Proceedings of the Fourth International Conference on Inventive Systems and Control (ICISC 2020)
IEEE Xplore Part Number: CFP20J06-ART; ISBN: 978-1-7281-2813-9

Human Robot Interaction on Navigation platform using Robot Operating System

Rajesh Kannan Megalingam
Dept. of Electronics and Communication Engineering
Amrita Vishwa Vidyapeetham, Amritapuri, India
rajeshkannan@ieee.org

Motheram Manaswini
Dept. of Electronics and Communication Engineering
Amrita Vishwa Vidyapeetham, Amritapuri, India
motherammanaswini30@gmail.com

Jahnavi Yannam
Dept. of Electronics and Communication Engineering
Amrita Vishwa Vidyapeetham,
Amritapuri, India
jahnaviyannam@gmail.com

Vignesh S Naick
Dept. of Electronics and Communication Engineering
Amrita Vishwa Vidyapeetham,
Amritapuri, India
vsnayak160@gmail.com

Gutlapalli Nikhil Chowdary
Dept. of Electronics and Communication Engineering
Amrita Vishwa Vidyapeetham,
Amritapuri, India
g.nikhilchowdaryatp@gmail.com

Abstract **- In the last few decades, automation has been one of the leading and emerging technologies to develop sophisticated and robust robotic solutions for various applications like military, agriculture, transport, shopping, house, hospitals, etc. One of these emerging fields is service robotics which is implemented in the shop, home, and hospital environments. As these domains deal with the interaction between Humans and Robots, the concept called Human-Robot Interaction (HRI) has come into existence. In service robotics, voice is the most important mode of communication for HRI. This research proposes the implementation and integration of such HRI on an autonomous navigation system using the Robot Operating System (ROS). This paper also discusses the system architecture that is developed for this research.**

Keywords -- autonomous, navigation, mapping, speech, integration

I. INTRODUCTION

Symbrachydactyly is one of those congenital hand disorders characterized by abnormally short fingers that are sometimes webbed or conjoined. Approximately one out of every 32,000 births is suffering from Symbrachydactyly [3]. There are many physical challenges being experienced by people and Symbrachydactyly is one of those. By 2050, the world's population aged 60 years and older is expected to total 2 billion, up from 900 million in 2015. By Feb 5, 2018, 125 million people were aged 80 years or older which is certainly high now. These people are in need of assistance for their regular physical activities.

The joystick or app-driven robots couldn't solve this problem because of the obvious case that a person will not be able to use a joystick or an app to operate the robot because of the physical and mental challenges. It would be helpful if a person is able to operate the robot using voice. This paper discusses an idea to develop a Voice operated Autonomous Service Robot that would serve the directed tasks for the old and disabled or physically challenged people to replace human assistance thereby minimizing human mistakes and labor. The integrated system performs tasks such as Human-Robot Interaction and Cooperation, Mapping and Navigation with Obstacle Avoidance in dynamic environments. Here the development of Autonomous Service Robot integrated with a voice that can autonomously navigate in the indoor environment without any human intervention is explained.

II. MOTIVATION and PROBLEM STATEMENT

Though technology has been improving, there is an increase in the number of physically disabled people every year due to many reasons such as Congenital disorders, Arthritis and other musculoskeletal problems, injuries sustained in accidents, etc. These people are in need of assistance for their regular daily life activities where there should not be any human mistakes such as negligence and laziness. So, the implementation of robots would solve the problem of assisting these people without human errors.

Here the design and implementation of the voice-based autonomous service robot that can be used in home and hospital environments to serve the physically challenged people are explained. The working of the robot and operating system is also explained.

III. RELATED WORKS

In recent years much research has been carried out on navigation, ROS, and voice. The research paper [8] explains how the robot behaves when it encounters an obstacle and the simulation of the shortest pathfinding algorithm in the Gazebo environment. Paper [5] deals with the integration of both navigation and MEMS IMU. In Paper [12], a wheeled robot with Arduino as their microcontroller is tested which is implemented using both mapping slam and hector slam algorithms where both methods are compared. Paper [6] presents customized techniques for autonomous localization and mapping of micro UAVs flying in complex environments. Papers [7] and [11] deals with the applications of navigation and its uses in the shopping and cleaning purpose. In this research paper [4], they developed an app where users are able to use the robot for their shopping purpose and it is connected to the bot via Bluetooth HC05 module. Paper [13] shows us that a dual robotic arm can be controlled using voice on the ROS platform. In [4], [10] papers researchers discuss how to operate the home appliances using an android app over a wireless network. Papers [14] and [1] shows the application of an android app with a wheelchair and for interaction with a random human.

IV. SYSTEM ARCHITECTURE

A. *Hardware Architecture*

Fig. 2 shows the complete infrastructure installed in the robot. A laptop is used as a processor in which complete software integration is done using ROS which serves as the master for the whole system. A Smartphone in which an android speech application is being run is used to receive the input speech commands and a speaker for the output speech. For Mapping and Navigation purposes LiDAR (Light Detection and Ranging) is used and MPU6050 (Inertial Measurement Unit) sensors along with a pair of Rotary Quadrature encoders (Enc1 and Enc2). The encoder gives the values proportional to the rotation of the motors where the speed is calculated whereas MPU6050 gives the values of change in orientation of the robot with respect to x, y and z axes. The data from encoders and IMU sensors are collectively called the Odometry data which is used to determine the location of the robot in the given indoor environment. The LiDAR gives the data about the obstacles in a single plane along the robot's path by continuous transmission and reflection of the Near-Infrared Laser light. Teensy 3.2 controllers, a 32-bit ARM processor is used to drive the motors, to manipulate the Odometry and laser data with BTS7960 43 Amp motor drivers (MD1 and MD2) and GR-EP 45mm 12v high torque planetary geared DC motors (M1 and M2) for the navigation purpose. A 3S 11.1v LiPo (Lithium Polymer) battery of 10,000mAh capacity has been used as the power supply and a 5v dc to dc step down buck converter to supply 5v to the encoders from the 11.1v battery source instead of using another external source. An emergency switch is also used to shut down the robot for safety purposes.

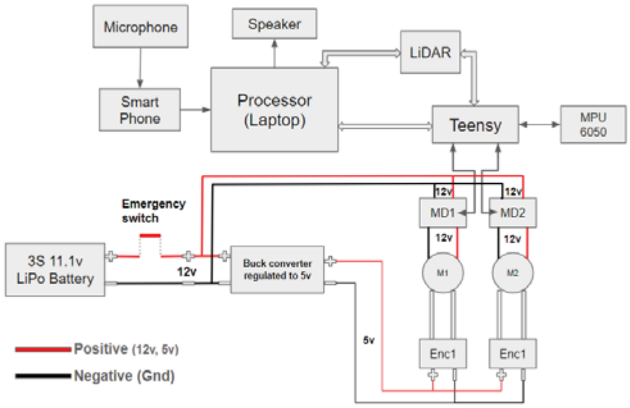

Fig. 1. Hardware Architecture

B. *Software Architecture*

The workspace consists of three main nodes which in turn have many nodes interconnected through different topics. The three main nodes are the following:

1. Speech
2. Mapping
3. Navigation

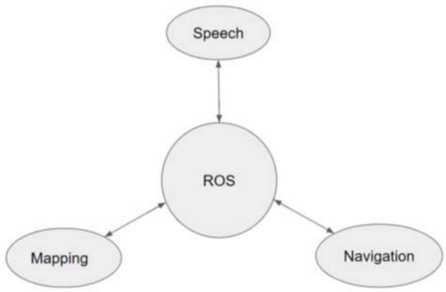

Fig. 2. Software Architecture

Fig. 2 represents the software architecture of the system which is entirely dependent on ROS that acts as a master to all those independent nodes. The further explanation of these nodes and master is given in the Integration section.

1. **Robot Operating System (ROS)**

ROS is a Robot Operating System which is empowering the world in the field of robotics. It is a meta-operating system built on the Ubuntu platform providing the services of an operating system like hardware abstraction, package management and message passing between processes. The reuse of codes in Robotics and automation is the primary objective of ROS. This is highly efficient as communication of different nodes is done through subscribing or publishing required topics instead of reading or writing to shared resources thereby reducing the complexity in multithreaded systems. The basic computation graph consists of concepts master, nodes, messages, topics, services, and bags.

1.1. Nodes

Nodes perform the computation of the algorithms and each robot control system consists of many nodes. For example, localization consists of one node, speech recognition possesses one node, etc. Using ROS client libraries like roscpp nodes are written.

1.2. Master

It provides registration services and naming for all other nodes in the ROS system. Without a master, nodes will not be able to find each other or exchange any sort of information.

1.3. Messages

It is a data structure containing typed fields and supporting standard types like integer, float, Boolean, etc. Nodes pass messages to communicate.

1.4. Topics

Topics are "named buses" through which nodes exchange messages. They comprise of publish/subscribe semantics. Nodes are not aware of the nodes with whom they are communicating with. Nodes subscribe to the corresponding topic for data and publish to the relevant topic. For each topic, many publishers or subscribers can be accommodated.

2. ***Speech Recognition***

Speech recognition is dynamic in nature and is complex to build, use and train data. As Smartphones are becoming part of our day to day lives, it becomes easy for an individual to communicate with the robotic system through speech. According to the "Statista Research Development" survey 2019, there are over 373 million people using smartphones in India. It will be easily accessible for any unskilled person to operate a system using his/her mobile phone.

2.1. Android speech application

Speech recognition of the system is developed using "Android Speech Recognition", an android application as in Fig. 3. The scripting language for the application is Java. It supports two languages English and German. The system turns active when the ON button in the application is selected. When the command from the user is given, it considers three best possibilities of which sentence it could be and locks the highest probability sentence as the final output using sequential recognition models such as Hidden Markov's Model (HMM) which is supported by Viterbi Algorithm to choose the most probable output.

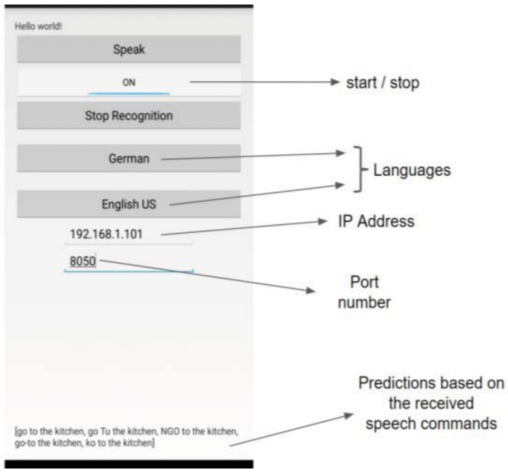

Fig. 3. Android speech application

2.2. IBM Watson Assistant

IBM Watson is developed by IBM's DeepQA project research team. It is a question-answer system, which is capable of answering the pre-defined questions for the system.

There are certain parameters like intents and entities which help to establish the required dialog. Intents are represented with symbol '# 'and entity is represented with '@ '. Example of intents and entities as mentioned in the IBM I'd.

Intent name: #direct_way

1. Go to
2. Move to
3. Proceed to Entity name: @location
4. Room1
5. Room2
6. Room3

After the required data created in the IBM Watson Assistance id, a dialog box is created as:

If assistance recognizes: #direct_way and @location

Assistance responses with Yes, going to @location.

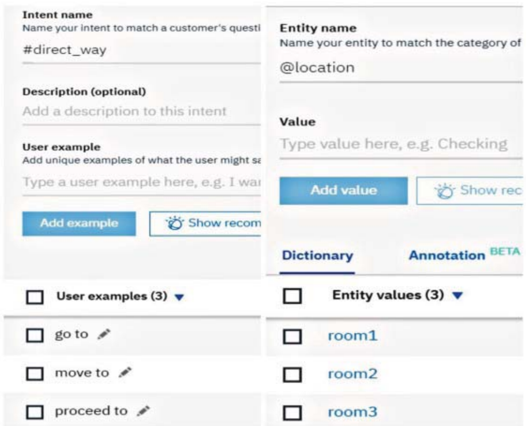

Fig. 4. Addition of Intents and Entities

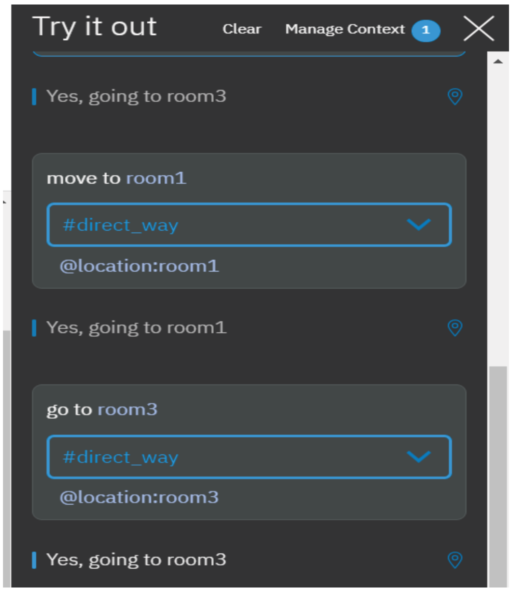

Fig. 5. Output response when speech data sent to IBM chatbot

The speech message data which is received from the ROS node is sent to the IBM Watson assistant where the data is analyzed and only the corresponding entity value is considered. This is because the entity represents which location the system is referred to move to. Each entity is given predefined location coordinates on the map, after receiving the coordinates the system plans for the optimized path to relocate to the specified point which is explained under the Navigation section below.

2.3. *Text to Speech conversion*

Text to speech (TTS) is the conversion of text into a speech that is more like a native speaker giving a simple reading. This conversion is done in 2 steps 1. Front end and Back end. In Front-end, first, all the numbers and abbreviations are converted into equivalent words called tokenization. Then each word is assigned with phonetic transcription and divides the words into clauses, phrases, and sentences. This is called a text-to-phoneme conversion. The output of the front end is the combination of tokenization and text-to-phoneme conversion gives a symbolic linguistic representation. The back-end converts the output of the front end into sound. Basically, this is referred to as a synthesizer.

3. *Navigation*

3.1. *SLAM*

SLAM (Simultaneous Localization and mapping) means generating and updating a map of the real unknown environment. This should simultaneously keep track of the robot position in the environment, which means the current position of the robot should be known for navigating in the real environment. This initially appears as a chicken-egg problem that can be solved by many algorithms such as G mapping, Hector_slam, google cartographer, core_slam, RGB-D slam. Based on the usage of sensors the various SLAM is differentiated. RGBD_slam generates 3D maps using RGB cameras whereas G mapping and Hector_slam use LiDAR sensor which gives laser_data for map generation.

3.2. *G mapping*

Odometry_data obtained from encoders, IMU sensors and laser_data obtained from Lidar are used in mapping to generate the map of the environment. Rao-Blackwellized particle filter (RBPF) occupancy grid mapping is the algorithm used for mapping. In this algorithm, particle-filtering is done thereby reducing the number of particles representing the SLAM posterior. Therefore, precise maps can be constructed from resampling as computational exertion is reduced.

3.3. *Localization*

After the generation of the map, for the robot to navigate autonomously in the environment, it is necessary that the robot should have known its current position on the map. To obtain the current pose of the robot Adaptive Monte Carlo Localization (AMCL) package is used. This determines the probabilistic location of the robot moving in 2D on the generated map as shown in Fig. 4. AMCL uses a particle filter to track the position of the robot by matching the points on the map to the laserscan_data. Whenever a float is observed, the current position of the robot is determined by the previously obtained odometry_data which is known as Dead Reckoning. The float is re-compensated by publishing a transform between the odometry frame and the map. AMCL maintains a set of high probable poses such that whenever the robot moves, the odometry data is compared to the expected poses according to the map. The probability of the pose increases with the consistency of the readings with respect to the map. The probability decreases if the readings are inconsistent. As the robot moves, the poses are obtained based on the odometry estimations. When a robot is simulated in the virtual environment, localization is not an issue because the starting position of the robot is always the origin. Whereas in the real world, localization plays an important role because of the dynamic applications.

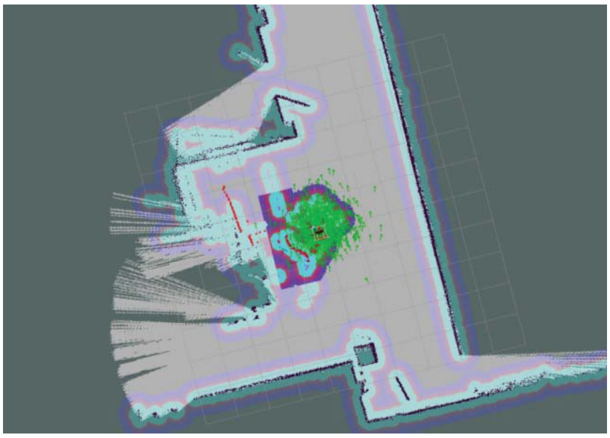

Fig. 6. Localization

3.4. *Navigation Stack*

After map generation and localization of the robot, navigation to the desired destination should be accomplished. For the navigation purpose, the move_base package is used which consists of multiple nodes running in parallel to produce the new sub-goals in order to avoid obstacles based on shortest path algorithms. The different nodes of move_base are:

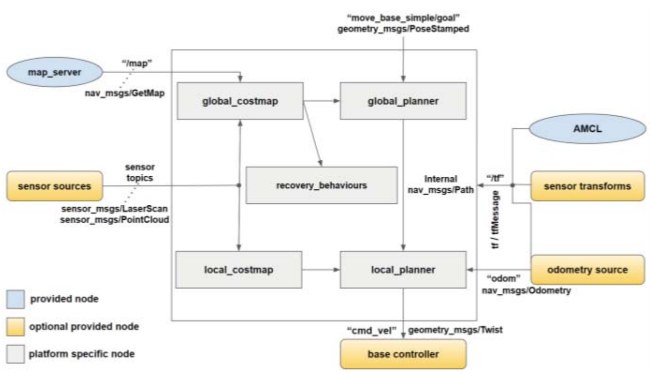

Fig. 7. Navigation stack

3.4.1. *Global planner*

This package provides an algorithm for planning an optimum path to the destination by using shortest path algorithms like Dijkstra's algorithm which is forwarded to the Local_planner.

3.4.2. *Local_planner*

Taking odometry data, sensor information and the path provided by Global_planner, angular and linear coordinates along the path are generated and sent through 'cmd_vel' topic to the controller that drives the robot accordingly.

3.4.3. *Recovery behavior*

If any dynamic obstacles are found, then the robot rotates 360° to find the gap. When the robot is unable to avoid the obstacles, the robot stops by giving an error indicating that "No Recovery behavior found".

V. SYSTEM INTEGRATION

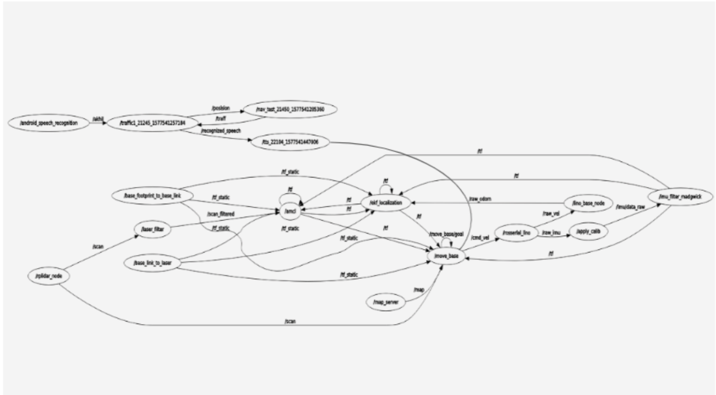

Fig. 8. rqt_graph of the integrated system

Fig. 8 depicts the rqt_graph of the entire system plotted in ROS indicates the communication between all the main nodes and intermediate nodes through different topics. Android app will be containing IP addresses of the Wi-Fi network and port number of the user's smartphone. The screen of the application has 5 buttons -Recognize, ON-OFFFF, Speak, German, English US. When the Recognize button is clicked on the android app, the app starts listening to the user through the microphone. This input is published to the PC for further processing. This process is done using Sockets. Sockets and the Socket API are used to send messages across a network. They provide a form of inter-process communication (IPC). The network can be a logical local network to the computer or one that's physically connected to an external network with its own connections to other networks. The obvious example is the Internet, which is connected via ISP. Here the text data will be converted to string data type on the topic 'Akhil'. The Pc is already equipped with a database containing keywords of a set of locations in a house mapped to coordinates of locations on the map as the entire 2D map will have coordinates with respect to the origin. The PC also has a set of verbs in the database like come, go, assist, etc... When the voice command is given as input, the sentence is fragmented into 2 categories - verb and location. All the remaining words which don't fall into any category are considered as unwanted words. Here "Location" is given the first priority and next is given to the "Verb". Input location and verbs are compared to locations and verbs in the database. If both get matched, the input is sent to the ibm_watson assistant through which desired speech output is obtained on the topic "recognised_speech". Also, the respective coordinates are published to navigation stack on the topic "position". Now the stack chooses the best path to reach the desired location and the robot is directed to the goal to be reached. If the goal is reached by the robot it displays "Hurray!! Reached the pose" and sends to the text_to_speech algorithm for speech output. If there is an error in reaching the destination, recovery behavior is observed to relocate its path. If no behavior is found “No recovery behaviors found” is displayed and sent to text_to_speech. If the input sentence doesn't contain any keyword from the database, then the sentence will be transferred to IBM Watson which is directed to respond as "No destination is given and the task cannot be done".

Location	Verb
Kitchen	Go
Bedroom	Pick
Dining area	Skedaddle
Hall	Come

Table. 1

A. Example

COMMAND - Go to the kitchen

This is divided into fragments like “Go”, “to”, “the”, “kitchen”. Now each of these words is compared to the words in the database. Go gets matched to a database of verbs and the Kitchen gets matched to the database of locations. Then words are published to ibm_watson for speech output and destination coordinate is published to the navigation stack.

COMMAND - Go to the airport

Here though the verb” go” gets matched, the place is not matched to any of the locations in the database. So ibm_watson gives “No destination is given and the task cannot be done" and the task is stopped.

VI. MEASUREMENT AND SETUP

In the arena, the robot is parked in the parking location. Whenever a person needs assistance from the robot, the customer should activate the robot using the keyword ***“hello”***. Now the robot gets activated and replies by saying

“Hello! Arena has different sections like room1, room2, room3, and room4.”

room1 - Kitchen
room2 - Bedroom
room3 - Dining area
room4 - Hall

According to the requirement, the directions to the destination are given as the route to every location is already stored in the database of the robot. For the implementation, the robot is tested in our research lab where the lab is mapped and is divided into 4 rooms as shown in Fig. 9. The robot was tested in each of these sections and the performance was noted down.

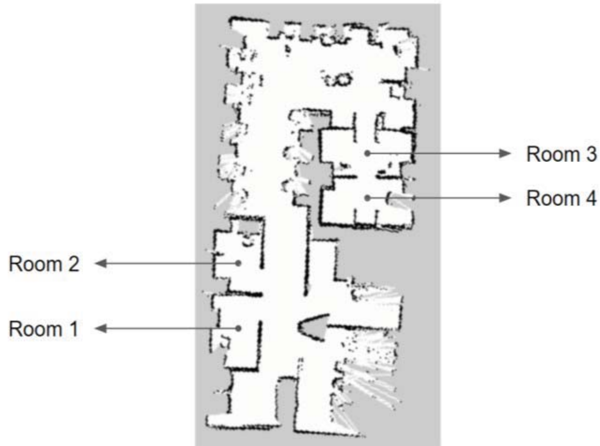

Fig. 9. Map of the lab

VII. RESULTS AND DISCUSSION

The robot is expected to perform the task accordingly. If it hears the word “GO” then navigation mode is activated. If it hears the word “CAN” it will answer. If it hears “GO+ROOM_ _” it will go to that specific location which is mentioned by the user.

1. *Experiment 1*

Initial position: Kitchen
Final position: Dining area
Input speech command: “*skedaddle to Dining area*”
go and Dining area is the keywords recognized by the system and the processing takes place and the robot is navigated to the Dining area.

2. *Experiment 2*

Initial position: Bedroom
Final position: Dining area
Input speech command: “*please go to Dining area*”
go and Dining area is the keywords recognized by the system and the processing takes place and the robot is navigated to the Dining area.

3. *Experiment 3 (with obstacles)*

Initial position: Dining area
Final position: Kitchen
Input speech command: “*skedaddle to Kitchen”*
skedaddle and Kitchen are the keywords recognized by the body and the processing takes place and the robot is navigated to the Dining area.

4. *Experiment 4 (with obstacles)*

Initial position: Kitchen
Final position: Hall
Input speech command: “*skedaddle to Hall*”
go and Hall is the keywords recognized by the body and the processing takes place and the robot is navigated to the Hall.

5. *Experiment 5*

Initial position: Hall
Final position: Bedroom
Input speech command: "*Can you please go to Bedroom*"
Can and Bedroom are the keywords recognized by the body and the processing takes place and the robot is navigated to the Bedroom.

Trials	Command recognition	speech output	Time taken to initiate Navigation	Time taken to reach the destination	Error in distance	Error in Orientation
exp1	No	No	--	--	--	--
exp2	Yes	Yes	6 sec	108 sec	8 cm	17°
exp3	Yes	Yes	5 sec	--	--	--
exp4	Yes	Yes	4 sec	102 sec	15 cm	25°
exp5	Yes	Yes	5 sec	115 sec	20 cm	14°

Table 2. Observations of the repeated experiments

Table. 2 shows the observations noted while testing the whole system in the practical environment. In English, many words have multiple pronunciations. If a user pronounces any word in a different ascent other than the trained ascent, the robot mightn't recognize the word. As a result, when ascent is changed the robot failed to detect the speech input in exp1. In exp3 though the input speech is recognized, because of the many obstacles in the arena the robot could not find any recovery behavior and stops path planning by giving speech output "No recovery behaviors found." In the remaining experiments, the robot detects the speech and navigate to a destination. The error rate is 10-12 cm in driving the robot to the desired goal. This occurs when the robot is tested in a sophisticated environment with many obstacles. The slight delay is observed because of processing time for decision making and mapping algorithms.

VIII. CONCLUSION

The integration of speech recognition and autonomous navigation platform will be an ideal system to serve the disabled and physically handicapped people. For speech recognition, Hidden Markov's Model (HMM) supported by the Viterbi Algorithm was used for the desired output and for mutual communication *ibm_watson* assistant was used. Apart from the Speech, G mapping and SLAM are used for navigating the robot from one place to another place. The implementation of the whole system in dynamic environments procured efficient and accurate results.

IX. FUTURE WORK

The accuracy of the robot navigation to the desired location is tested in different environments and conditions where it was observed that accuracy varies with the change in the IMU model. By using some sophisticated algorithms and machine learning techniques, the robot can be trained and tested simultaneously. In the next phase, we are going to make the 3D map for navigation purposes by incorporating the Visual data with the Odometry data to achieve high accuracy navigation. Filters can be added to the microphone for efficient output.

ACKNOWLEDGMENT

It is gladdening to thank Humanitarian Technology Labs and the Department of Electronics and Communication Engineering, Amrita Vishwa Vidyapeetham, Kollam who provided us with a highly encouraging work environment and lab facilities, which was the backbone for completion of this project.

REFERENCES

[1] E. Voisan, B. Paulis, R. Precup and F. Dragan, "ROS-based robot navigation and human interaction in indoor environment," *2015 IEEE 10th Jubilee International Symposium on Applied Computational Intelligence and Informatics*, Timisoara, 2015, pp. 31-36.

[2] https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=2ahUKEwjXkbfAna_kAhWRYH0KHWDzAYcQFjABegQIChAE&url=https%3A%2F%2Fwww.childrenshospital.org%2F-%2Fmedia%2FCenters-and-Services%2FPrograms%2FF_N%2FHand-and-Upper-Orthopedic-Extremity-Program%2FPatient-Fact-Sheets%2FSymbrachydactyly.ashx%3Fla%3Den%26hash%3D4A7B05F172D1294EA59C9F8794183670BA3B5866&usg=AOvVaw11ZlXU9eFQXxM240soovyb

[3] https://www.who.int/news-room/fact-sheets/detail/ageing-and-health

[4] M. Tharaniya soundhari and S. Brilly Sangeetha, "Intelligent interface based speech recognition for home automation using android application," *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Coimbatore, 2015, pp. 1-11.

[5] R. Li, J. Liu, L. Zhang and Y. Hang, "LIDAR/MEMS IMU integrated navigation (SLAM) method for a small UAV in indoor environments," *2014 DGON Inertial Sensors and Systems (ISS)*, Karlsruhe, 2014, pp. 1-15.

[6] R. Opromolla, G. Fasano, G. Rufino, M. Grassi and A. Savvaris, "LIDAR-inertial integration for UAV localization and mapping in complex environments," *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, Arlington, VA, 2016, pp. 649-656.

[7] R. K. Megalingam, S. Vishnu, S. Sekhar, V. Sasikumar, S. Sreekumar and T. R. Nair, "Design and Implementation of an Android Application for Smart Shopping," *2019 International Conference on Communication and Signal Processing (ICCSP)*, Chennai, India, 2019, pp. 0470-0474.

[8] R. Kannan Megalingam, C. Ravi Teja, S. Sreekanth, and A. Raj, "ROS based Autonomous Indoor Navigation Simulation Using SLAM Algorithm," International Journal of Pure and Applied Mathematics, vol. 118, no. 7, pp. 199–205, 2018

[9] R. K. Megalingam, R. N. Nair and S. M. Prakhya, "Automated voice based home navigation system for the elderly and the physically challenged," *2011 2nd International Conference on Wireless*

Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), Chennai, 2011, pp. 1-5.

[10] S. Mishra, R. Kankal, S. Lunawat, P. Ram and V. Gupta, "Interactive automation interface using android app and speech recognition over wireless LAN and Internet," *International Conference & Workshop on Electronics & Telecommunication Engineering (ICWET 2016)*, Mumbai, 2016, pp. 93-99.

[11] S. Gatesichapakorn, J. Takamatsu and M. Ruchanurucks, "ROS based Autonomous Mobile Robot Navigation using 2D LiDAR and RGB-D Camera," *2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP)*, Bangkok, Thailand, 2019, pp. 151-154.

[12] Y. Cheng and G. Y. Wang, "Mobile robot navigation based on lidar," *2018 Chinese Control And Decision Conference (CCDC)*, Shenyang, 2018, pp. 1243-1246.

[13] Y. Zhang, Z. Lu, C. Wang, C. Liu and Y. Wang, "Voice control dual arm robot based on ROS system," *2018 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*, Shenyang, 2018, pp. 232-237.

[14] Z. Li, Y. Xiong and L. Zhou, "ROS-Based Indoor Autonomous Exploration and Navigation Wheelchair," *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, Hangzhou, 2017, pp. 132-135.