Robot operating system based autonomous navigation platform with human robot interaction

Rajesh Kannan Megalingam[1], Vignesh S. Naick[1], Manaswini Motheram [1], Jahnavi Yannam[1], Nikhil Chowdary Gutlapalli[1], Vinu Sivananthamr[2]
[1]Department of Electronics and Communication Engineering, Amrita Vishwa Vidyapeetham, Amritapuri, India
[2]Engineering Product Development, Singapore University of Technology and Design, Singapore

Article Info

Article history:

Received Aug 12, 2021
Revised Mar 31, 2022
Accepted Jun 01, 2022

Keywords:

Human robot interaction
Object recognition
ROS
SLAM
Speech recognition

ABSTRACT

In emerging technologies, indoor service robots are playing a vital role for people who are physically challenged and visually impaired. The service robots are efficient and beneficial for people to overcome the challenges faced during their regular chores. This paper proposes the implementation of autonomous navigation platforms with human-robot interaction which can be used in service robots to avoid the difficulties faced in daily activities. We used the robot operating system (ROS) framework for the implementation of algorithms used in auto navigation, speech processing and recognition, and object detection and recognition. A suitable robot model was designed and tested in the Gazebo environment to evaluate the algorithms. The confusion matrix that was created from 125 different cases points to the decent correctness of the model.

Corresponding Author:

Rajesh Kannan Megalingam
Department of Electronics and Communication Engineering, Amrita Vishwa Vidyapeetham
Amritapuri, Kerala 690525, India
Email: rajeshm@am.amrita.edu

1. INTRODUCTION

Robotics has been an emerging field of study and application in recent times. Almost all the industry in the world and several sectors use robotics and automation for handling several tasks. A service robot is one such system that can be of great help in disaster management, helping people in need like elders and handicapped. The applications vary from using them in industrial spaces, mines and hazardous places for human intervention, disaster management and even in homes as caring robots. One of the factors that decide the success of these service robots lies in the human robot interaction (HRI). The way we control it and the modes in which the robots accept the commands is also a concern. Speech recognition and processing it to provide meaningful speech output that has enhanced the HRI. This paper proposes to develop a autonomous voice-controlled service robot that would serve for the old and physically challenged or disabled peoples to replace human assistance thereby minimizing the labor and human intervention. This integrated system performs tasks such as mapping, human-robot interaction and cooperation, object detection and navigation with obstacle avoidance system in the dynamic environments. Here, the development of an autonomous service robot integrated with a voice can autonomously navigate in the indoor environment without any human intervention. This paper is an extended version of the HRI on the navigation platform using the robot operating system [1].

There's been extensive research going on human-robot interaction. To achieve this, there are many fields that need to be focused on, like simultaneous localization and mapping (SLAM) for mapping the environment and localization, image detection and recognition, speech recognition and conversation. For mapping the unknown environment, Li *et al.* [2] proposes a method by registering the 3-D light detection and ranging (LiDAR) point and frequently updating the Bayesian probability. Davison *et al.* [3] came up with a method of using single monocular cameras which can be used in many applications like mapping the environment, localizing in the map and in augmented reality. In that way, Chan *et al.* [4] explains a method to fuse the laser scan from LiDAR sensors and data from monocular cameras by using the trajectory matching method for robust localization. For localizing the robot in the physical harsh condition, Ullah *et al.* [5] have proposed a novelty by modifying the Kalman filter with the help of particle filter (PF) and unscented Kalman filter (UKF) localization algorithms.

Obstacle detection is one of the main requirements to be considered for advanced driver assistance systems (ADAS). Catapang and Ramos [6], detection of obstacles was made using the LiDAR-Lite v1. By keeping the field of view as 360 degrees, the detection was made by the clustering method. To understand the robot behavior, the [7] has shown the comparison of the robot planned path and its travelled path when one of the destinations was provided. In the [8], the mobile robot was navigated using global positioning system (GPS) after mapping the environment using SLAM. Wavefront algorithm was used for path planning and very high frequency (VHF) algorithm for obstacles avoidance. The detailed steps for mapping the environment, localizing the robot in the map and the navigation in the Gazebo environment was shown by Chikurtev [9]. To evaluate the indoor navigation system, Testa *et al.* [10], Tao and Ganz [11] proposed a cost-effective simulation framework for all types of users. In the application-wise, Megalingam *et al.* [12] has shown how an intelligent wheelchair was developed and simulated in the virtual environment.

Speech recognition is an integral part and one of the main requirements for human-robot interaction. Hinton *et al.* [13] proposed the use of deep neural networks (DNN) which has many hidden layers and well-trained models as an alternative to the use of Gaussian mixture modeling (GMM). It is observed that the joint training framework for speech enhancement and recognition have made the end to end automatic speech recognition (ASR) more robust. Fan *et al.* [14] proposes a method in which gated recurrent is being fusion with a joint training framework for enhancing the performance of ASR. In natural environment circumstances speech is interrupted by more than one talker. For such cases, Yousefi and Hansen [15] explains the block-based convolutional neural network (CNN) architecture to address overlapping speech in audio streams with short frames of 25 ms. A method of implementing speech recognition in the robot operating system (ROS) is proposed in Megalingam *et al.* [16] using hidden Markov model (HMM). Daniel *et al.* [17] has shown a development framework for the low-code chatbot named Xatkit, which is open source. In the field of computer vision tasks, DNNs was the eminent performer in the detection and recognition tasks. However, these DNNs require powerful devices which have the fast computational power and sufficient memory. As a contradiction to this, a new algorithm named Tinier you only look once (YOLO) which shrinks the size of the model while improving its detection accuracy and its performance in real-time was proposed in [18].

The research papers [19]-[21], discuss about the object detection and recognition for various robotic applications using various algorithms including YOLO, YOLO_v3 and audio feedback and audio output. Human-robot interaction (HSR) has recently become an interesting field of study where robots interact along with humans in their day to day activities making life easier for humans especially those physically challenged. A sensor network capable of observing human to human interactions in the real world from which it takes the data to model the behaviour of the robot is proposed in [22]. HRI has also found its space in a multi-robot system where multiple robots communicate with each other to solve problems. Huang *et al.* [23] showed a new human decision-making model for the control of multi-robot systems through HRI. To obtain appropriate human aid, meaningful questions should be generated by the robot regarding the task execution and procedures and then modifying its plans and behavior according to the speech input. Kim and Yoon [24] proposes a script-based task planning. The planning is executed from a set of scripts which helps in easy to manipulate robot behavior in a practical environment. A robot using HRI and detecting the faces using OpenCV algorithms using ROS is proposed in [25]. In the interaction tasks of navigation and speech, İskender *et al.* [26] proposes a method to control the vehicle autonomously using speech commands. Here the VoiceBot application and google voice were used for recognizing speech commands. In this paper, we study robots working and effectiveness in a home-like environment executing the multiple tasks on a virtual environment scenario created in Gazebo. A confusion matrix is obtained based on the several test cases inferring the performance of the robot in observation and pick and place scenarios.

2. METHOD

The software architecture of the system is shown in Figure 1. There are three major modules including the sensor feedback, the ROS module and the control command module for the ROS Gazebo. Autonomous systems rely mostly on their sensor feedback for executing the closed-loop control. The proposed robot is equipped with a range of sensors for navigation like LiDAR for mapping and obstacle avoidance during the auto navigation, odometry sensor for the accelerometer and gyroscope data which is used while the robot is in motion, encoder data for localization and position feedback. We are using a microphone for providing the speech input to the robot as voice commands. Based on the voice command the robot can execute the tasks. A camera is provided for object detection and recognition. The second module is the ROS module where the algorithms function using the sensor feedback. There are three main working algorithms for navigation, speech and object detection. The odometry showing its position and orientation data combined with the localization and path planning algorithm form the decision-making block for the navigation. We use voice commands for the HRI. This block consists of speech recognition and its processing part. From this block, inputs are fed to the navigation and object detection blocks. The third module is the ROS Gazebo module. This module has several plugins which provide greater functionality to the universal robot description format (URDF) by integrating the ROS messages and service calls for sensor data output and motor inputs for simulating in the virtual environment.

2.1. Human-robot interface

Autonomous robots are made more efficient through the introduction of human-robot interaction. Assistive robots use HRI, mostly for providing the best care and service. Speech is considered as a primary and important input to the robot which helps the robot to have clarity on what the user wants. This task of making the robot understand what needs to be done happens in speech module which is made up of three sub modules.

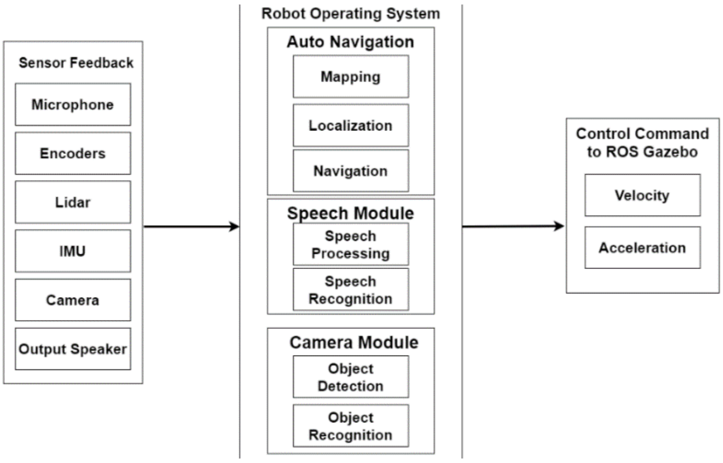

Figure 1. System architecture

2.1.1. Speech recognization, processing and response

To achieve accurate speech recognition, continuous training of huge speech data is required. Google speech recognition uses separate components like the hidden Markov model, language model and deep neural network-based acoustic models for speech recognition. As deploying and storing this information on mobile devices is an impossible process, the training process is done in a cloud platform. More the training data would be the accuracy of speech recognition. The target domain utterances are matched with training speech corpus based on Viterbi and hidden Markov models. Speech processing is the processing of the natural language for the robots to understand. This enables us to interact with robots where they are able to understand what we told and are able to create responses based on it. We are using the artificial intelligence markup language (AIML) for speech processing. AIML is a XML based markup language. The recognized speech input is fed to the AIML module where it compares the recognized data with the corpus file stored in the cloud and generates responses based on that. Text to speech (TTS) conversion is done in 2 steps: front-end and back-end. In the front-end, all the abbreviations and numbers are converted into corresponding words called tokenization. Then each word is assigned with phonetic transcription and the words are divided into phrases, clauses and sentences which is 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. Whereas, back end converts the output of the front end into sound.

2.2. Autonomous navigation

Many sensors like LiDAR, inertial measurement unit (IMU), and encoders are used to monitor and produce commands for navigation. LiDAR is a light detection and ranging sensor used for mapping as well as for obstacle avoidance purposes. IMU is an inertial measurement unit used to get orientation, angular velocity and angular acceleration. Encoders are used for getting the motor data to know how much distance the robot travelled. The sequence of steps in navigation include mapping, localization, global path planning, and local path planning.

Mapping is the first step in auto navigation where the robot is made to move all around the space to create a 2D map of the environment. The LiDAR emits the laser rays and receives the reflected rays. Based on the laser signal strength the map of the whole environment is created with the obstacles. We use the Gmapping technique for mapping the environment. The next step that follows mapping is the localization which helps the robot to know and locate its position in the environment. ROS uses the adaptive Monte Carlo localization method to localize itself. This uses a particle filter to track the robots pose against the known map. Localization is followed by global path planning. Here the robot based on its saved map computes the shortest distance to the destination for the robot to travel. We use a star algorithm as the efficiency is higher compared to Dijkstra's. Based on the global path planning a local planner provides a controller for the robot for executing the navigation. It also helps the robot in path planning several local path planning algorithms are developed like dynamic window approach (DWA), EBand, timed elastic band (TEB), and carrot planners. For this study, we are using the carrot planner.

YOLO is a real-time object detection system. Multiple bounding boxes and class probabilities of those bounding boxes are predicted using a single convolution network. During training and testing of the pictures, YOLO sees the whole image so that it can encode contextual information about the classes and their aspects. The most important advantage of YOLO is the speed with which it can process forty-five frames in a second. The Gazebo is a part of the "player project" and allows simulation of robotic and sensor applications in three-dimensional indoor and outdoor environments. The architecture of the Gazebo is of client/server architecture. It uses the publishers and subscribers for their inter-process communication.

2.3. Design and implementation

For this study, we have created our robot and designed a ROS Gazebo environment for testing the software. For creating our own robot and importing it to the Gazebo environment, we first created a computer aided design (CAD) model in the SolidWorks software. The SolidWorks model of our robot is shown in Figure 2. The robot base also holds a small cylindrical structure to support the LiDAR sensor to help in mapping and obstacle avoidance. The facility/environment with four furnished rooms including living room, kitchen, bedroom and dining room, is created in Gazebo as shown in Figure 3. This environment is used for simulation tests. The living room has two couches, a small table, a TV stand and a shelf. At the other end of the living room is the kitchen. Apart from the living room, the environment is having a dining room with a 4-seater table and chairs. A person can also be located in the bedroom for recognizing as a dynamic obstacle while testing.

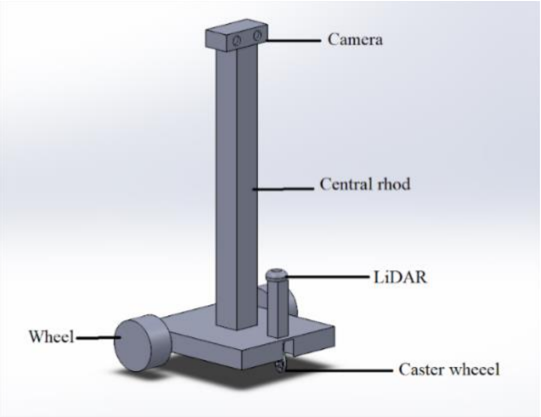

Figure 2. Solidworks design for the proposed robot

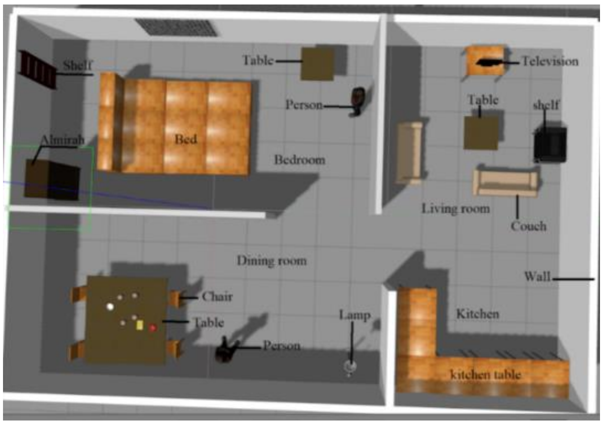

Figure 3. Gazebo environment

The pseudo code of our software implementation is shown in Algorithm 1. Initially, the map which is created by Gmapping is passed to the robot to know the locations available and to localize itself in that map. After the robot localizes itself in the provided map, the robot tries to listen for the input commands using speech recognition by the microphone which was attached to it. From the input command, the robot divides the sentence into words and checks whether the location and item were provided for searching. For example, if the input command for the robot is "go to the kitchen and find a bottle", the system divides the sentence into each word like 'go', 'to', 'the', 'kitchen', 'and', 'find' and 'bottle'. In these individual words, the system will search whether the location and item for the detection were present.

2.4. Pseudo code

In the example, the ‘kitchen’ is mentioned as a location and the ‘bottle’ is mentioned as the object to be searched in the kitchen. With the help of ROS navigation stack, the robot navigated to the provided location, which is the kitchen in this case. Once the robot reaches the location, it starts searching for the object given. If the object is recognized using YOLO, the robot comes to its starting position and informs the user that it has found the object in the kitchen, via the speaker. If the object is not found, the robot informs that it cannot find the object. If the command is not in the list of known commands by the robot, it informs that it cannot recognize the command.

Algorithm 1. The pseudo code of software implementation

```
IMPORT locations
IMPORT items_trained
FUNCTION goToLocation(location):
        location_coordinates = get the destination coordinates from RVIZ,
        path = Using Path planning algorithms(present_cooridinate, location coorinate)
        Provide the path to robot for navigation
ENDFUNCTION goToLocation
FUNCTION speechOutput(text):
        output_speaker = OUTPUT: text
ENDFUNCTION speechOutput
FUNCTION searchForItem(item_trained):
        findItem = YoloAlgotithm(item_trained)
ENDFUNCTION searchForItem
speech_processed = INPUT : “How Can I Help You?”
location = find any locations in speech_processed
item = find any items_trained in speech_processed
start_location = present_location
IF any locations was found in speech_processed:
      IF any items_trained were found in speech_processed
            goToLocation(locations)
            IF success:
                  searchForItem(items_trained)
                  IF object found:
                        goToLocation(start_location)
                        speechOutput(“I found the object in the location”)
                  ELSE:
                        goToLocation(start_location)
                        speechOutput(“I didn’t found the object in the location”)
            ELSE:
                  goToLocation(start_location)
                  speechOutput(“I couldn’t go to that location”)
      ELSE:
            speechOutput(“I didn’t recognize any object to find”)
ELSE:
      speechOutput(“I didn’t recognized any location.”)
```

3. RESULTS AND DISCUSSION

To evaluate the robot model performance, it was tested with various input commands. Each command was tested for 6 trials each. Five objects were palced in the virtual environment to be identified by the robot including bottle, plate, bowl, coke and ball. Table 1 to Table 4 contain the list of input commands which were used for testing. All the objects were placed in each of the rooms in the Gazebo environment as shown in the Figure 4. The robot was given speech commands to carry out the specific tasks in the command and provide information about the object kept in the rooms. The robot executed the tasks according to the speech input. The system was evaluated on four factors: factor a) the robot recognizes the voice command correctly and completes the task successfully; factor b) the robot recognized the voice command wrongly and completes some other task other than what was requested; factor c) the robot recognizes the voice command correctly, but the task is not completed successfully; and factor d) the robot doesn’t recognize the voice command and the task is not completed. The recorded observations are shown from Table 1 to Table 4.

The number entries under each of the factor a to d indicates how many times the robot obeyed the command successfully out of the 6 trials conducted. The confusion matrix describes the performance of a model based on a set of tests data. Table 5 shows the confusion matrix for our study on HRI. There are two possible predicted classes "yes" or "no". The classifier is made from a total of 120 predictions (n). The matrix mainly accommodates four terms that can be read from the matrix which are: a) true positive (TP): recognized the voice command and executed the task; b) true negative (TN): voice command not recognized and task not executed; c) false-positive (FP): voice command recognized incorrectly, and a wrong task executed; and d) false-negative (FN): recognized voice command but task not executed. On analyzing the parameters from the confusion matrix in Table 5, the accuracy of the model is found with an accuracy rate of 0.925. The rate at which the model not executing the task on recognizing the speech output is found to be 0 and the misclassification rate is only 0.075. The effectiveness of the speech module is proved with the precision and the true positive rate both at 1. The prevalence parameter shows the effectiveness of the robot in performing the tasks is good at 0.86.

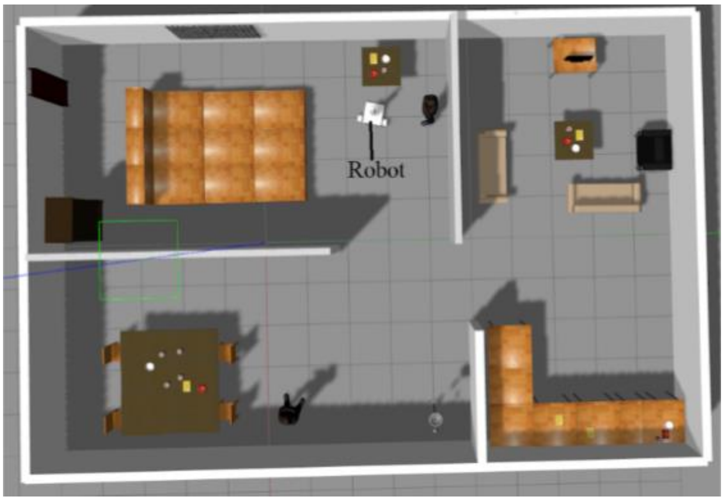

Figure 4. Gazebo environment while testing with robots

Table 1. Observations obtained from kitchen

Input command	Results from the input command			
	Factor (a)	Factor (b)	Factor (c)	Factor (d)
Go to kitchen and find bottle	6	0	0	0
Go to kitchen and find plate	6	0	0	0
Go to kitchen and find bowl	3	0	3	0
Go to kitchen and find coke	6	0	0	0
Go to kitchen and find ball	3	0	0	3

Table 2. Observations obtained from bedroom

Input command	Results from the input command			
	Factor (a)	Factor (b)	Factor (c)	Factor (d)
Go to bedroom and find bottle	6	0	0	0
Go to bedroom and find plate	6	0	0	0
Go to bedroom and find bowl	6	0	0	0
Go to bedroom and find coke	6	0	0	0
Go to bedroom and find ball	4	0	0	2

Table 3. Observations obtained from living room

Input command	Results from the input command			
	Factor (a)	Factor (b)	Factor (c)	Factor (d)
Go to living room and find bottle	5	0	0	1
Go to living room and find plate	6	0	0	0
Go to living room and find bowl	6	0	0	0
Go to living room and find coke	5	0	1	0
Go to living room and find ball	6	0	0	0

Table 4. Observations obtained from dining room

Input command	Results from the input command			
	Factor (a)	Factor (b)	Factor (c)	Factor (d)
Go to the dining room and find bottle	6	0	0	0
Go to dining room and find plate	6	0	0	0
Go to dining room and find bowl	6	0	0	0
Go to dining room and find coke	5	0	0	1
Go to dining room and find the ball	4	0	2	0

Table 5. Confusion matrix of our results

n = 120	Voice command recognition yes	Voice command recognition no
Task execution Yes	TP = 104	FP = 0
Task execution No	FN = 9	TN = 7

4. CONCLUSION

As a step towards exploring the HRI, we have studied and analyzed the performance of the speech and image integrated over to a navigation platform. In this work we presented the system architecture designed for this particular research work followed by the design and implementation. A command list of 20 commands and five objects was created for the simulation tests. Each of the command was tested for six trials and the results were noted. Finally, a confusion matrix was created with noted results to find the accuracy of the system. The accuracy rate of the proposed system was found to be 0.925 which reflects a very good accuracy of the proposed system.

ACKNOWLEDGEMENTS

We are gratified to express our gratitude towards the Electronics and Communication Engineering Department and HuT Labs of Amritapuri campus of Amrita Vishwa Vidyapeetham University for their ceaseless success, without which this work would not have progressed.

REFERENCES

[1] R. K. Megalingam, M. Manaswini, J. Yannam, V. S. Naick, and G. N. Chowdary, “Human Robot Interaction on Navigation platform using Robot Operating System,” in *2020 Fourth International Conference on Inventive Systems and Control (ICISC)*, 2020, pp. 898-905, doi: 10.1109/ICISC47916.2020.9171065.

[2] B. Li, L. Yang, J. Xiao, R. Valde, M. Wrenn, and J. Leflar, “Collaborative Mapping and Autonomous Parking for Multi-Story Parking Garage,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1629-1639, 2018, doi: 10.1109/TITS.2018.2791430.

[3] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, “MonoSLAM: Real-Time Single Camera SLAM,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052-1067, 2007, doi: 10.1109/TPAMI.2007.1049.

[4] S. -H. Chan, P. -T. Wu, and L. -C. Fu, “Robust 2D Indoor Localization Through Laser SLAM and Visual SLAM Fusion,” in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2018, pp. 1263-1268, doi: 10.1109/SMC.2018.00221.

[5] I. Ullah, Y. Shen, X. Su, C. Esposito, and C. Choi, “A Localization Based on Unscented Kalman Filter and Particle Filter Localization Algorithms,” *IEEE Access*, vol. 8, pp. 2233-2246, 2020, doi: 10.1109/ACCESS.2019.2961740.

[6] A. N. Catapang and M. Ramos, “Obstacle detection using a 2D LIDAR system for an Autonomous Vehicle,” in *2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, 2016, pp. 441-445, doi: 10.1109/ICCSCE.2016.7893614.

[7] R. K. Megalingam, A. Rajendraprasad, and S. K. Manoharan, “Comparison of Planned Path and Travelled Path Using ROS Navigation Stack,” in *2020 International Conference for Emerging Technology (INCET)*, 2020, pp. 1-6, doi: 10.1109/INCET49848.2020.9154132.

[8] D. Reyes, G. Millan, R. O. -Corparan, and G. Lefranc, “Mobile Robot Navigation Assisted by GPS,” *IEEE Latin America Transactions*, vol. 13, no. 6, pp. 1915-1920, 2015, doi: 10.1109/TLA.2015.7164217.

[9] D. Chikurtev, “Mobile Robot Simulation and Navigation in ROS and Gazebo,” in *2020 International Conference Automatics and Informatics (ICAI)*, 2020, pp. 1-6, doi: 10.1109/ICAI50593.2020.9311330.

[10] A. Testa, M. Cinque, A. Coronato, G. D. Pietro, and J. C. Augusto, Heuristic strategies for assessing wireless sensor network resiliency: an event-based formal approach,” *Journal of Heuristics*, vol. 21, pp. 145–175, 2015, doi: 10.1007/s10732-014-9258-x.

[11] Y. Tao and A. Ganz, “Simulation Framework for Evaluation of Indoor Navigation Systems,” *IEEE Access*, vol. 8, pp. 20028-20042, 2020, doi: 10.1109/ACCESS.2020.2968435.

[12] R. K. Megalingam, G. B. Vishnu, and M. Pillai, “Development of intelligent wheelchair simulator for indoor navigation simulation and analysis,” in *2015 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, 2015, pp. 74-77, doi: 10.1109/WIECON-ECE.2015.7444002.

[13] G. Hinton *et al.*, “Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82-97, 2012, doi: 10.1109/MSP.2012.2205597.

[14] C. Fan, J. Yi, J. Tao, Z. Tian, B. Liu, and Z. Wen, “Gated Recurrent Fusion With Joint Training Framework for Robust End-to-End Speech Recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 198-209, 2021, doi: 10.1109/TASLP.2020.3039600.

[15] M. Yousefi and J. H. L. Hansen, “Block-Based High Performance CNN Architectures for Frame-Level Overlapping Speech Detection,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 28-40, 2021, doi: 10.1109/TASLP.2020.3036237.

[16] R. K. Megalingam, R. S. Reddy, Y. Jahnavi, and M. Motheram, “ROS Based Control of Robot Using Voice Recognition,” *2019 Third International Conference on Inventive Systems and Control (ICISC)*, 2019, pp. 501-507, doi: 10.1109/ICISC44355.2019.9036443.

[17] G. Daniel, J. Cabot, L. Deruelle, and M. Derras, “Xatkit: A Multimodal Low-Code Chatbot Development Framework,” *IEEE Access*, vol. 8, pp. 15332-15346, 2020, doi: 10.1109/ACCESS.2020.2966919.

[18] W. Fang, L. Wang, and P. Ren, “Tinier-YOLO: A Real-Time Object Detection Method for Constrained Environments,” *IEEE Access*, vol. 8, pp. 1935-1944, 2020, doi: 10.1109/ACCESS.2019.2961959.

[19] E. M.-Martin and A. P. D. Pobil, “Object Detection and Recognition for Assistive Robots: Experimentation and Implementation,” *IEEE Robotics & Automation Magazine*, vol. 24, no. 3, pp. 123-138, 2017, doi: 10.1109/MRA.2016.2615329.

[20] M. Mahendru and S. K. Dubey, “Real Time Object Detection with Audio Feedback using Yolo vs. Yolo_v3,” in *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 2021, pp. 734-740, doi: 10.1109/Confluence51648.2021.9377064.

[21] V. Y. Mariano *et al.*, “Performance evaluation of object detection algorithms,” in *2002 International Conference on Pattern Recognition*, 2002, pp. 965-969, vol. 3, doi: 10.1109/ICPR.2002.1048198.

[22] P. Liu, D. F. Glas, T. Kanda, and H. Ishiguro, “Data-Driven HRI: Learning Social Behaviors by Example From Human–Human Interaction,” *IEEE Transactions on Robotics*, vol. 32, no. 4, pp. 988-1008, 2016, doi: 10.1109/TRO.2016.2588880.

[23] J. Huang, W. Wu, Z. Zhang, and Y. Chen, “A Human Decision-Making Behavior Model for Human-Robot Interaction in Multi-Robot Systems,” *IEEE Access*, vol. 8, pp. 197853-197862, 2020, doi: 10.1109/ACCESS.2020.3035348.

[24] Y. Kim and W. C. Yoon, “Generating Task-Oriented Interactions of Service Robots,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 8, pp. 981-994, 2014, doi: 10.1109/TSMC.2014.2298214.

[25] E. -I. Voisan, B. Paulis, R. E. Precup, and F. Dragan, “ROS-based robot navigation and human interaction in indoor environment,” in *2015 IEEE 10th Jubilee International Symposium on Applied Computational Intelligence and Informatics*, 2015, pp. 31-36, doi: 10.1109/SACI.2015.7208244.

[26] A. İskender, H. Üçgün, U. Yüzgeç, and M. Kesler, “Voice command controlled mobile vehicle application,” in *2017 International Conference on Computer Science and Engineering (UBMK)*, 2017, pp. 929-933, doi: 10.1109/UBMK.2017.8093565.

BIOGRAPHIES OF AUTHORS

Rajesh Kannan Megalingam received his bachelor’s degree in engineering from College of Engineering, Guindy, Chennai and masters and PhD from Amrita Vishwa Vidyapeetham, Kollam, India. He is the Director of HuT Labs and Asst. Professor of ECE Dept., Amrita Vishwa Vidyapeetham University India. His research areas include Low Power VLSI, Design for Manufacturing and Embedded Systems. He has worked as VLSI Design and Verification Engineer at various companies like STMicro Electronics, Insilicon Incorporation. in Bay Area, California for six years. He has published more than 140 research papers at various international conferences, journals and book chapters. He has won several awards including the IEEE Undergraduate Teaching Award 2020, the Outstanding Branch Counselor and Advisor award from IEEE, NJ, USA, Outstanding Branch Counselor Award from IEEE Kerala Section, Award of Excellence from Amrita University. He is currently leading several research projects in robotics and automation. He can be contacted at email: rajeshm@am.amrita.edu.

Vignesh S. Naick received his bachelor’s degree in engineering from Amrita School of Engineering, Amrita Vishwa Vidyapeetham University, Amritapuri Campus, Kollam, Kerala. He is currently working in Infosys, Mysore Campus, India. He can be contacted at email: vsnayak160@gmail.com.

Manaswini Motheram received his bachelor’s degree in engineering from Amrita School of Engineering, Amrita Vishwa Vidyapeetham University, Amritapuri Campus, Kolla, Kerala. She is currently persuing masters in University of Twente, Netherlands. She can be contacted at email: motherammanaswini30@gmail.com.

Jahnavi Yannam received his bachelor's degree in engineering from Amrita School of Engineering, Amrita Vishwa Vidyapeetham University, Amritapuri Campus, Kolla, Kerala. She is currently working in Hyundai Mobis, India. She can be contacted at email: jahnaviyannam@gmail.com.

Nikhil Chowdary Gutlapalli received his bachelor's degree in engineering from Amrita School of Engineering, Amrita Vishwa Vidyapeetham University, Amritapuri Campus, Kollam, Kerala. He is currently working in Multicoreware, India. He can be contacted at email: g.nikhilchowdaryatp@gmail.com.

Vinu Sivanantham completed his bachelors in 2017 at Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Amritapuri Campus, Kerala, India. He completed his dual Master's degree in Masters in Nano Electronic Engineering and Design from CGU, Taiwan and SUTD, Singapore in 2020. Currently is a researcher at SUTD, Singapore. He area of focus is on reconfigurable robots. He can be contacted at email: vnu.619@gmail.com.