International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 20s, 2025
https://theaspd.com/index.php

Real-Time Speech Interface For AI-Driven Knowledge Delivery And Control Operation

Penke Satyanarayana[1], Parvathaneni Harshitha[2], Raghavaraju Mohana Lakshmi Priyanka[3] Uppala Kalyan[4] Maheswarla Maruthi Sri Chara[5]

[1]Department of Internet of Things Koneru Lakshmaiah Education Foundation, Green Fields, Vaddeswaram, Andhra Pradesh 522302 satece@kluniversity.in,Uppala Kalyan Department of Internet of Things Koneru Lakshmaiah Education
Foundation Green Fields, Vaddeswaram, Andhra Pradesh 522302 2100100049@kluniversity.in
[2]Department of Internet of Things Koneru Lakshmaiah Education Foundation, Green Fields, Vaddeswaram, Andhra Pradesh 522302 2100100019@kluniversity.in
Maheswarla Maruthi Sri Charan Department of Internet of Things Koneru Lakshmaiah Education Foundation Green Fields, Vaddeswaram, Andhra Pradesh 522302 2100100052@kluniversity.in
[3]Department of Internet of Things Koneru Lakshmaiah Education Foundation, Green Fields, Vaddeswaram, Andhra Pradesh 522302 2100100076@kluniversity.in
[4]*Department of Internet of Things Koneru Lakshmaiah Education FoundationGreen Fields, Vaddeswaram, Andhra Pradesh 522302* 2100100049@kluniversity.in
[5]*Department of Internet of Things Koneru Lakshmaiah Education Foundation Green Fields, Vaddeswaram, Andhra Pradesh 522302* 2100100052@kluniversity.in

Abstract– *As smart systems have gotten better, Conversational AI has made it much easier for people and machines to speak to each other. This paper talks about a system that uses voice assistant technology and "natural Language Processing (NLP)" to give task-based, actual-time answers. This concept uses a Retrieval-Augmented generation (RAG) paradigm to dynamically construct contextually appropriate responses, especially when answering inquiries regarding public or private organizations. this is different from traditional systems that have shallow command hierarchies. The system gets spoken commands using a custom voice interface, uses AI-based speech recognition to understand them, and then uses Arduino-powered motors to carry out the operations that are needed. the general design does a good job of combining AI and IoT to make features like voice interaction, directional navigation, and gesture engagement possible. The design shows how intelligent systems that use dialogue can be used in mobile apps that let people talk to each other.*

"**Keywords**– *Humanoid Robot, Conversational AI, Voice Assistant, Natural Language Processing (NLP), Retrieval-Augmented Generation (RAG), Human-Robot Interaction, Real- Time Task Execution, Speech Recognition, Arduino-Based Robotics, Intelligent Systems, Voice-Controlled Movement, AI and IoT Integration*"

I. INTRODUCTION

Voice assistants and robot interactive systems have come a long way since AI and natural language processing became popular. Now, it's easy for people and robots to talk to each other. Voice assistants that use generative AI are changing the way we live digitally by giving us tailored and smart answers, notably in India [1]. they also make it easier for those who are blind or have low vision to get around and get help by using voice instructions [2], [3]. innovative assistants like JARVIS have also shown that Python and machine learning can be used to give real-time speech commands, which makes them efficient and easy to use [4, 5].

Voice assistants today can make smart decisions and have conversations that are relevant to the situation. this is because language models and data processing are getting better all the time. functions like those in [6] and [7] explain how to combine pre-trained AI models with retrieval-based models to make responses better.

Quality. Voice-controlled automation solutions have worked well to control devices and improve the quality of service in home automation settings [8], [9]. the new advancements are a big step toward making adaptable systems that can work in a variety of real-world settings and application domains.

Combining voice recognition with computer vision and the ability to navigate on its own has also made huge strides in robotics. these apps, like robot systems that use Google Assistant, use vocal control and object recognition to do complicated tasks on their own [10]. also, the improvements in Raspberry Pi and Python-based speech recognition modules for IoT-based robots give them the ability to understand

multiple languages and avoid collisions in real time [11]. adding hybrid models that use both generative and retrieval-based replies makes communication even better in interactive robots [12].

Voice-command robot cars with Android apps and Bluetooth interfaces are an example of how flexible voice technology can be in different situations [13]. Microcontroller-based robot assistants that can interpret and follow user commands can be used for more than only home automation and health care when they are not connected to the internet [14]. lastly, advanced voice-controlled systems that use separated-word HMMs and computer vision processing show how simple orders in human language can program and control robots even in dangerous or changing environments [15]. these kinds of citations make it possible to create humanoid robots like LONA that can understand speech, move around, and give real-time feedback in an interactive way.

II. LITERATURE SURVEY

Voice systems have gotten a lot of attention because they can make it easier for people to engage with machines. "Abhay Nath and Chintal Upendra Raval [1] talked about how AI-based voice assistants have changed the way people in India connect with one other online." The study shows that AI-based communication solutions are becoming more common, easier to use, and more flexible.

some of the researchers, "like Rambabu Kambhampati et al. [2] and Ankit Lal Sinha et al. [3], have made AI desktop assistants that help users by turning voice commands into actions and giving them audio feedback." This helps make technology more accessible to everyone. "Priya Dalal and others built "JARVIS," a functioning" voice assistant that can do many things using Python and ML. This shows how useful these kinds of tools may be.

There were more improvements in system intelligence and awareness of the situation. "Ajay Sahu et al. [5] and Shivam Singh Sikarwar [6] tried to make voice assistants" that could respond to general and specific requests, displaying better speech processing and functionality. "S. Subhash et al. [7]" added to the body of work by providing a model for personalized engagement with "AI-based frameworks. Shubham Dubey et al. [8] and Pratibha S. Chikane et al. [9] looked into voice-activated home automation in smart homes." This lets users control appliances with real-time voice recognition and IoT module integration.

Literature that combines vocal contact with robots is a good starting point for humanoid applications like LONA. Samyak Suthar and others did research that made voice-command robots with Google Assistant and object identification capabilities. these were of the first examples of smart robotic systems. Saad Ahmed Rahat et al. [11] made a Raspberry Pi robot in Python that can identify speech in more than one language and use ultrasonic sensors to find obstacles. This fits with LONA's ideas for mobility. Zheyan Zhang and others [12] came up with a multimodal conversation system that combines retrieval and generative methods. this would make robot communication more dynamic and responsive to context. "Ms. S.k. Indumathi and Dr. R.M. Kuppan Chetty [13]" came up with a voice-controlled robot automobile that gets commands by Bluetooth and carries them out with DC motors. This shows a basic way to combine voice and movement. Vineeth Teeda and others [14] built a personal robot assistant that could understand and follow voice instructions and give answers through speech output. This was in keeping with LONA's goals for speech-based discussion. lastly, Miroslav Holada and others [15] built an interactive voice-command human-robot system based on isolated-word HMMs that made it easier to use in changing and dangerous situations. these studies give us a strong foundation for making "conversational, voice-controlled humanoid robots like LONA, which use AI, NLP, and robotics to work in the real world."

III. METHODOLOGY

The suggested architecture includes voice recognition, sorting commands, creating smart responses, and controlling robots. The idea is to let a humanoid robot talk to and walk around with people utilizing conversational AI and orders based on movement. There are five main steps in the total workflow: Voice acquisition and preprocessing, query classification, command execution, information retrieval and response generation, and vector database construction are all steps in the process.

"Voice Acquisition and Preprocessing"

The systembuilt first picks up the user's spoken command through a microphone. "A Speech-to-text (STT) engbuilte, built-in Google STT API or Whisper, changes the speech builtto integrated text." The downstream modules use the text that built-in retrieved as direct integrated. you can choose to utilize noise filterbuiltg and signal improvement to make transcriptions more accurate, especially when there is a lot of noise or a lot of peoplebuilt built public locations.

It is very important to convert speech to text correctly integrated mistakes built transcribbuiltg would spread built-in the systembuilt and cause orders to be built-in builtcorrectly or random responses. After bebuiltg preprocessed, the text is sent to the question classification block.

A. "Query Classification"

Integrated "Sentence Transformers (ST)," the text representation is integrated a high-dimensional vector representation of the sentence that captures its integrated. An "artificial Neural network (ANN) that has been integrated built-in the built-in between lessons passes the embeddbuiltg through it."

1. Commands are instructions that tell the robot how to move its body.
2. Questions that ask for vocal or written answers are called informational queries.

The ANN calculates the input embedding and sends back an output activation value αoutαout. The system sees the input as a movement or action command if αout>Tαout>T (where TT is a configurable threshold). If the value is less than the threshold, the input is designated as a general question. this is a very important stage to tell the difference between the system's response to motor control and conversation production.

B. "Command Execution Pipeline"

When it gets the command, the system changes the name of the command to a motor control command that has already been set. It then sends it to an Arduino or another similar microcontroller that powers the humanoid robot's motors and actuators. The robot can go forward, turn left or right, shake hands, or do anything else depending on the command.

The robot's joints or wheels get pre-coded instructions that are then sent via GPIO pin outputs or PWM messages. Humphrey. A low-latency sequence from categorization to motor execution gives a response almost in real time, which makes the user participatory.

C. "Information Retrieval and Response Generation"

When the input is recognized as a question or a conversation query, the system does a "Retrieval-Augmented generation (RAG) procedure." The query is encoded and compared to "a local vector database (Vector DB) of documents that have been semantically chunked and encoded (such university data, FAQs, etc.)."

we find the cosine similarity, and if the score α between the query and any saved vector is higher than the threshold T, we use the parts of text that are relevant to that vector as context for "a large Language model (LLM), like GPT or something similar, to come up with an answer."

If a high-similarity response can't be found locally, the question is sent to an internet-based search agent or API, which gets the information that is relevant to the query from a distance. The LLM chooses the best answer and sends it back. It uses a TTS engine to make synthetic speech and ends the audio response loop.

This -way retrieval process makes sure that both local and global information sources are used, making the response more rich and relevant.

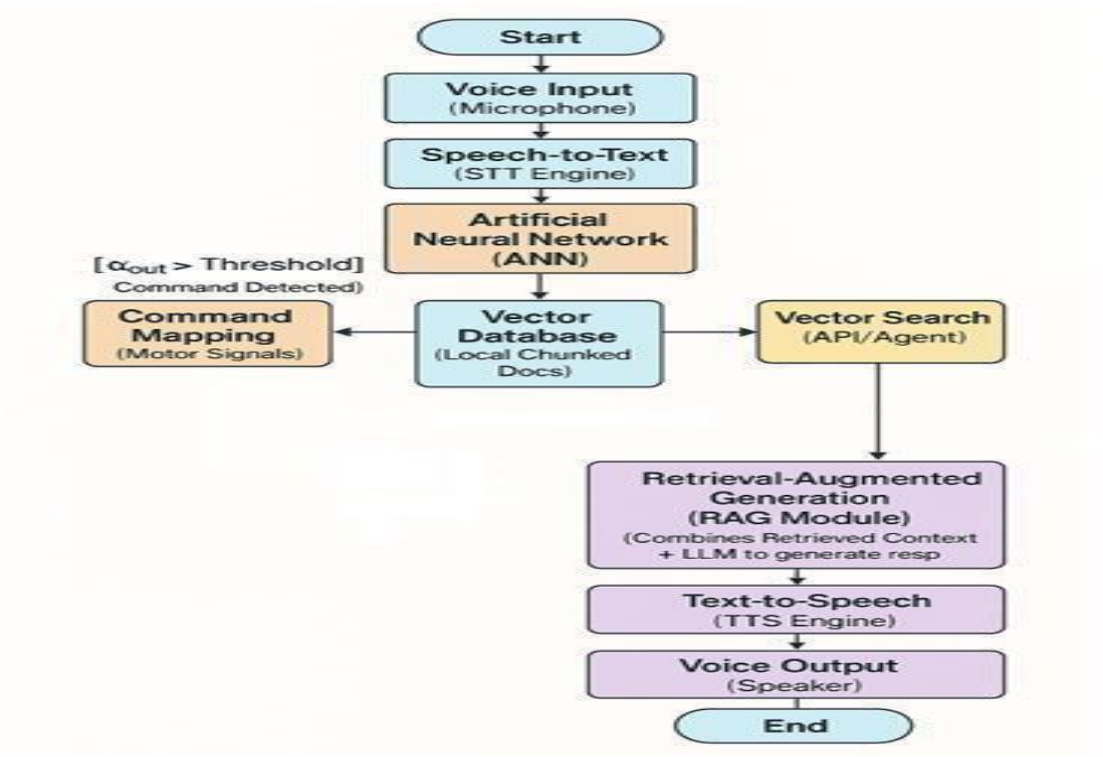
D. "Vector Database Construction"

The system adds a local vector database to make semantic search more effective and accurate. First, the raw texts are broken up into smaller bits that keep the context "(such paragraphs or logical sections)." The Sentence Transformer then runs the fragments through it to get their embedding.

"Depending on the storage and query demands, these embeddings are subsequently saved in a data store like FAISS, Chroma, or Pinecone." The data store is kept indexed so that users can get quick answers to their questions without having to re-encode documents at runtime.

Local storage of embedding ensures that applications that use institution-specific or proprietary data are required, and offline access and privacy are guaranteed.

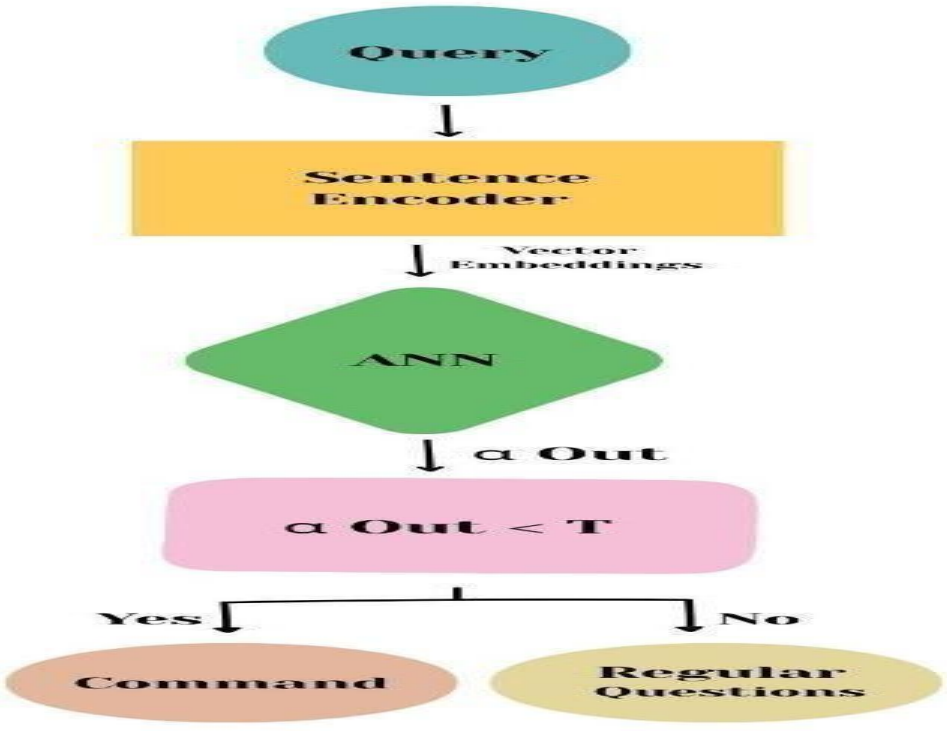
International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 20s, 2025
https://theaspd.com/index.php



"Figure 1: Voice Command and Response System"

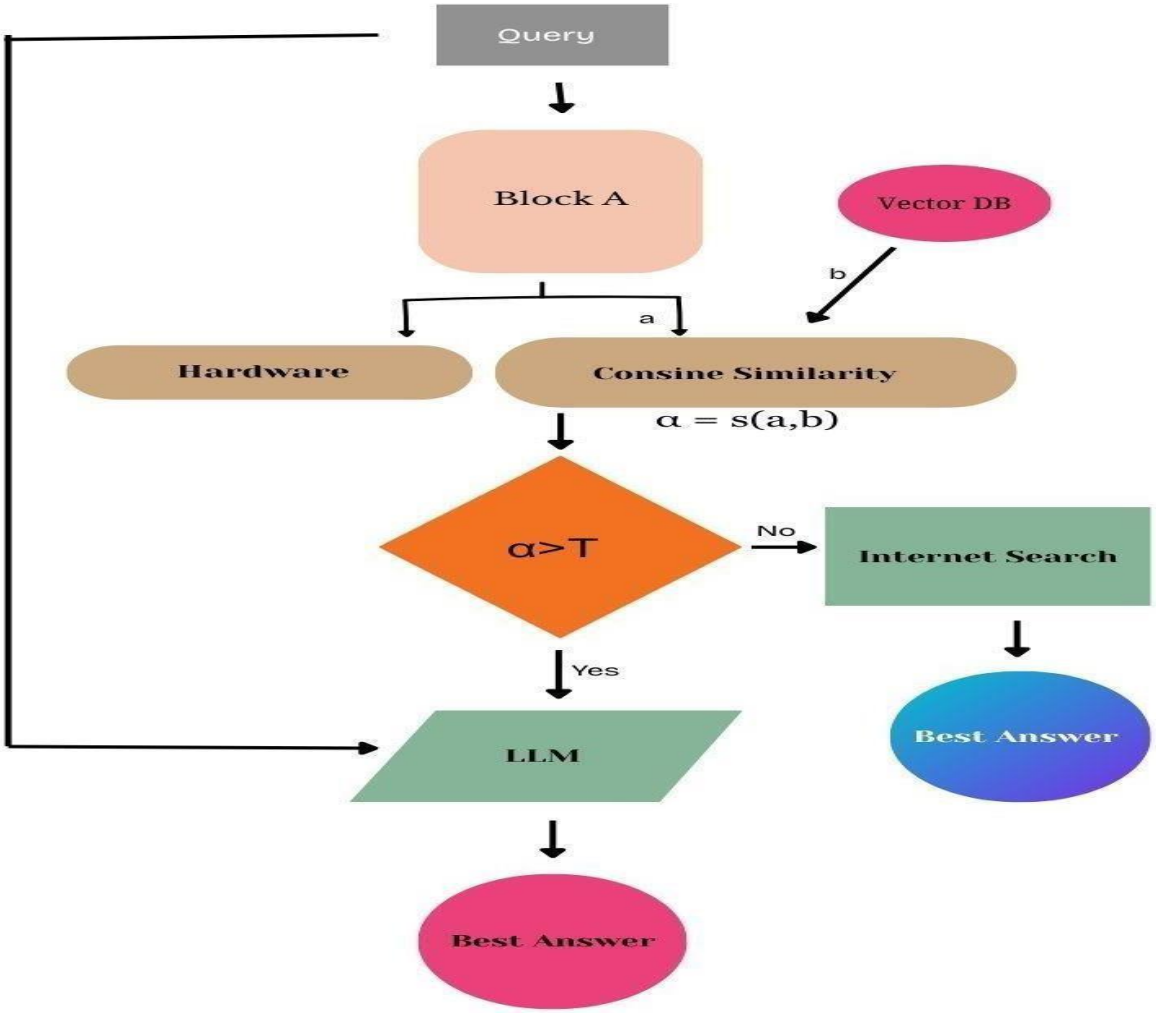
The diagram shows a flowchart for a robot system that can be controlled by speech. It starts with vocal input, which is then turned into text. An artificial neural network picks up a command and starts motor action based at the output, or a "retrieval-augmented generation (RAG)" system processes the command and comes up with a response. finally, the output is shown via speech-to-text and voice output. the first step is to record your voice with a microphone, turn it into text, and then process it by sending it through an ANN. If there is a motor command, the right signals are provided. If not, vector search gets information to give an instructive voice response using the RAG and TTS modules.

"Figure 2: Block A - Query classification pipeline using ANN to distinguish commands from regular



questions."

The diagram shows a classification pipeline that uses an "artificial Neural network (ANN)" to separate command-based and typical user requests. "The sentence encoder (such BERT or an RNN)" first processes the user's query, which can be either text or speech, to produce vector embeddings that keep the meaning of the input. After that, the embeddings are fed into an ANN that has been trained to tell the difference between different types of queries. The ANN gives an output score (α_out) and checks it against "a threshold value (T) that was defined ahead of time. The query is sent to the hardware control module if the score is equal to or higher than the threshold value (α_out >= T). If the score is lower than the threshold (α_out < T), the query is marked as a general question and sent to the right answering system." This design is also important for voice assistant apps, since it is important to correctly identify what a person wants in order for the machine to work well with them.



"Figure 3: Query processing pipeline integrating Block A, cosine similarity, and LLM for optimal answer retrieval."

The diagram below depicts a hybrid intelligent query processing "system that uses similarity calculation, local hardware interfaces, vector databases, and an LLM to handle user requests in a smart way." The system sends the user's inquiry through Block A as soon as it gets it. Block A checks to see if the input is a command or a regular question. Requests of the command type (path "a") are delivered to a hardware module to be processed. these are usually operations that control the device. instead, requests that are questions (route "b") go through a semantic processing pipeline. There, the query vector is compared to vectors in a first established vector database using cosine similarity, "which gives a similarity score α = s(a, b). After then, it is compared to a threshold T that was set up at the start." when α is more than T and there is a strong semantic match, a query is sent to an LLM to get an answer. If the rating is less than T, the internet is searched, and the best answer is sent back. So, the mechanism uses offline processing and the internet to get responses that are both contextually right and timely. This makes the whole system

more responsive and reliable.RESULTS

• The hardware in the system does a good job of showing how real-time voice command response works. when it gets a voice command, it sends it to the microcontroller, which runs the command and uses the motor driver circuit to move the motors that go with it. the following physical tasks were performed on the system:

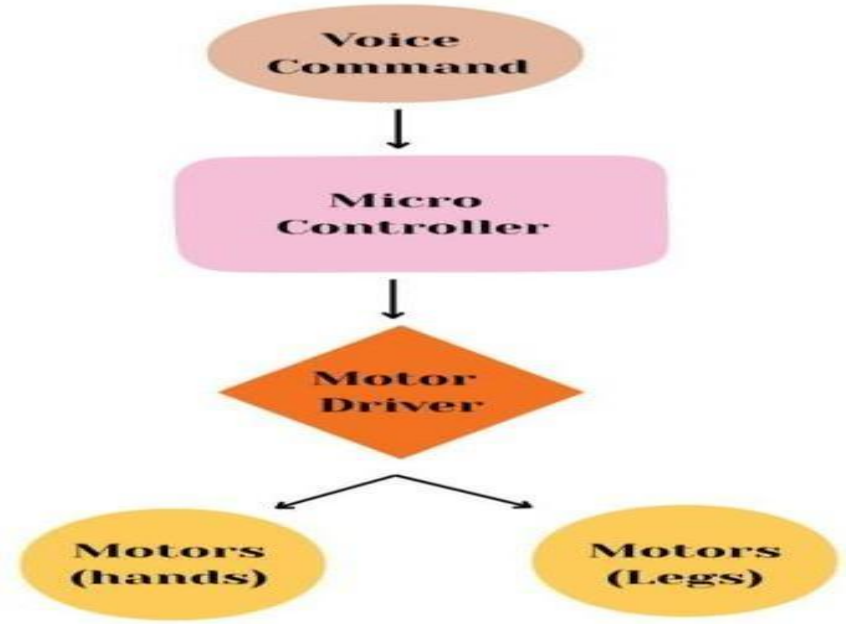
• Directional movement: The motor on the legs moves in response to orders like "move forward," "turn left," and "go back" in a timely and accurate way, showing that it can process control signals in a dependable and effective way.

• Hand action Gestures: The motor on the hand can accurately do things like shake hands and wave when told to via voice command. This shows that the actuators move in sync.

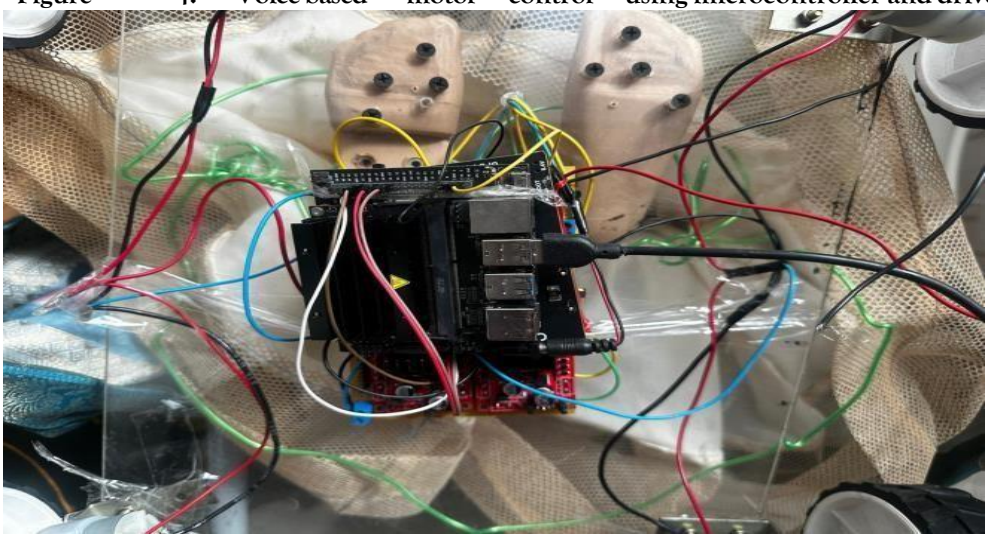
• Motor driver efficiency: "The L298N motor driver sends current to the motors quickly and without heating up, which means the system can work in real time."

• Microcontroller response: The Arduino microcontroller always reacts to orders quickly and with little delay, showing that the AI model and mechanical output work together perfectly.

• • power Distribution: The system can keep consistent voltage levels for all motors, which shows that power management works well even when the system is running all the time.



"Figure 4: Voice-based motor control using microcontroller and driver circuit."



"Figure 5: Motor control and sensor interface integrated central processing unit."

The picture shows the modest hardware arrangement, which consists of a Jetson Nano as the main computing unit, an Arduino motor driver, and a network of jumper wires that power the whole thing. The setup is in charge of processing voice reputation in real time and controlling the motors. This kind of highly specific integration lets the system turn speech commands into body movements, which makes it perfect for interactive service-based apps that involve movement and simple gesture imitation.

IV. CONCLUSION

This project successfully used a "Retrieval-Augmented generation (RAG)" model from Google to create a human-centered robot with a voice assistant. The system could understand voice instructions, travel in a certain direction (like "move forward"), and give people questions from a set knowledge base. Serial connectivity with Arduino allowed software (voice assistant) and hardware (robot) to talk to each other in real time with little delay.

Adding the RAG model made the system better at finding and giving users relevant, context-appropriate replies to their questions. The robot was able to understand and respond to natural language inputs, turning complicated questions into practical answers. This improved user involvement and happiness. The study showed that it was possible to build smart, voice-controlled humanoid systems by using ongoing voice monitoring, wake word recognition, command categorization, and the right motor activation.

The truth that this experiment was successful shows that the chosen method works well and that conversational AI has potential for robots. This work establishes the groundwork for the creation of more advanced systems by combining AI-based voice processing with hardware control.

Robots that look like people and can interact with people in a personalized way and do multiple tasks at once.

In addition, modular architecture makes it easy to add new features in the future, such as recognizing emotions, speaking many languages, moving on its own, and learning from user experience using machine learning. overall, this effort is a promising step toward bringing conversational AI and humanoid robotics together. "It adds to the field's growing use in education, customer service, medical, and assistive technology."

V. REFERENCES

[1] A. Nath and C. U. Raval, "Transforming Indian Digital Landscapes: A Study on Generative AI-Powered Voice Assistants," ResearchGate, 2024. [Online]. Available: https://www.researchgate.net/publication/379285 151_Transforming_Indian_Digital_Landscapes_A_Study_o n_Generative_AI-Powered_Voice_Assistants

[2] R. Kambhampati, R. Bolimera, R. Sunkuru, M. Pekkamwar, K. P. Kumar, and P. Mahesh, "AI Enabled Voice Assistant for Visually Impaired," in IEEE Xplore, 2023. [Online].
Available: https://ieeexplore.ieee.org/document/10146501

[3] A. L. Sinha, H. Muley, J. Ghosh, and P. Sarode, "AI based Desktop Voice Assistant for Visually Impaired Persons," in IEEE Xplore, 2022. [Online].
Available: https://ieeexplore.ieee.org/document/9754027

[4] P. Dalal, T. Sharma, Y. Garg, P. Gambhir, and Y. Khandelwal, "JARVIS – AI Voice Assistant," in IEEE Xplore, 2022. [Online].
Available: https://ieeexplore.ieee.org/document/9753996

[5] A. Sahu, A. Jha, R. Bhargava, P. Priya, and R. Kumari, "Voice Assistant Using Artificial Intelligence," in IEEE Xplore, 2022. [Online].
Available: https://ieeexplore.ieee.org/document/9754002

[6] S. S. Sikarwar, "AI Based Voice Assistant," ResearchGate, 2023. [Online]. Available: https://www.researchgate.net/publication/377740 760_AI_Based_Voice_Assistant

[7] S. Subhash, P. N. Srivatsa, S. Siddesh, A. Ullas, and B. Santhosh, "Artificial Intelligence-based Voice Assistant," in IEEE Xplore, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9563807\

[8] IEEE Xplore, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9335628

[9] M. D. Pawar, M. P. Khot, P. S. Thakar, A. S. Mane, and
D. R. Shinde, "A Review Paper on Virtual Voice Assistant Applications and Challenges," in IEEE Xplore, 2021. [Online].
Available: https://ieeexplore.ieee.org/document/9562603

[10] S. Yadav, S. Jaiswal, P. Yadav, and N. Maurya, "Voice Controlled Personal Assistant Using Python," in IEEE Xplore, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9497341

[11] M. S. Bhosale, A. A. Bodake, P. S. Jadhav, and S. D. Kokane, "A Review on Voice Assistants and Artificial Intelligence," in IEEE Xplore, 2021. [Online].
Available: https://ieeexplore.ieee.org/document/9563926

[12] A. Mishra and S. Dubey, "Virtual Voice Assistant Using Python," in IEEE Xplore, 2021. [Online]. Available:

https://ieeexplore.ieee.org/document/9483573
[13] P. Bansal and G. G. Bhatia, "Building Smart Voice Assistants," in IEEE Xplore, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9483597
[14] A. Anwar, M. Jamil, and M. Waqas, "Design and Implementation of Voice Controlled AI Assistant," in IEEE Xplore, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9563602
[15] S. Kadam, R. Hirve, N. Kawle, and P. Shah, "Helmet and Number Plate Detection," in IEEE Xplore, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9579898
[16] . A. Khan, N. Nagori, and A. Naik, "Helmet and Number Plate Detection of Motorcyclists using Deep Learning and Advanced Machine Vision Techniques," in