An IoT based Interactive Speech Recognizable Robot with Distance control using Raspberry Pi

[1]Saad Ahmed Rahat, [2]Ahmed Imteaj and [1]Tanveer Rahman
[1]Department of Computer Science and Engineering, International Islamic University Chittagong, Chittagong, Bangladesh
[2]School of Computing and Information Sciences, Florida International University, Miami, FL 33199, USA
Email: rahatcse37@gmail.com, aimte001@fiu.edu, shakil10vr@hotmail.com

***Abstract*—It is not convenient to use custom controller for operating a robot car and most easy way to control a robot is definitely with voice. In this paper, we have propounded a voice recognition based system where controlling process will be very much expedient to use. In our speech recognition based Robot car, anyone can easily control the system with his voice command and can also manage the robot to cover a guesstimate distance. For recognizing the voice, we are using Python speech recognition module and the most interesting thing is that the system can be configured with approximately 119 languages without much hazard. It can handle four different basic movement commands and the key feature of this system is that, if the user wants to run the system with a certain distance, the system can recognize that also. The system can detect human existence and the human detection part is handled by the ultrasonic sensor. The robot car will be automatically stopped by sensing obstacle if anything on its way and this obstacle detection process is handled by the ultrasonic sensor. All the system is controlled from a single board computer named Raspberry Pi 3 and the voice speech is recognized using Google speech reorganization engine.**

Keywords— Bangla speech recognition; Raspberry Pi; Sensors; Human Robot interaction.

I. INTRODUCTION

Controlling devices with voice is more convenient way to control things and in this paper, we have propounded a robot car controlled by voice which can do movement in any direction along with calculative distance to cover if it is stated by the user during command. If anyone need to control this robot for a custom path, it and say “Go 1 meter back”, the robot is intelligent enough to understand the distance to cover.We are using a custom equation where the main distance control is happening. At first, the system need to identify the RPM (rotation per minute) of the motor integrated within the robot car. Then we measured the wheel radius and calculated the circumference of the wheel. We have formulated an equation which is given in the system description part. We used two languages- Bangla and English to demonstrate our system. We are using a passive infrared sensor for the detection of human. The human detection process is also voice automated. Moreover, the system can also interact with the user and can reply in the specific language the user asked that robot. The author in [1] build a framework which can recognize Bangla speech but the system recognize bangle text by comparing with English text library which sometimes can generate wrong information. Another process is stated in [2], where they divide this process is two parts. The first stage is speech processing and the second stage is pattern recognition. The speech processing stage is based on a number of signal processing stages which are speech endpoint detection, windowing of the speech signal, filtering the speech samples so that there would be no noise left in the speech signal, linear predictive coding of speech, computing the cepstral coefficients from the LPC coefficients and then perform the vector quantization of the signal to obtain the codebook, which is used in the pattern recognition stage. But the process is quite lengthy and takes more processing time to generate speech. We have used Speech Application Program Interface (SAPI), which is an STT (speech to text) library and it is easy to manipulate the text and easy to use in the program.

II. RELATED WORKS

Several research works are already done on voice recognition process as it is very effective to apply in giving command and in this paper, we have proposed a system which can recognize voice in different languages through raspberry pi which is cost efficient and fit for a robot. We have trained that robot so that it can recognize commands, interact with the user and perform actions based on keywords. Some related works which have similiar approach to our proposed system is described and a marginable distinction as well as advantages of our system are showed in this section. The author in in [3] proposed a Tigal Voice recognition where the word capacity is not enriched but in our system there is no limitation of word capacity as we used google voice recognizer which word capacity is almost unlimited.

The author in [4] presented another voice recognition based approach which can only recognize six different languages. Another voice recognition based work is proposed in [5] where the recognition process is handled by the ATmega162 microcontroller. For recognition of voice, they used an ADC (Analog to digital converter) which can only handle the basic operations, like forward, backward, left and right. In our proposed system, all the recognition process is not only occuring in cloud but also can execute wide range of different languages of commands and also can estimate the distance to cover.

A voice record software is used in [6],in which the user creates the vocabulary words. The recorded words should be compressed using quick synthesizer 4 (QS4) from sensory and built. But in our system, vocabulary is stored in cloud with large scale word capacity which can b retieved in short processing time. Fezari *et al.* [7] proposed a system where the training process is very lengthy and the system can only recognize French Words. In our system, the system is configurable with any language and training process is simple.

978-1-5386-8524-2/18/$31.00 ©2018 IEEE

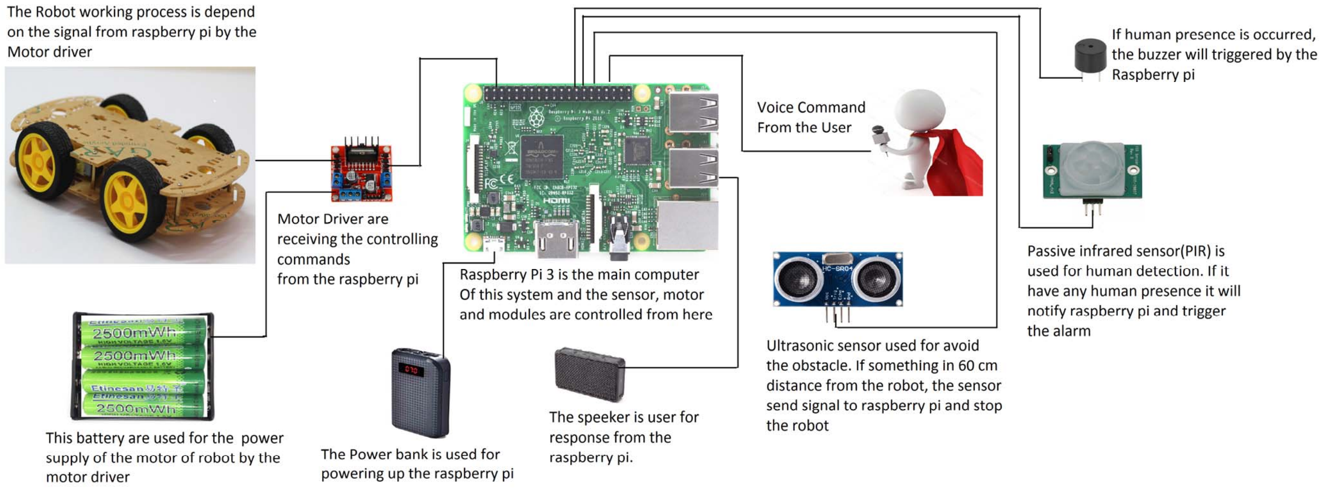

The Robot working process is depend on the signal from raspberry pi by the Motor driver
Motor Driver are receiving the controlling commands from the raspberry pi
Raspberry Pi 3 is the main computer Of this system and the sensor, motor and modules are controlled from here
Voice Command From the User
If human presence is occurred, the buzzer will triggered by the Raspberry pi
Passive infrared sensor(PIR) is used for human detection. If it have any human presence it will notify raspberry pi and trigger the alarm
Ultrasonic sensor used for avoid the obstacle. If something in 60 cm distance from the robot, the sensor send signal to raspberry pi and stop the robot
The speeker is user for response from the raspberry pi.
The Power bank is used for powering up the raspberry pi
This battery are used for the power supply of the motor of robot by the motor driver

Fig. 1. System architecture.

Sajkowski *et al.* [8] presented a robot, which can understand only 4 commands and there is no extra features rather than executing those four commands. Similarly in [9][10][11], the authors presented systems and those systems can only accept basic 2-3 commands. But in our system, the robot can perform basic 5 commands, left, right, go, back, forward and detect along with capturing image, interact with user, measurement of distance as well as can notify user in case of any unexpected situation. Arduino micro controller based voice recognition method is displayed in [12]. The system have all the basic controlling features but the processing time is slow because of the relatively weak processing unit of arduino. In [13], the authors proposed a complex voice recognizing process. They receive the voice then convert the voice analog to digital, then pre-process signal analysis, Mel Frequency Campestral Coefficients (MFCC) calculation and finally Pattern matching calculation using DTW. But the system can not be performed in real time in weak processing unit. Besides in [14], the user creates the vocabulary words using any voice record software. The recorded words should be compressed using quick synthesizer 4 (QS4) from sensory and built. But in our system, we don't need to create any training things. The system vocabulary is stored in cloud with large scale word capacity. We used the basic concept to configure raspberry pi with the system and configure IoT environment from [15], and [16].

III. SYSTEM DESCRIPTION

A. System Architecture

The propounded autonomous system (Fig. 1) uses Raspberry Pi 3 as main device and for capturaing the voice, we used a microphone which is connected with a USB to 3.5 mm connector. For the Human detection and avoid obstacle, we used PIR sensor and ultrasonic sensor respectively. The four motor are controlled and connected with the motor driver. The power supply of the raspberry pi comes from a power bank and for speech recognition, we used google speech recognizer. This module helps to generate the text from speech which has a higher accuracy than conventional recognition method’s techniques.

Our proposed recognition process works in several steps. First, the system needs to receive voice through microphone which converted that voice to digital signal by applying FLAC encoder process. Then the digital signal is applied to recognition algorithm which further produce text by applying deep learning algorithms. Figure 2 shows the overall process of our propoed speech recognition.

To transform speech or voice commands to on-screen text or a computer command, a computer or laptop has to go through several complex phases. When anyone speak, you generate vibrations in the air environment. The analog-to-digital converter (ADC) interprets this analog wave or signal into digital data that the computer can recognize. To do this, it models or digitizes the sound by taking exact dimensions of the wave at frequent intermissions. The system sifts the digitized sound to eliminate unwanted noise or anything is not needing, and sometimes to distinct it into different bands of frequency that frequency is the wavelength of the sound waves and heard by humans as alterations in any pitch. It also standardizes the sound, or regulates it to a constant volume or sound level. It may also have to be temporally associated. People don't all-time speak or talk at the same speed or same type, so the sound must be adjusted to particular match the speed of the template sound mockups already stored in the system's memory or databases. Next the signal is separated into slight segments as short as a few hundredths of a second, or even thousandths in the case of plosive consonant sounds -- consistent with stops produced by obstructing airflow in the vocal tract -- like "p" or "t." The program then matches these segments to known phonemes in the appropriate language. A phoneme is the smallest element of a language -- a representation of the sounds we make and put composed to form expressive terms. There are incompletely forty phonemes in the English language (changed linguists have changed opinions on the meticulous number), while other languages have extra or rarer phonemes.

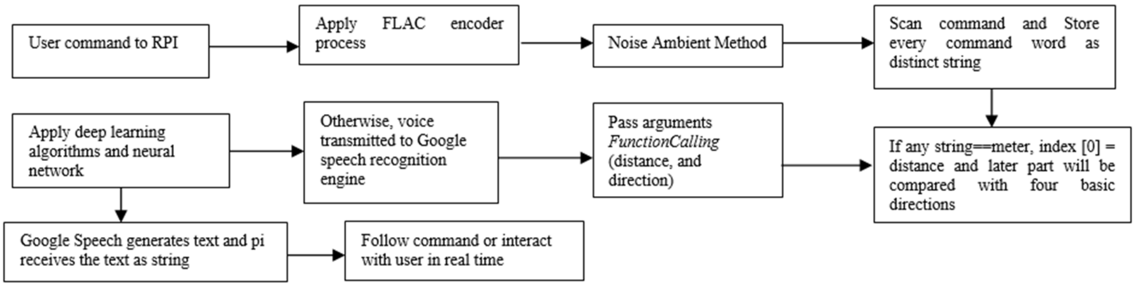

User command to RPI
Apply FLAC encoder process
Noise Ambient Method
Scan command and Store every command word as distinct string
Apply deep learning algorithms and neural network
Otherwise, voice transmitted to Google speech recognition engine
Pass arguments *FunctionCalling* (distance, and direction)
If any string==meter, index [0] = distance and later part will be compared with four basic directions
Google Speech generates text and pi receives the text as string
Follow command or interact with user in real time

Fig 2: Recognition process of speech

B. Noise embient method

The robot cannot recognize speech accurately after it starts hearing or listening for the first time. The recognizer_instance.energy_threshold property is set to a value that is too high to start off with, and then actuality accustomed lower spontaneously by dynamic energy threshold modification. Beforehand it is at a respectable flat, the energy threshold is so high that speech or voice is just measured ambient noise. The solution is the reduction of this threshold, or call the recognizer_instance.adjust_for_ambient_noise earlier, which will established a particular point or threshold to a decent value habitually.

C. Distance control

We have proposed a custom equation for distance control. For controlling the distance, we need to identify the RPM (rotation per minute) of the motor along with loads calculation and radius of wheel. In our working motor, the RPM (rotation per minute) is 180. The rotation of the motor per second =180/60=3

In our working motor wheel, the radius, r =3.3 cm

The circumference of the wheel = 2*3.31416*3.3 cm = 20.73 cm. So, in one rotation, the wheel crossed 20.73 cm distance

For 1 second the wheel crossed 20.73*3=62.20 cm [In 1 second the wheel rotates 3 times]

Eventually, 100 cm distance is crossed by the robot in 100/62.20 sec= 1.607 sec For weight balance, we need to add some extra time. After adding load [raspberry pi, battery, etc.], we have calculated that the time will be increased by 7% and required time to cover 100 cm will be 1.67.

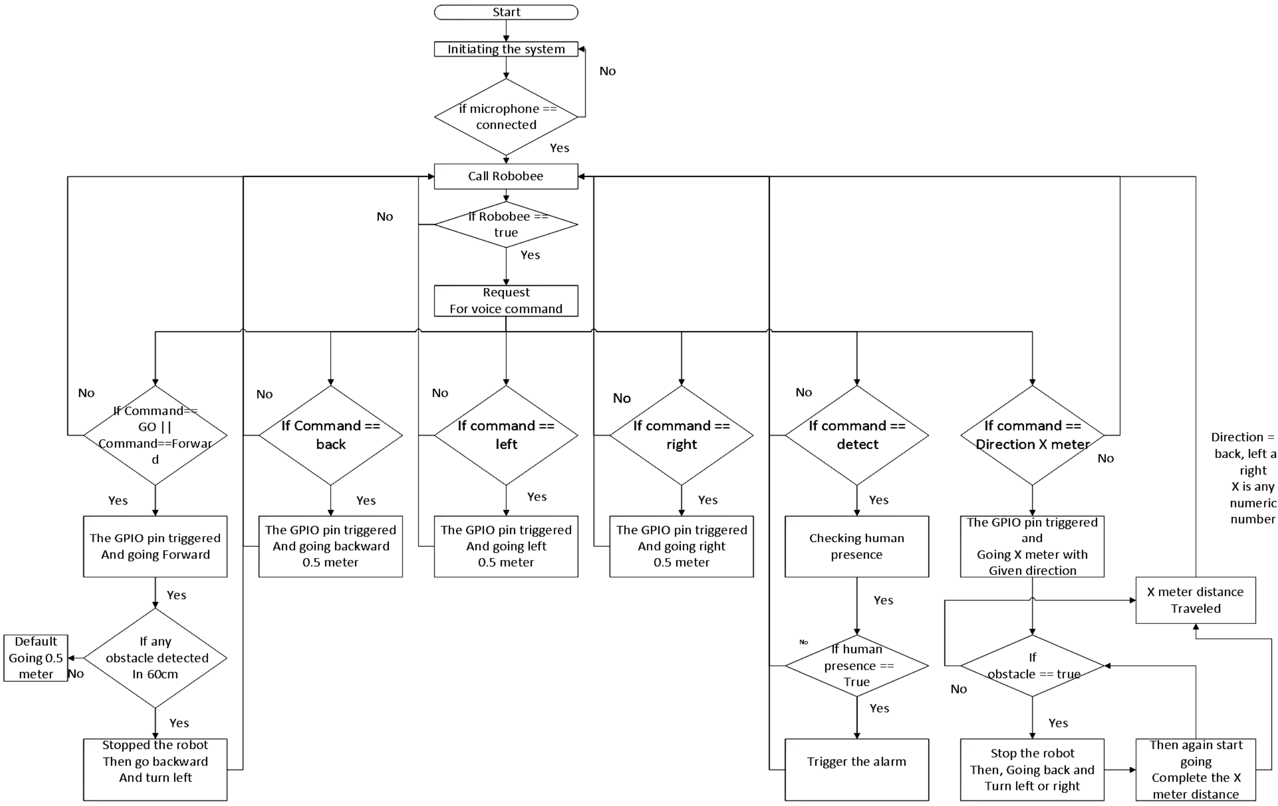

Start
Initiating the system
If microphone == connected
No
Yes
Call Robobee
If Robobee == true
No
Yes
Request For voice command
If Command== GO || Command==Forward
If Command == back
If command == left
If command == right
If command == detect
If command == Direction X meter
No
Yes
The GPIO pin triggered And going Forward
The GPIO pin triggered And going backward 0.5 meter
The GPIO pin triggered And going left 0.5 meter
The GPIO pin triggered And going right 0.5 meter
Checking human presence
The GPIO pin triggered and Going X meter with Given direction
Direction = go, back, left and right X is any numeric number
Default Going 0.5 meter
If any obstacle detected In 60cm
Stopped the robot Then go backward And turn left
If human presence == True
Trigger the alarm
X meter distance Traveled
If obstacle == true
Stop the robot Then, Going back and Turn left or right
Then again start going Complete the X meter distance

Fig. 3. Workflow of proposed system.

D. *Speech Recognition (Google Speech recognition engine)*

This process is handled by the Google speech recognition engine. But at first, need to install the python. After that may software module will be needed. So, that using a package manager is a right decision. We are using pip as package manager of python. For manipulating voice we need something. PyAduio helps us in this part. Receiving voice command with a microphone and manipulating the voice is handled by the PyAudio. When we receive raw voice there can be any noise or unwanted sound element. That's why we are using a module name noise ambient, which will remove the unwanted noise in our voice command. This module helps to increase the voice command accuracy rate. We are working on a Jessie operating system. In this type of operating system when any voice command received and send to somewhere, the main voice command can be lost or corrupted. For that, we need to use the FLAC encoder. This FLAC encoder will encode our voice command and make it stable. This will not be needed if anyone working in windows, Linux or Mac operating system. At first, the voice is captured by a microphone then it will send the voice is received by the raspberry pi. The raspberry pi sends the voice command to the google speech recognition engine. The STT (Speech to text) engine change the voice command into the text message. We manipulating this text in our system. This system can be configurable with 119 different languages. In our case, we were implementing two languages. For configuring the language we need to change the parameter of the language. There is a fixed data sheet for that. But sometimes it causes an encoding problem. We solved our encoding problem for Bangla speech recognition using perfect encoding for the raspberry pi.

E. *Human Detection (PIR sensor)*

For human detection is handled by the PIR sensor. If the sensor has the existence of any human, it will trigger a buzzer and also say that "human detected". If not detected then it will say not detected. If anyone says detect it will detect the human existence.

F. *Obstacle Detection (Ultrasonic sensor)*

This feature is handled by the ultrasonic sensor and if something in its way the robot will stop. If anything in 60 cm the robot will stop and turn left or right as define a condition.

G. *Response from Robot*

This part is handled by the TTS (text to speech) library named pytts3. For installing pytts3 we were using the pip package manager. But configuring the sound we need to use sudo raspi-config command and configure the port to sound card. If the configuration is wrong, the response will not work. We were using a speaker as an output of the response. The response of the robot depends on the command given by the user. The response depends on a particular command. The system is smart enough to decide the response. It can say how many distances it covers or what command it received from the user. This is a smart response system for the robot.

IV. SYSTEM IMPLEMENTATION

A. *Software Setup:*

1) *Operating System Setup:* As we are using Raspberry Pi for implementing the proposed system, we burnt the image file downloaded from raspberry.org/downloads in our micro sd card using Win32 Disk Imager software. We then enable ssh protocol and mentioned our WIFI ssid and password, for accessing it wirelessly. We entered the sd card, into our pi and powered it up.

2) *Operating System Setup and Configuring System:* After accessing raspberry pi successfully, we configured our raspberry pi for implementing further steps. We changed the default id and password, set up boot mode Desktop/Client, setup our locale, time zone and keyboard layout. We set up our wi-fi country to bn-bd. From interfacing we enabled remote gpio and i2c interfaces. As we are using USB sound card for taking voice input, we configured the USB sound card as default, so that it can take input from the correct source from it's next boot. After configuring the system properly, we reboot the system.

3) *Testing default voice source:* As we are developing a voice-controlled robot, we wanted to make sure that our system is taking input from our desired device by default. We connected our USB sound card to raspberry pi and connected mic into the USB sound. We then used async and around for taking input from the default sound card and were able to record our voice successfully.

4) *Installing Modules:* The core modules that we used are SpeechRecognition, PyAudio, PulseAudio, PortAudio, FLAC Encoder, Swig, Pyttsx3, RPi-GPIO. Among the

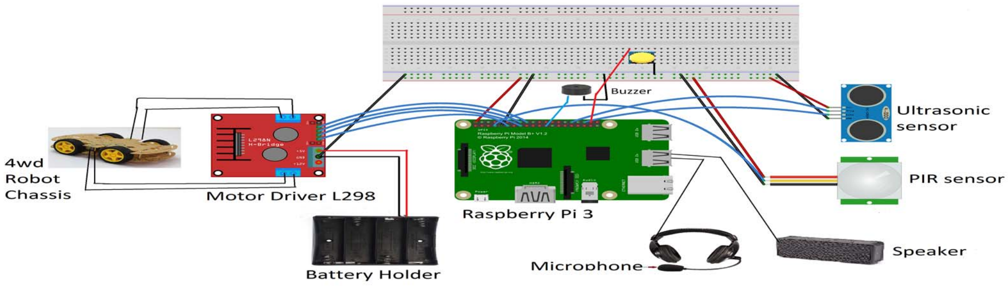


Fig. 4. Circuit diagram of the proposed system.

modules RPi-GPIO was pre-installed in Raspbian OS and others were installed from ubuntu stretch repository. As speech recognizer, we used both online and offline recognizer. We used Google Speech Recognizer as online recognizer and Pocketsphinx as offline recognizer. For offline English voice recognition, we used cmu-sphinx en-US (English) and en-HI (English-Hindi) speech dictionary and corpus. For offline Bangla recognition we used, SHRUTI-Bangla-Speech dictionary and corpus.

B. Hardware Setup:

1) *Preparing Chassis, Motor and Wheels:* We have used four-wheel driving (4WD) car chassis for implementing the system. We joined the chassis, motor, wheels together. The motor we used had RPM of 180 and wheels were of 6.6 cm diameter. We used these features for developing the distance measurement formula in our system. We positioned each part by adjusting weight and dimension, the system won't work properly with an inappropriate adjustment.

2) *Connecting Motor driver, Motors and Raspberry pi:* Motor driver is the media that communicates between the motors and raspberry pi. For implementing the system, we used L298N motor driver. We positioned it in the 2nd level of the chassis, so that it can maintain moderate distance between motors and the controller. For the left-side motors, we have connected the positive of motor1 to the negative of motor2 which we named joint A1 and A2. Then we have connected the joint A1 to output A1 of motor driver and joint A2 to the output A2 of the motor driver. For the right-side motors, we have connected the positive of motor3 to the negative of motor4 which we named joint B1 and negative of motor3 to the positive of motor4 which we named joint B2. Then we have connected the joint B1 to output B1 of motor driver and joint B2 to the output B2 of the motor driver. We fixed the raspberry pi at the top level of the chassis.

3) *Applying Distance Measurement fomula:* For making the system capable of reaching exact distance, we needed do some calculation. We decided to rotate our wheel for a certain period to reach the distance according to the command. For doing that, we need to calculate the time taken for reaching a meter. We calculated Time Per Meter by generating an equation as follows:

$$TPM = \frac{100}{\left(\frac{RPM}{60}\right) * 2\pi r} \pm 5\%\, err$$

where, 'RPM' is the Rotation Per Minute by the motors and 'r' is the radius of the wheel. 5% is the error rate that has been added for extra weight. As we found that, our RPM is 180 and radius of the wheel is 3.3 cm, so the TPM for our system was 1.6 sec. After adding the error rate, the value was 1.6 sec. We added the value in our program and continued.

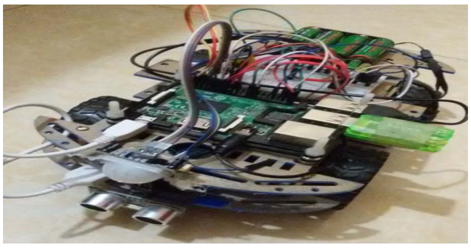


Fig. 5. Our proposed robot.

V. EXPERIMENTAL RESULTS AND EVALUATION

A. Voice Command Accuracy:

The voice command are tested with 25 persons and here we have displayed 5 among them. Each of the command is tested in ten to fifteen times to the robot and we have attained the following results which are based on real time testing. In this testing phase, at first we tested our robot by giving command in English. The result are given bellow:

TABLE I. English command Accuracy

Command	**Person-1**	**Person-2**	**Person-3**	**Person-4**	**Person-5**
Go	88%	85%	86%	88%	86%
Back	76%	80%	82%	80%	82%
Left	95%	95%	93%	95%	95%
Right	80%	84%	83%	80%	83%
Forward	51%	54%	54%	53%	54%
Detect	68%	75%	72%	70%	72%

After doing all the testing on English voice recognition of basic operation, the accuracy level is 77.83%.

Table 2 showed the test results which are acquired by giving command in Bangla.

TABLE II. Bangla command Accuracy

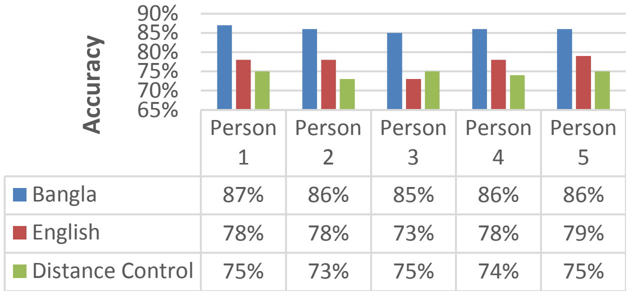
Command	**Person-1**	**Person-2**	**Person-3**	**Person-4**	**Person-5**
সামনে	95%	92%	94%	94%	95%
পেছনে	95%	95%	93%	94%	94%
ডানে	98%	96%	96%	96%	98%
বামে	58%	58%	56%	60%	54%

The accuracy level of the Bangla speech recognition is 85.5%. After that, we have tested our distance control feature and the results are displayed in table 3.

TABLE III. Distance Control Accuracy

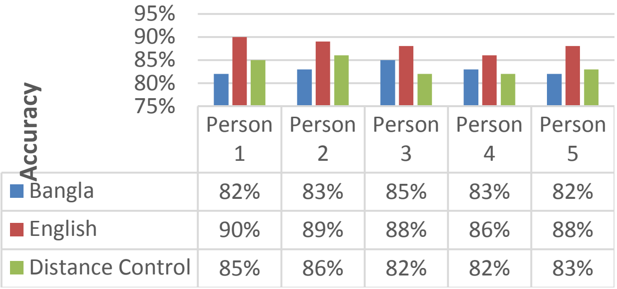
Command	**Person-1**	**Person-2**	**Person-3**	**Person-4**	**Person-5**
Go 3 meter	75%	72%	73%	73%	74%
Back 3 meter	75%	73%	78%	75%	74%
Go 10 meter	75%	72%	73%	73%	75%

The accuracy level of the distance control is the 74.33%.



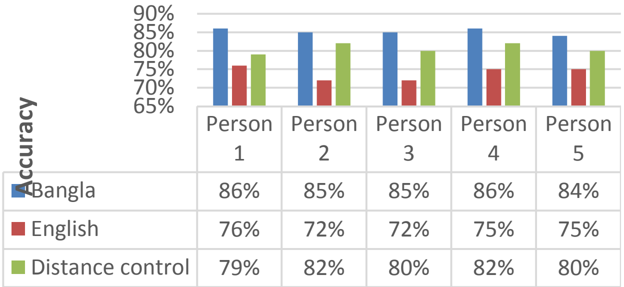
	Person 1	Person 2	Person 3	Person 4	Person 5
🟦 Bangla	87%	86%	85%	86%	86%
🟥 English	78%	78%	73%	78%	79%
🟩 Distance Control	75%	73%	75%	74%	75%

Fig. 6. Voice accuracy on raspberry pi in normal condition.



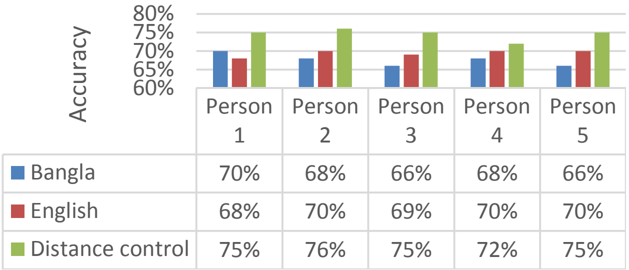
	Person 1	Person 2	Person 3	Person 4	Person 5
🟦 Bangla	82%	83%	85%	83%	82%
🟥 English	90%	89%	88%	86%	88%
🟩 Distance Control	85%	86%	82%	82%	83%

Fig. 7. Voice accuracy in calm environemnt.



	Person 1	Person 2	Person 3	Person 4	Person 5
🟦 Bangla	86%	85%	85%	86%	84%
🟥 English	76%	72%	72%	75%	75%
🟩 Distance control	79%	82%	80%	82%	80%

Fig. 8. Voice accuracy in room environemnt



	Person 1	Person 2	Person 3	Person 4	Person 5
🟦 Bangla	70%	68%	66%	68%	66%
🟥 English	68%	70%	69%	70%	70%
🟩 Distance control	75%	76%	75%	72%	75%

Fig. 9. Voice accuracy in wind environemnt.

CONCLUSION

In this paper, we discussed the latest technology to interact with a robot through voice command. Previously, a lot of works have been done but our proposed system presents a perfect declension of a robot and very swift execution of a command taking less processing time as we used very small sized microcontroller with efficient processing speed. As raspberry pi 3 is latest technology, so using bangla command and other hundred plus languages through this device is new and configuration of this device is always a challenge. We have deployed all the necessary implementation steps in this paper and generated an equation to cover any distance using voice command. We have tasted our system in different environments and have attained a very satisfactory result of accuracy. This robot can be used in any product marketing as well as in service related works as it can execute command as well as can track any object on its way. In future, we have a plan to integrate computer vision technology so that it can make decision by visualizing the environment.

REFERENCES

[1] Muslima, Umme, and M. Babul Islam. "Experimental framework for mel-scaled LP based Bangla speech recognition." *Computer and Information Technology (ICCIT), 2013 16th International Conference on*. IEEE, 2014.

[2] Paul, Anup Kumar, Dipankar Das, and Md Mustafa Kamal. "Bangla speech recognition system using LPC and ANN." *Advances in Pattern Recognition, 2009. ICAPR'09. Seventh International Conference on*. IEEE, 2009.

[3] Budiharto, Widodo, and Derwin Suhartono. "Intelligent service robot with voice recognition and telepresence capabilities." In *SAI Intelligent Systems Conference (IntelliSys), 2015*, pp. 301-304. IEEE, 2015.

[4] Haro, Luis Fernando D., et al. "Low-cost speaker and language recognition systems running on a Raspberry Pi." *IEEE Latin America Transactions* 12.4 (2014): 755-763.

[5] Thiang, Dhanny Wijaya. "Limited speech recognition for controlling movement of mobile robot implemented on atmega162 microcontroller." In *Computer and Automation Engineering, 2009. ICCAE'09. International Conference on*, pp. 347-350. IEEE, 2009.

[6] Fezari, Mohamed. "New speech processor and ultrasonic sensors based embedded system to improve the control of a motorised wheelchair." *Journal of Applied Sciences Research*5.10 (2009): 1750-1755.

[7] Fezari, Mohamed, and Mounir Bousbia-Salah. "A voice command system for autonomous robots guidance." In *Advanced Motion Control, 2006. 9th IEEE International Workshop on*, pp. 261-265. IEEE, 2006.

[8] Sajkowski, M. "Voice control of dual-drive mobile robots-survey of algorithms." In *Robot Motion and Control, 2002. RoMoCo'02. Proceedings of the Third International Workshop on*, pp. 387-392. IEEE, 2002.

[9] Kubik, T., and M. Sugisaka. "Use of a cellular phone in mobile robot voice control." In *SICE 2001. Proceedings of the 40th SICE Annual Conference. International Session Papers*, pp. 106-111. IEEE, 2001.

[10] Pleshkova, Snejana, and Zahari Zahariev. "Development of system model for audio visual control of mobile robots with voice and gesture commands." In *Electronics Technology (ISSE), 2017 40th International Spring Seminar on*, pp. 1-4. IEEE, 2017.

[11] Wang, Duojin, and Hongliu Yu. "Development of the control system of a voice-operated wheelchair with multi-posture characteristics." In *Intelligent Robot Systems (ACIRS), 2017 2nd Asia-Pacific Conference on*, pp. 151-155. IEEE, 2017.

[12] Mishra, Anurag, Pooja Makula, Akshay Kumar, Krit Karan, and V. K. Mittal. "A voice-controlled personal assistant robot." In *Industrial Instrumentation and Control (ICIC), 2015 International Conference on*, pp. 523-528. IEEE, 2015.

[13] XiaolingLv and Minglu Zhang and Hui Li. Robot Control Based on Voice Command. 2008 IEEE International Conference on Automation and Logistics Year: 2008 Pages: 2490 – 2494.

[14] Fezari, Mohamed. "New speech processor and ultrasonic sensors based embedded system to improve the control of a motorised wheelchair." *Journal of Applied Sciences Research*5.10 (2009): 1750-1755.

[15] Imteaj, A., Rahman, T., Hossain, M.K., Alam, M.S. and Rahat, S.A., 2017, February. An IoT based fire alarming and authentication system for workhouse using Raspberry Pi 3. In *Electrical, Computer and Communication Engineering (ECCE), International Conference on* (pp. 899-904). IEEE.

[16] Imteaj, Ahmed, Tanveer Rahman, Muhammad Kamrul Hossain, and Saika Zaman. "IoT based autonomous percipient irrigation system using raspberry Pi." In *Computer and Information Technology (ICCIT), 2016 19th International Conference on*, pp. 563-568. IEEE, 2016.