Design and Implementation of Custom Built Quarantine Service Mobile Robot Using Deep Learning and ROS

Sanjuna Mariam Mathews
College Of Engineering Trivandrum
Trivandrum , India
sanjunamariam25@gmail.com

Sreeja. S
College Of Engineering Trivandrum
Trivandrum , India
sreejas@cet.ac.in

***Abstract*— An autonomously navigated custom-built quarantine service robot is a vehicle with sensing ability of its environment and is moving safely in static and dynamic environments. The indoor environment under consideration for robot installation is isolation wards. The robot also serves applications such as patient inspection, greeting the patient using name, logging the name to document for attendance purpose, food delivery, medicine delivery, etc. Also, there are many integrated features like automatic room sanitation, non-contact hand sanitizer, touch less drinking water dispenser, dedicated telemedicine chat application etc. A ROS hybrid algorithm, using the combination of DWA and TEB Path Planners is used for the dynamic path planning of the service robot. The proposed hybrid algorithm used for simulation and hardware implementation has less execution time as that of DWA and TEB path planners. With obstacles, DWA planner and TEB planner takes 55 and 47 seconds respectively for navigating to a fixed goal. The proposed system has an efficient path planning with an execution time of 43 seconds for navigating to the fixed goal. The tuned hyper parameters provides an optimized path plan to the goal. The objective is to deliver services to patients in quarantine by avoiding maximum human contact.**

Keywords*— *ROS, Mobile Robot , Autonomous Navigation , SLAM , DWA , TEB , AMCL , Quarantine Service Robot, Face Recognition

INTRODUCTION

A lot of people in India are currently under quarantine since the world is fighting against many highly infectious communicable diseases like COVID -19. The existing hospital robots like Hospital Robot Teams[1], Helpmate[2], etc. mainly concentrated on the path planning and task distribution algorithms. The proposed quarantine service robot can autonomously navigate using ROS - Hybrid Algorithm with execution time far better than existing algorithms. The ROS-Navigation algorithm parameters are tuned to provide the best performance in path planning and navigation. The robot is cost-effective and can serve multiple applications. In isolation wards, the daily attendance is taken manually by a human. But in the presence of highly communicable diseases, it is necessary to minimize human contact. This warrants a multipurpose robotic solution.

Several works on the implementation of hospital robots for various applications are found in the literature. Team Co-operative Robot[1] in hospitality is a concept for Real-time task scheduling and allocation or distribution for a team of hospital robots. VisBug algorithm for hospital transport is used in Helpmate[2]. Single/ multiple lines following mobile robots are implemented in Tray carrying hospital robots[3]. Implementation of a face recognition neural network is given in [4] which was used for integration with ROS for daily attendance application. For telemedicine application, referred [5] which is an implementation of WebRTC for video conferencing. An appropriate task allocation algorithm for multiple robots has been explained in a Hospital Logistics Robot [6]. Implementation of Mask detection network[7] is integrated to ROS Architecture for better intelligence of Covid Robot. The study[9] explains multiple path planning algorithms for a team of robots in a hospital environment. Inspection Robots[10] make queries to the obstacle's avoidance strategies table, and safely avoid obstacles. There are multiple path planning algorithms for a team of robots in a hospital environment is explained in [11]. [12] explains different path planning techniques for coverage path planning. ORB-SLAM Technique is explained in [13]. Robots in [14] and [15] use ROS Architecture for its mapping and navigation. The robot proposed serves the purpose of food/ medicine delivery, non-contact hand sanitation, automatic room sanitation, telemedicine, daily attendance, sound instruction, etc.

DESIGN AND FABRICATION OF MODEL

A. Robot Design

The proposed Quarantine Service Robot is a two-wheeled, low-cost robot that gives position and speed feedback to the ROS environment for proper mapping and navigation. The design is created using CAD software and exported as URDF for ROS simulation and hardware implementation. The 3D model created using CAD software is given in Fig.1.

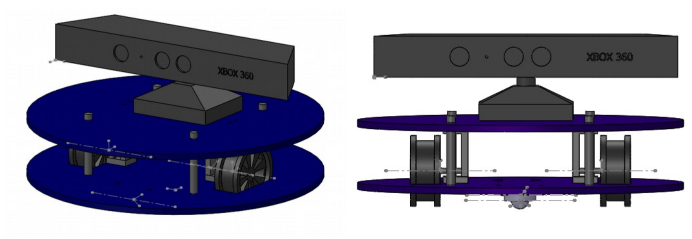

Fig. 1. Different views of 3D model of mobile robot

A rack for holding food/medicines is shown in Fig.2 and a tripod for holding tablet for the online doctor consultation is also integrated into the system. Both are detachable.

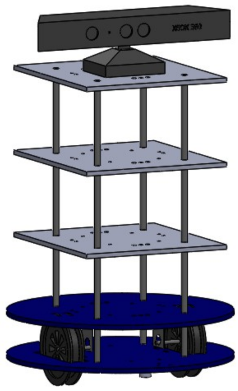

Fig. 2. Modified structure for food/medicine delivery.

B. Fabrication

Used CAD software drawing for laser cutting of acrylic sheet and assembled components properly as per the design shown in Fig. 3. DC motor with position encoder is used for the wheel control. Feedback data from the Inertial Measurement Unit(IMU) and position encoders are used for setting odometry, joint states, and transformation function.

Fig. 3. Hardware Model

This feedback data is required for proper localization and navigation of the service robot. The Kinect camera is used for getting the depth information for navigation. Camera sensor data is used for large distance mapping. IMU data used for short distance mapping. Combination of both provides better mapping and localization. The raspberry pi equipped with the system acts as slave and a remote computer acts as master. The non-contact hand sanitizer, spray sanitizer for room sanitation mode, tripod for fixing tablet for telemedicine chat functionality and racks are the detachable parts of the quarantine robot system. This is attached to the system as per requirement. For multiple task at same time, all parts can be attached.

C. Component Selection

- DC geared motors with position encoder: SGP30-60K
- IMU : MPU6050
- Microsoft Kinect Sensor
- Arduino Mega 2560 Board
- Raspberry Pi 3 Model B
- Servomotors : MG995

SYSTEM DESIGN

A. Overall Architecture

Fig. 4. shows the architecture of the overall setup. ROS installed Raspberry Pi is the brain of the mobile robot setup implemented. Raspberry Pi is connected remotely to the desktop system through an ssh connection. Arduino connected to Raspberry Pi takes data from the Inertial Measurement Unit and position encoders and uses this data for proper Odometry.

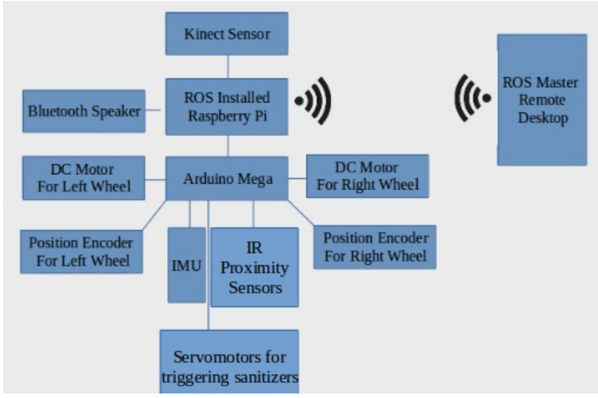

Fig. 4. Architecture of Overall Setup

The Raspberry Pi publishes Kinect Sensor data, Transformation Function, Odometry, and the Joint States to ROS Master. Localization of the robot in the Visualization tool Rviz uses feedback from IMU and position encoders. Data from IMU and Wheel encoders are mixed using EKF filter and the filtered odometry is used for mapping and navigation.

B. ROS – Mobile Robot Hardware Interface

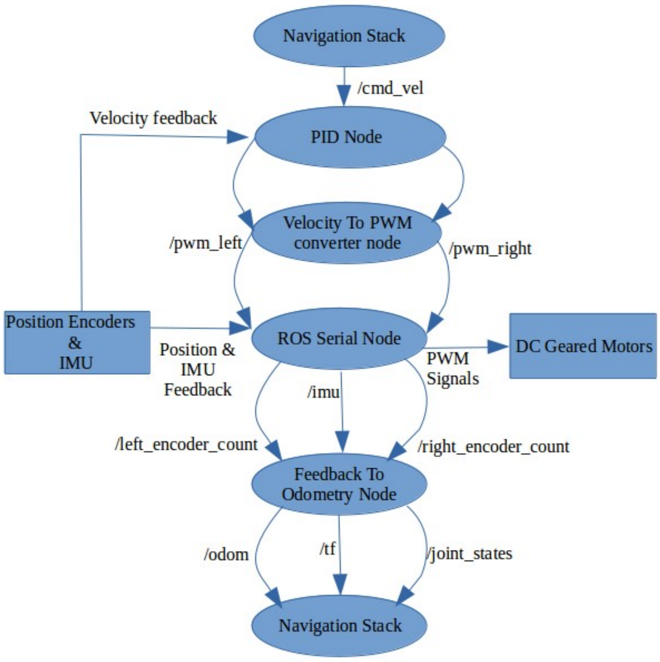

Fig. 5 . ROS – Mobile Robot Hardware Interface Architecture

Fig.5 shows the hardware interface of ROS and the mobile robot. The navigation stack provides the desired velocity to reach the goal position safely. It also estimates velocity at every instant and publishes it as a ROS topic named /cmd_vel. Navigation Stack has built-in navigation algorithms. The tuning of parameters is very crucial while working with the navigation stack. The PID Rospy Node controls the speed of the mobile robot. Then PID node calculates error by taking the difference between the desired velocity extracted from /cmd_vel topic and feedback velocity estimated from position encoders pulse count. The selection of K_P, K_I, and K_D values is essential in speed control.

The ROS – Hardware Interface is done through ROS Serial Communication. ROS Serial Node reads and writes to Digital I/O pins and publishes feedback data as ROS topics. This node also subscribes to the PWM values needed to control the motor. Navigation Stack computes the PWM data from the desired velocity published in the topic named /cmd_vel. Similarly, the ROS Serial node publishes topics that are used by other nodes for processing the feedback data and use for localization, mapping, and navigation.

C. Navigation Stack

The depth information from Kinect Sensor and inertial sensors data are used for navigation. Through the Openni ROS package, we can launch and receive the camera data as a topic. Through the Openni ROS package, we can launch and receive the camera data as a topic. The laser data that is converted from a depth image is used for mapping and navigation. The localization of the robot in the saved map makes use of the AMCL node. AMCL Node subscribes to the ROS topic named /map for preparing the global static map of the environment.

For local obstacle avoidance, it subscribes laser data topic named /scan. Initial pose and transformation function helps in localization. The /amcl_pose topic is used to identify the current position of the robot regarding the static map. Move base node controls navigation to the goal point. This node subscribes to the Goal topic and publishes the desired velocity to the PID controller. The motor control velocity is determined from the desired velocity and feedback velocity.

D. Daily Inspection Of Patients

The Deep learning Network will be trained using a dataset containing images of patients. Fig. 6. shows the flow diagram for the daily attendance system in isolation wards. ROS Node programmed for face recognition[4] and mask detection[7] subscribes to Kinect camera image topic and processes it.

If the mask is detected, the robot will instruct the patient to remove the mask for marking the daily attendance. Then it detects the face and identifies the name of the person in the camera frame. The name of the detected person in the frame will be added to the database for daily attendance. The robot will greet the person using the name.

Training of the network is done using the face pictures of persons in quarantine. So that the network will be able to identify the person, update the name to the database, and greet the person using the name. Then the robot will instruct the patient to wear a mask before leaving. This program logic is based on the consideration that patient is in COVID -19 isolation ward. For any other disease, can make slight changes in the program logic easily. ROS- Deep learning interface system is very flexible.

E. Non – Contact Hand Sanitizer

Fig. 7. shows a detachable non-contact hand sanitizer equipped with the service robot. When the patient brings his hand near the bottle, the IR proximity sensor senses the obstacle. Based on the signal from the IR proximity sensor, the low voltage DC Water Pump dispenses sanitizer liquid through the tube. A nozzle will be provided at end of the tube to control the flow.

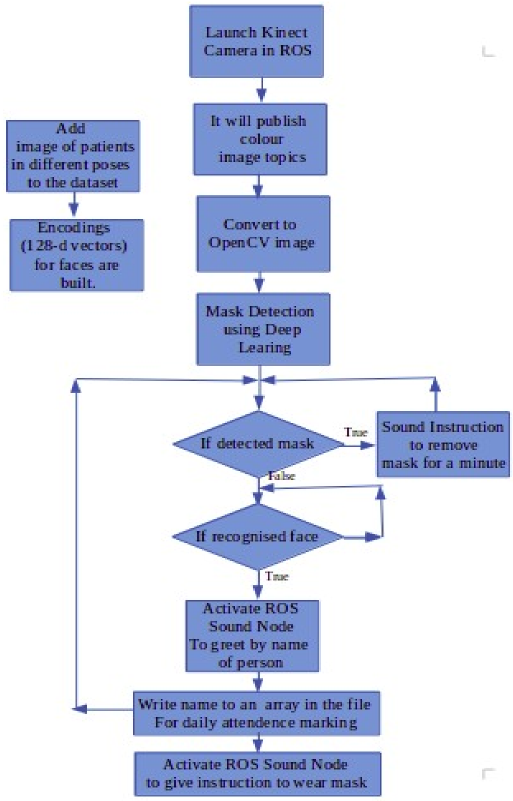

Fig. 6. Robot Inspection of patients, greeting and logging

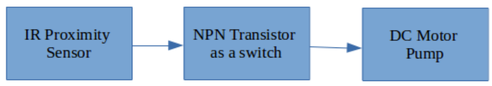

Fig.7 . Automatic Hand Sanitizer Dispenser System

F. Room Sanitation Mode System Design

Room sanitation mode is another feature of the quarantine service robot. A rospy program for the random movements throughout the room is implemented for this purpose. The spray sanitizer attached to the service robot is triggered mechanically during the cleaning mode. The robot wanders around the room randomly without colliding obstacles and sprays sanitation liquid continuously. Thus the entire room can be sanitized in a short time.

G. Telemedicine using quarantine service robot

Dedicated Video Conferencing Application is developed for Telemedicine. The doctor can sit in his room and select the patient to consult. The mobile robot will go to an individual room based on the amcl_pose set in the program. The patient can sanitize his hands using a Non-contact sanitizer attached and accept the request for a video call. After online doctor consultation, he can drop the call. During the doctor rounds time, the robot will go to individual rooms as per selection by the doctor and will be able to consult patients remotely. The doctor can decide whether the mobile robot should move to the room of another patient or return to the home location. This is the idea behind telemedicine using this robot.

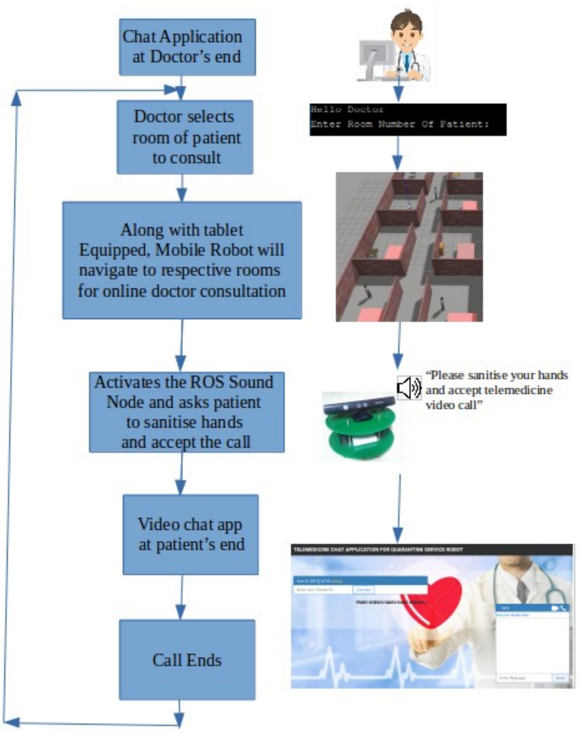

Fig. 8. Algorithm of Indirect Doctor Consultation System

H. Food and Medicine Delivery

The Fig.9. shows the flowchart for food and medicine delivery. The service robot navigates to the specified room using the hybrid ROS-Algorithm. Then instructs patient to take the materials. After delivery, the robot will go back to the home location.

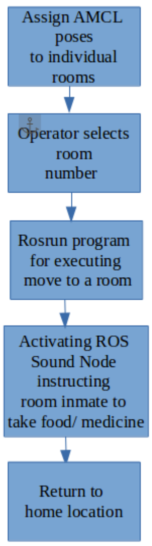

Fig.9. System description of Food and Medicine Delivery

SIMULATION

A. System Integration

This project has significance in the current scenario and will help to reduce the spread of diseases by minimizing social contact with the infected and suspected cases. The quarantine service robot proposed and implemented in this research work will help to do services by avoiding surface contact. The robot will be able to do the services needed by infected people in isolation wards without the need for human presence. The robot will also recognize the face of the patient and update the status in the patient attendance register. This data is significant in the current situation since many disease suspected persons try to escape from isolation wards. This feature can avoid daily manual inspection by a human. The quarantine service robot can navigate safely through its environments and reach the given room number and deliver services like food, medicine, etc. The robot will give light and sound command indication for the delivery trays assigned to respective rooms when it reaches the destination room. Non-contact hand sanitizer is attached to the mobile robot system so that the patients can use sanitize hands without hand touch. The robot takes care of room sanitation also. In sanitation mode, the disinfectant spray is triggered and thus serves the purpose. The user can stop the facility when required. Greeting a person by identifying the name of the patient is also given as a feature.

Dataset of the trained deep learning network contains face images of suspected cases or patients. After the successful encoding of the images in the dataset, the network will be able to recognize the face of the patient/suspect case. This system is useful for the daily inspection of patients for attendance. This research suggests the combination of two ROS navigation algorithms for better path planning. This algorithm decreases planning time and distance traveled through an optimized path planning algorithm. Generated a map of the isolation ward using the SLAM algorithm. The operator controls the remote robot and assigns the tasks to the robot through another ROS-installed master system. The robot navigates to respective rooms as per instruction from the operator. The service robot can navigate through dynamic environments and deliver services needed by the patient. The subsystems of system integration are as follows.

B. Mapping Gazebo Environment

Mapping can be accomplished in two ways. A program for random movements around the simulation world can perform the automatic mapping of a closed environment. The teleoperation node can perform the manual mapping of the environment. The depth information of the image converted to laser scan data named /scan used for navigation. Fig. 10. shows the custom Isolation Ward created in Gazebo. Fig. 11 shows the Gmapping technique. The Gmapping Node subscribes to the converted laser topic. The laser topic is the information regarding obstacles in the local cost map. The map server publishes the ROS topic named /map. Once mapped, the robot can navigate to any goal point freely. Fig. 12 shows a 3D Mapping Technique called Octomapping. It provides three-dimensional information about its surroundings. Gmapping is selected for the service robot.

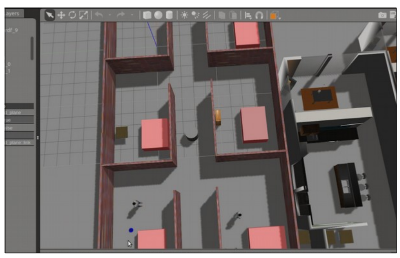

Fig. 10. Isolation Ward Created in Gazebo For Simulation

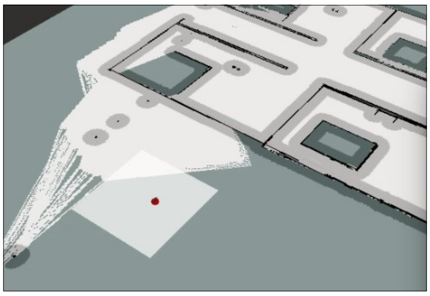

Fig. 11. Mapping of simulation world

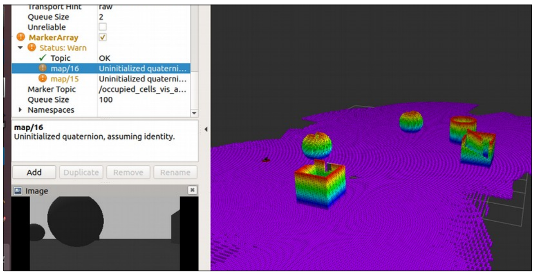

Fig. 12. Octomapping for 3D Map

C. Navigation in Gazebo

Navigation of the robot in the simulation environment is done using AMCL and move base nodes. AMCL helps in localization and move base node controls navigation. Fig. 14 shows navigation in the Gazebo world. Move base node receives a goal position and sends appropriate command velocities to drive the differential driven quarantine service mobile robot. The path to reach the goal position is planned through the combination of DWA Path Planner and TEB Local Path Planner. The path is indicated by the green line. The hyperparameters of these path planners are tuned in such a way that the robot should be able to plan an optimum path towards a goal location. So the hybrid tuned algorithm plans an optimum path that can avoid dynamic random obstacles also. TEB Path Planning algorithm and DWA Local Path planner are combined to give an optimum path plan having dynamic obstacle avoidance for the robot to move to the goal. A static map is provided to the AMCL node for collision avoidance and planning.

D. Simulation in application perspective

In the simulation world, the robot localizes itself using AMCL and finds the shortest path to the room by using path planners. ROS Sound Node should be launched for enabling the instruction feature of the robot. There are five features the robot can accomplish. The robot will perform the task assigned after reaching the goal. The Fig. 13. shows the simulation of the inspection task. The robot finds the shortest path to the specified room number, instructs to remove the mask if the mask detector deep learning module detects the mask, then recognizes the face of the patient using a trained neural network. Then the robot will greet the patient using his/her name and log the attendance to the database.

HARDWARE IMPLEMENTATION

A. System Dynamics

Dynamics for extraction of command velocity topic from Navigation stack and given as motor control signals are given below.

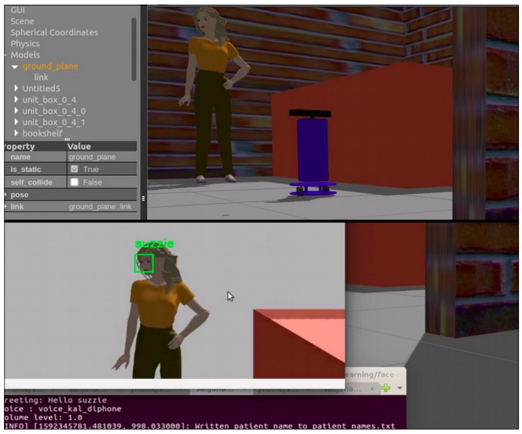

Fig. 13. Face recognition and logging to database in simulation environment

Fig. 14. Navigation using the hybrid algorithm proposed.

For a Differential Driven Robot, linear velocity v is the linear x value obtained from the topic. Angular velocity ω is the angular z value obtained from the topic. Desired right wheel velocity in terms of linear and angular velocities of the vehicle is

$$\omega_r = (v/r) + ((D * \omega)/(2*r)) \quad (1)$$

Desired left wheel velocity in terms of linear and angular velocities of the vehicle

$$\omega_l = (v/r) - ((D * \omega)/(2*r)) \quad (2)$$

where

r – wheel radius
D - distance between wheels

From the ticks from Position encoders, feedback velocity of left and right DC motor wheels is calculated.

For implementing PID Controller, we need to find the error of each motor

$$e = \omega - \omega_{fb} \quad (3)$$
$$\omega_{control} = (K_p * e) + (K_i * \int e) + (K_d * \dot{e}) \quad (4)$$

where

e - Error
ω - Desired wheel velocity

ω_{fb} - Feedback velocity
$\omega_{control}$ - Control velocity
$\dot{e}$ – *Derivative of error*
$\int e$ – *Integral of error*

The feedback topics are generated from the position encoder and the inertial measurement unit.

B. Real-World Mapping Process

Fig. 15 shows the mapping process of the real world. Mapping parameters like minimum_score, transform_publish_period is tuned for better mapping of the real environment where the quarantine service robot is installed. The Map server performs mapping using the gmapping node and successfully saves the map.

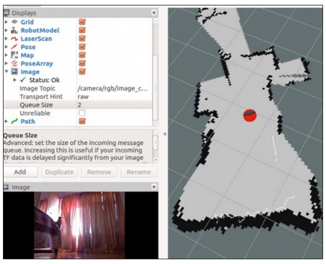

Fig. 15. Mapping Of Real Environment

C. Navigation in Real World

A combined algorithm for navigation using TEB path planner and DWA Path Planner helps in optimized path planning and navigation. Navigation makes use of depth image converted laser data obtained from the Kinect camera.

D. Inspection for Daily Attendance

For the integration of Deep Learning with ROS, subscribe to Camera Topic, convert to CV image, and write as an image file. Fig. 16 shows the mask detection and face recognition output frame. The deep learning network which is trained to detect masks instruct the patient to remove the mask during attendance marking time. The neural classifier network is trained using 2 categories of images. Categories are ‘No mask’ and ´With Mask´[7]. The facial landmarks are also considered for mask detection. The training using thousands of images helps the classifier to classify to mask/no mask category for the real-time input.

When the patient removes the mask as per the instruction from the robot, the trained network recognizes the face of patients in the isolation ward by reading each frame from the saved image file. Face recognition neural network is trained using the dataset of patients in the quarantine center along with their names. Face recognition module[4] is used for the same. Delay caused by saving and reading from image files affected the overall performance of the system. ROS - Deep Learning bridge program is developed to overcome this delay. The program is coded in such a way that it uses text to speech conversion features of ROS and greets the patient when the person is recognized for the first time. The name of the person is updated to the daily attendance database.

E. Non- Contact Hand Sanitizer

IR Proximity Sensor fixed on one side of the hand sanitizer for the detection of hand senses hand. If an obstacle is detected, it triggers a signal to the NPN transistor which acts as a switch. This activates the DC Water Pump and the sanitation liquid is dispensed without human contact. Diseases like COVID 19 spread through surfaces also. This system helps to avoid surface contact and reduce spread.

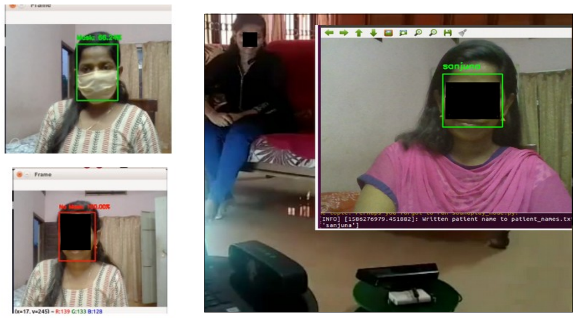

Fig. 16. Mask detection and Face Recognition for daily attendance

F. Room Sanitation Mode

The robot can be switched to room sanitation mode. In this mode, DC Motor triggers sanitation spray, and the robot moves randomly throughout the room. The room will be sanitized. The Fig. 17 shows detachable equipments designed for non-contact hand sanitation and automatic room sanitation.

Fig. 17. Non-contact Hand Sanitizer & Room Sanitation Spray

G. Telemedicine service equipped on the service robot

For telemedicine, developed a simple video chat service that works on a local network. Fig.18 shows the concept of telemedicine using a service robot. The user interface of the system is designed using HTML, CSS, and Javascript. Doctors can consult patients in quarantine through video conferences. Real-time communication is based on WebRTC[5]. A locally hosted PeerJS server that establishes a stable and secure connection between clients. Non-Contact Digital Infrared Forehead Thermometer can be equipped with a mobile base for fever detection.

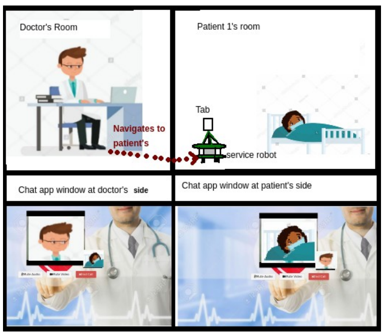

Fig.18. Telemedicine facility integrated on the service robot

A pulse sensor can also be integrated with the service robot to monitor the heartbeat rate. The temperature of the forehead and pulse rate can be sent to the doctor who is consulting the patient remotely through video conference.

EXPERIMENTAL RESULTS

Fig. 19 shows the velocity profile of the mobile robot developed. The X-axis of the graph denotes time. The Y-axis of the graph denotes velocity at the individual axis. Robot motion is smooth on sudden brakes and sudden acceleration. The linear velocity slowly increases and decreases as per the velocity commands from the navigation stack. This is the plot when mobile robots start moving after rest state.

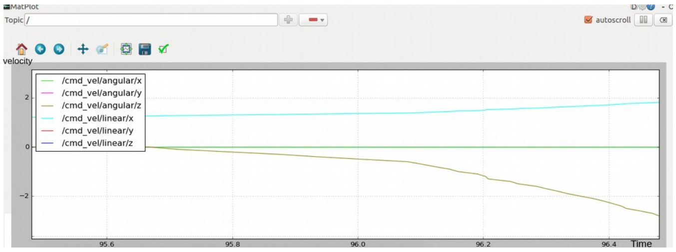

Fig.19. Velocity profile of the robot

For dynamic path planning, a hybrid ROS Navigation Algorithm combining DWA Path Planner and TEB path planning algorithm is used. The cost of traversing through the grid cells is computed and uses this value function to determine linear and angular velocities to send to the robot. DWA Algorithm samples the control space of the robot to dx, dy, and dtheta. It picks up the highest-scoring trajectory in such a way by avoiding illegal trajectories prone to collision. The algorithm will send the appropriate velocity to follow the path. The behavior of the ROS Planner can be tuned using the ROS parameters of the navigation stack. Forward Simulation Parameters sim_time is adjusted for better performance. Low values will provide simple arcs, and high values will provide long curves. So sim_time is set as 2 for the system designed. The inflation radius is set to 0.2m so that the robot can move close to walls for sanitation purposes. DWA planner[8] plans very slowly and is not much stable. Max/min velocity and acceleration are also modified. It will be able to find an optimal path considering time and distance. Using the DWA algorithm, the mobile robot avoids slowly moving dynamic obstacles.

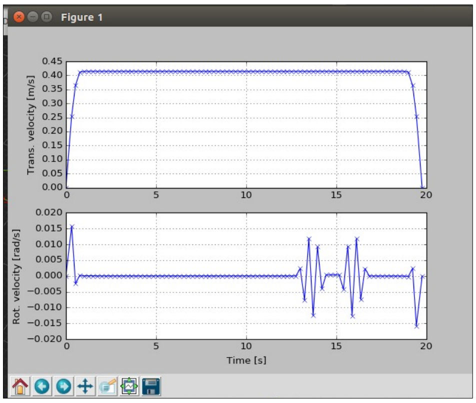

Fig. 20. Translational and Rotational Velocity profile using modified TEB Path Planner

Considering the distance from obstacles, execution time, and runtime kinodynamic constraints, Timed Elastic Band locally optimizes the robot's trajectory. Modified parameters to improve the performance of the path planner. TEB algorithm[6] plans better in the case of fast-moving obstacles. Fig. 20 shows Translational and Rotational Velocity using modified TEB Path Planner. When the robot starts moving towards the goal, translational velocity suddenly increases from zero to a constant velocity. Translational velocity is constant during the motion. The moving obstacles block the preplanned path at the 13th second. The robot changes its orientation at this point to avoid a collision. It continuously tries to find a better path considering the future motion of moving obstacles. At the 17th second, the robot can move in a particular orientation to the goal without much variations in rotational velocity. Instead of reducing translational velocity, the robot moves continuously at a constant speed and changes the direction of movement in case of any obstacle in the preplanned path. The robot will move in the updated path towards the goal. The translational velocity starts decreasing smoothly from the 19th second and becomes zero on the robot stop pose. The rotational velocity also becomes zero on the robot stop pose. When the penalty_epsilon parameter is 1.0, the robot becomes stuck at narrow passages. So value is reduced for better response. Update rate of the local cost map is increased for reducing execution time. TEB Path Planner gives a time-optimal solution. This algorithm can optimize multiple trajectories in different topologies. Initially, a path is planned and is transformed into a timed elastic band. At each loop, the path planner dynamically deletes previous values and inserts new configurations for adjustments on the remaining trajectory to be planned.

Fig. 21 shows a comparison of execution time between DWA, TEB, and hybrid path planners. The proposed hybrid path planner has better path planning and less execution time comparing to DWA and TEB Algorithm. Without any dynamic obstacles, the DWA planner and TEB planner took 44 and 38 seconds respectively for navigating to a fixed goal. The proposed system has an efficient path planning with an execution time of 34 seconds for navigating to the fixed goal. With obstacles, the DWA planner and TEB planner took 55 and 47 seconds respectively for navigating to a fixed goal. The proposed system has an execution time of 43 seconds for navigating to the fixed goal. Fig. 22 shows the new path planned by the robot when a dynamic obstacle is introduced in its way.

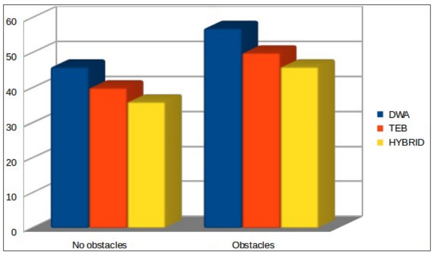

Fig. 21. Comparison of execution time between DWA, TEB and Hybrid Algorithm.

Using this navigation algorithm, robot will be able to navigate to different rooms given by the remote operator and thus deliver services to patients in quarantine by avoiding maximum human contact.

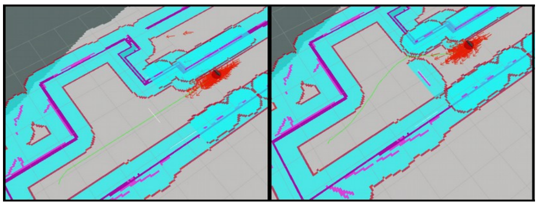

Fig. 22. Dynamic Obstacle Avoidance through Hybrid Algorithm

CONCLUSIONS

This paper proposes a quarantine service robot that can serve multiple purposes. The modified hybrid path planning algorithm takes less execution time comparing to the existing algorithms. Without any dynamic obstacles, the DWA planner and TEB planner takes an execution time of 44 and 38 seconds respectively for navigating to a fixed goal. Better performance is achieved by the proposed hybrid algorithm with an execution time of 34 seconds for navigating to the fixed goal. In a dynamic obstacle environment, the DWA planner and TEB planner showed 55 and 47 seconds respectively for navigating to a fixed goal. The proposed system has an execution time of 43 seconds for navigating to the fixed goal**.** Daily attendance marking system using trained neural networks integrated with ROS helps for an intelligent robotic inspection. The proposed system has a user interface for selecting different modes, to give delivery locations, etc. which are done using roslibjs which makes the system more users friendly.

This research can be extended by having multiple robots in the same hospital, all controlled by a center ROS Master system. Also can incorporate features like Cliff Detection, spot cleaning, Trash Can Localization, etc. in the future. ROS - deep learning integration is providing a wide range of scope for future applications. Currently, robots can detect the face only when the face is not covered. This is an expected area for future work. The five services provided by this low-cost service robot can reduce contact between health workers and the suspect cases to a greater extend. The system provides a cost-effective robotic solution to reduce the spread of communicable diseases and helps to automate hospital services.

REFERENCES

1. Simon Thiel , Dagmar Häbe , Micha Block, “Co-operative robot teams in a hospital environment” , IEEE International Conference on Intelligent Computing and Intelligent Systems 2009
2. J. Evans ,B. Krishnamurthy, B. Barrows,T. Skewis , V. Lumelsky, ¨Handling real-world motion planning: a hospital transport robot¨, IEEE Control Systems Magazine 1992
3. Y. Takahashi ,M. Kohda ,Y. Kanbayashi , K. Yamahira , T. Hoshi , ¨Tray carrying robot for hospital use¨, IEEE International Conference on Industrial Technology 2005
4. https://www.pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning
5. https://github.com/valarpirai/video-chat-app
6. Maison Glad Jeju, Jeju, "Performance Analysis of Scheduling Multiple Robots for Hospital Logistics" Korea June 28 - July 1, 2017
7. https://www.pyimagesearch.com/2020/05/04/covid-19-face-mask-detector-with-opencv-keras-tensorflow-and-deep-learning/
8. http://wiki.ros.org
9. Xianqun Huang , Qixin Cao , Xiaoxiao Zhu , “Mixed path planning for multi-robots in structured hospital environment”, The 14th Asian Conference on Computer Aided Surgery (ACCAS) 2018
10. Jinshan Yu, Hao Wu , Xin Wang , “The Research on Obstacle Avoidance of Hospital Ward Inspection Robots in Complex Environment” , Chinese Control and Decision Conference (CCDC) , May 2016
11. T. Skewis! J. Evans: V. Lumelsky! B. Krishnamurthy: B. Barrows, "Motion Planning for a Hospital Transport Robot" Proceedings of the 199 1 IEEE International Conference on Robotics and Automation Sacramento, California April 1991
12. Richard Bormann, Florian Jordan, Joshua Hampp, Martin Hagele, "Indoor Coverage Path Planning: Survey, Implementation, Analysis" IEEE International Conference on Robotics and Automation (ICRA) , 2018
13. Raul Mur-Artal, J. M. M. Montiel, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System" IEEE Transactions on Robotics , 2015
14. Abishek B, Gautham S, "ROS based Stereo Vision System for Autonomous Vehicle" IEEE International Conference on Power, Control, Signals and Instrumentation Engineering , 2017
15. Murat Koseoglu, Orkan Murat Celik and Omer Pektas ,"Design of an Autonomous Mobile Robot Based on ROS",IEEE , 2017