TAC.2014.2299342.

[23] J. Lafaye, D. Gouaillier, and P.-B. Wieber, "Linear Model Predictive Control of the Locomotion of Pepper, a Humanoid Robot with Omnidirectional Wheels," *2014 IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 336–341; doi: 10.1109/HUMANOIDS.2014.7041381.

[24] A.K. Pandey and R. Gelin, "A Mass-Produced Sociable Humanoid Robot: Pepper: The First Machine of Its Kind," *IEEE Robot. Autom. Mag.*, vol. 25, no. 3, 2018, pp. 40–48; doi: 10.1109/MRA.2018.2833157.

[25] Z. Bauer et al., "Refining the Fusion of Pepper Robot and Estimated Depth Maps Method for Improved 3D Perception," *IEEE Access*, vol. 7, 2019, pp. 185076–185085; doi: 10.1109/ACCESS.2019.2960798.

[26] L. Cascone et al., "DTPAAL: Digital Twinning Pepper and Ambient Assisted Living," *IEEE Trans. Ind. Inform.*, vol. 18, no. 2, 2022, pp. 1397–1404; doi: 10.1109/TII.2021.3090363.

[27] G. Ercolano et al., "Gesture Recognition with a 2D Low-Resolution Embedded Camera to Minimise Intrusion in Robot-Led Training of Children with Autism Spectrum Disorder," *Appl. Intell.*, vol. 54, no. 8, 2024, pp. 6579–6591; doi: 10.1007/s10489-024-05477-z.

[28] Y. Wei, "A Comprehensive Approach to the Generation of Human-Like Arm Movements on Robot NAO," *IEEE Access*, vol. 8, 2020, pp. 172869–172881; doi: 10.1109/ACCESS.2020.3025532.

[29] C. Zheng, P. Zhang, and Y. Li, "Semantic SLAM System for Mobile Robots Based on Large Visual Model in Complex Environments," *Sci. Rep.*, vol. 15, no. 1, 2025, p. 8450; doi: 10.1038/s41598-025-90340-5.

[30] K.M. Ahmad Yousef et al., "Personal Assistant Robot using Reinforcement Learning: DARWIN-OP2 as a Case Study," *Intell. Serv. Robot.*, vol. 17, no. 4, 2024, pp. 815–831; doi: 10.1007/s11370-024-00540-7.

[31] P. Allgeuer et al., "The igus Humanoid Open Platform," *KI - Künstl. Intell.*, vol. 30, no. 3, 2016, pp. 315–319; doi: 10.1007/s13218-016-0448-6.

[32] G. Ficht et al., "NimbRo-OP2X: Adult-Sized Open-Source 3D Printed Humanoid Robot," *2018 IEEE-RAS 18th International Conference on Humanoid Robots* (Humanoids), 2018, pp. 1–9; doi: 10.1109/HUMANOIDS.2018.8625038.

[33] T.P. Tomo et al., "A New Silicone Structure for uSkin—A Soft, Distributed, Digital 3-Axis Skin Sensor and Its Integration on the Humanoid Robot iCub," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, 2018, pp. 2584–2591; doi: 10.1109/LRA.2018.2812915.

[34] F. Naveros et al., "VOR Adaptation on a Humanoid iCub Robot Using a Spiking Cerebellar Model," *IEEE Trans. Cybern.*, vol. 50, no. 11, 2020, pp. 4744–4757; doi: 10.1109/TCYB.2019.2899246.

[35] G. Lombardi et al., "Humanoid Facial Expressions as a Tool to Study Human Behaviour," *Sci. Rep.*, vol. 14, no. 1, 2024, p. 133; doi: 10.1038/s41598-023-45825-6.

[36] L. Fortunati et al., "Exploring the Perceptions of Cognitive and Affective Capabilities of Four, Real, Physical Robots with a Decreasing Degree of Morphological Human Likeness," *Int. J. Soc. Robot.*, vol. 15, no. 3, 2023, pp. 547–561; doi: 10.1007/s12369-021-00827-0.

[37] D. Suresh, F. Dax, and D. Borrmann, "Advancing Robotics with an Interactive and Musical Humanoid Robot Based on the InMoov Project," *2024 22nd International Conference on Research and Education in Mechatronics* (REM), 2024, pp. 237–243; doi: 10.1109/REM63063.2024.10735692.

[38] J. Ma et al., "A Multi-LIDAR SLAM Method Based on Hector-SLAM," *2024 China Automation Congress* (CAC), 2024, pp. 345–350; doi: 10.1109/CAC63892.2024.10865468.

[39] S. Huang et al., "A Robust 2D Lidar SLAM Method in Complex Environment," *Photonic Sens.*, vol. 12, no. 4, 2022, p. 220416; doi: 10.1007/s13320-022-0657-6.

[40] N. Abdelaziz and A. El-Rabbany, "LiDAR/Visual SLAM-Aided Vehicular Inertial Navigation System for GNSS-Denied Environments," *2022 5th International Conference on Communications, Signal Processing, and their Applications* (ICCSPA), 2022, pp. 1–5; doi: 10.1109/ICCSPA55860.2022.10019210.

[41] Q. Du et al., "Tightly-coupled Lidar-Visual-Inertial SLAM for Mobile Robot," *2024 International Conference on Image Processing, Computer Vision and Machine Learning* (ICICML), 2024, pp. 842–845; doi: 10.1109/ICICML63543.2024.10957948.

[42] J. Lin et al., "R ^2 LIVE: A Robust, Real-Time, LiDAR-Inertial-Visual Tightly-Coupled State Estimator and Mapping," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, 2021, pp. 7469–7476; doi: 10.1109/LRA.2021.3095515.

[43] Z. Wang et al., "SLAM Mapping of Information Fusion between Lidar and Depth Camera," *2022 International Conference on Image Processing, Computer Vision and Machine Learning* (ICICML), 2022, pp. 142–145; doi: 10.1109/ICICML57342.2022.10009758.

[44] R.C. Hsu et al., "Real-Time Interaction System of Human-Robot with Hand Gestures," *2020 IEEE Eurasia Conference on IOT, Communication and Engineering* (ECICE), 2020, pp. 396–398; doi: 10.1109/ECICE50847.2020.9301957.

[45] I. Diddeniya et al., "Human–Robot Communication System for an Isolated Environment," *IEEE Access*, vol. 10, 2022, pp. 63258–63269; doi: 10.1109/ACCESS.2022.3183110.

[46] D. Song et al., "VLM-Social-Nav: Socially Aware Robot Navigation Through Scoring Using Vision-Language Models," *IEEE Robot. Autom. Lett.*, vol. 10, no. 1, 2025, pp. 508–515; doi: 10.1109/LRA.2024.3511409.