

Medicine Supplier Autonomous Robot Navigation in COVID-19 Pandemic

Amna Hameed¹, Muhammad Talha Shabbir² and Ahtisham Ali³

¹Department of Electrical, Electronics and Telecommunication Engineering, University of Engineering and Technology Lahore, Faisalabad, Punjab, Pakistan

²Department of Electrical, Electronics and Telecommunication Engineering, University of Engineering and Technology Lahore, Faisalabad, Punjab, Pakistan

³Department of Electrical, Electronics and Telecommunication Engineering, University of Engineering and Technology Lahore, Faisalabad, Punjab, Pakistan

Abstract

This paper describes an application of autonomous navigation robot system in current pandemic of Coronavirus by supplying medicine to make sure the touchless service by freely navigating under fixed and dynamic conditions regarding objects as obstacles and obtains the map of the indoor environment to successfully plan its path towards the desired goal. We are presenting a SLAM (Simultaneous Localization and Mapping) algorithm combined with the particle filter to estimate the location of robot in map of unknown environment and that map is being generated using odometry data from differential drive robot which solves the problem of mapping and localization accurately. Moreover, a Dijkstra algorithm coupled with the reactive motion planning algorithm known as Trajectory Rollout is used. The robot is navigating in the environment by PID controller implemented for the control of actuators. Results have been verified by real time experiment.

Keywords: SLAM, Navigation, Differential drive, Localization, Path Planning, COVID-19.

1. Introduction

From the previous few years, robots are widely used in almost every field of life and controlled by humans like drones & vehicles, however with the increased usage of human control things, many of the researchers started study on autonomous robot system i.e. robots can move from one place to other without human guidance. Moreover, the importance of mobile robots can be estimated from their recent

applications to tackle the COVID-19 virus originated from china, where mobile robots are deployed on the frontlines to supply the medicines and other things in a dynamic environment where humans and machines coexist. The purpose of this research is to provide a state-of-the-art framework considering current challenges of the pandemic situation so that further advancement can be made to obtain the better movement of socially assistive and service robots and to control the spread of COVID-19. The self-governing portable robot development in the human environment firmly influences apparent knowledge [1]. In this cutting-edge period, where robots are advancing from modern applications day by day [2], human and robot social interaction is required to turn the application and usage of robots into a significant ability in the up-coming age of social robots [3]. Robots are currently present alongside individuals in lodgings [4], medical services communities [5] and shopping centers. Many indoor service robots are deployed for spraying disinfectant and for the food delivery to persons infected with virus [6]. A differential drive robot will be used for implementing medicine supplier autonomous robot during COVID-19 Pandemic. So, we have deployed an application of supplying medicine in indoor environment by an autonomous robot and considering the needs of present conditions. Our system is a LIDAR-mounted differential drive robot. Other RGB-D sensors (Kinect can also be used but they do not provide laser scan data directly conversions are required to get the scan data which may slower the mapping frequency).

One main advantage of LIDAR is that it is not sensitive to light or illumination conditions, climatic changes or illumination effects, sunlight does not affect its efficiency. But its disadvantage is that the density of point cloud decreases with the distances because of divergence between the laser beams. We have divided this paper in sections, first half of Section II is describing the bright background of autonomous robots, how they were used in different applications by presenting a literature review, and second half of Section II is describing our working methodology and explaining the algorithms which we used, Section III is discussing the problems we faced during robot design to provide a suitable match with indoor environment along with a solution model as a final product and Section IV is describing the real time experiment and discussion of results.

2. Survey of literature and adopted methodology

2.1 Literature review

Simultaneous localization and mapping (SLAM) mainly consist of following three steps: initializing, tracking, and mapping [16]. A navigation system can be obtained by using techniques of computer vision. A 3D sensor is generating the point cloud of environment in which start, and goal position has been recognized by computer vision techniques and solving the problem of path planning by RRT algorithm [7]. 3D Path planning can also be carried out with the help of Kinect V2 sensor by implementing rapidly exploring random tree (RRT*) algorithm for generating collision free paths [7]. In [8] the SLAM is carried out by using RGBD camera and RRT algorithm on platform of Robot Operating System (ROS). Geometrical and the visual information acquired from the RGB-D data of Kinect sensor has been used for localization and object recognition techniques [19]. Moreover, LIDAR sensor can also be used in SLAM techniques like EKF-SLAM which is based on kinematics of vehicle [20]. In [17] SLAM algorithm combines oriented FAST and rotated BRIEF (ORB), and progressive sample consensus (PROSAC) algorithm has introduced as well. SLAM algorithms are very sophisticated and there are several solutions to perform SLAM, but they require higher computational cost for simultaneous localization and mapping. Its main steps include obtaining the map of indoor environment, locating its position in the map and move towards the desired goal autonomously by planning a path, for the mapping purpose an

occupancy grid maps implemented [9]. In [11], a path planning method from start to goal position has proposed using a global planner, continuously checking whether the unknown obstacles are colliding with robot with a threat assessment, if any, then an alternative path is produced. This paper [18] describes many approaches and classify different approaches for path planning and divide the path planning algorithms into the two main classes, named as, the Heuristic Approaches and Conventional Approaches.

Moreover, according to literature survey differential drive robots are mostly preferred by researchers for social navigation as compared to other drives like Skid steering and Ackerman steering. The reason of using the differential drive has been discussed in detail in section 3.

2.2. Simultaneous Localization and Mapping

Indoor environments are mostly composed of perpendicular or parallel planes, algorithm of orthogonal SLAM saves the memory usage by reducing the computational cost as this algorithm extracts the line segments of orthogonal shape. The Rao Blackwellized particle filter approach generates the map in which every particle pose is known. Most efficient maps are generated by orthogonal SLAM based on RBPF (Rao Blackwellized particle filter) approach. We are using Gmapping algorithm to generate map of indoor environment as Gmapping is graph based and the occupancy grid mapping of RBPF. Efficient maps are generated by Gmapping as it requires very less processing power and limited use of memory to show SLAM posterior by using very less numbers of particles. Gmapping generates the map by laser-scan data from LIDAR and odometry data obtained from encoder motors placed in wheels of robot [21].

There are two possible ways of localization: (a) global localization and (b) position tracking. In position tracking the robot is already aware of initial position and it eliminates the errors in odometry data occur during robot movement. But in global localization robot has no initial position and must find its location from scratch [22]. The probabilistic model is required to track the pose of robot in the unknown environment. An improved Monte Carlo Localization calculation is applied, expecting to construct a LIDAR-based robot localization framework in a perplexing and unstructured condition [12]. Monte Carlo localization solves problem of global localization. MCL shows the set of particles drawn with respect to posterior distribution

according to the poses of robot [23]. Process of MCL is in two ways. Firstly, along with the robot movement new N samples are generated to approximate the pose of robot with the motion. These are randomly generated samples produced by the probability values calculated from likelihood of previous set of sample values. Secondly, re-weighting of sample values is performed to incorporate the readings from sensor. $O(N)$ is the time complexity to efficiently process that resampling. Addition of random small samples helps MCL to re-locate the pose of robot effectively [22]. MCL's adaptive approach is used because it uses the adaptive sample set if robot meets uncertainties at point where its sample set grows to the threshold but still robot lost its pose. In this case adaptive methodology proves superior to the fixed sample set [22].



Figure 1: Four Wheeled Mobile Robot

2.3. Dijkstra algorithm for the global plan

Global motion planner is the first thing we do for motion planning in which a path is generated from starting position to the goal. It considers all length of the path. For global motion planning, Dijkstra's algorithm is a great calculation for finding the most limited way between two points. Adjacency matrix is a square matrix of the algorithm. This matrix has constrained the utilization of the algorithm as it grows [13]. The test result shows that the presented method of path planning created the most ideal way between two focuses when it is applied to the map of any indoor landscape. Dijkstra algorithm is

the breadthfirst search (BFS) calculation for estimating the shortest paths from a single source vertex to all different vertices. It forms vertices in expanding request of their good ways from source, which are known as root vertices. The shortest way between two vertices is a way with the shortest length, additionally known as link distance.

Let $U=(G,V)$ is the weighted undirected graph and it has the weight functions $w: E \rightarrow R$ that map edges to real-valued weights. If $e(g, v)$, then we write $w(g, v)$ for the $w(e)$. the length of the path p is the total length of the weights of the consisting edges. The distance from g to v is the length of the minimum path if there is a path from g to v otherwise it would be infinity. The idea behind Dijkstra's algorithm is that it reports the vertices in an ascending order of the distances from the vertex of source and construct the shortest path tree edge by edge and it add one new edge at each step corresponding to the construction of smallest path to current new vertex [14].

2.4 Trajectory Rollout motion planning algorithm for local plan

Local motion planning algorithms are the obstacle avoidance algorithms and generate trajectories on which robot have to move, they work on the nearest map of environment not on entire map like global planner. They can work without global planner but stuck on a local minima issue that's why a local planner comprised with a global one, is more feasible [15].

The basic concept of trajectory Rollout is as follows:

- In the control space of robot discretely do sampling of $(dx, dy, d\theta)$.
- For every sample of velocity, do the forward simulation from the robot's present state to anticipate what might occur if the tested velocity were applied for a few (brief) times frames.
- Assess (score) every direction coming about because of the forward simulation, utilizing the metric that joins attributes, for example, closeness to goal, nearness to the goal, vicinity to the global path, and the velocity. Dispose of illicit directions (those that slam into obstructions).
- Pick up the most noteworthy scoring direction and send the related velocities to moving base.
- Do this process again.

3. Robot Design Model

Figure1 is our four wheeled robot model with LIDAR on its top, which we design for navigation in indoor environment especially for COVID-19 Pandemic, but we faced some practical limitations. We analyze these limitations and try to find the possible solutions to remove or overcome these limitations. Let's discuss some of the practical issues and solutions of these issues.

When an experiment is performed in the real-time environment instead of a simulated environment, there are always some sort of practical limitations and errors. We faced problems with our four-wheeled mobile robot, it had three main problems as compared to a simulated four-wheeled mobile robot. The problems occurred and the techniques we adopted to avoid them are listed below in detail.

3.1 Non-linear Trajectory

The reasons robot followed non-linear trajectory (although same pulse width modulation (PWM) was applied to all motors) was that each motor had a bit difference in its rated torque value although all the motors were purchased from same manufacturer. As a result, as seen in the Figure2 PWM vs speed graph, Figure2(a) shows the No load condition while Figure2(b) on Full Load condition. Each motor deviates from the speed when same PWM voltage was applied. Ideally all the four graphs must overlap each other.

3.1.1 Both graphs with & without load

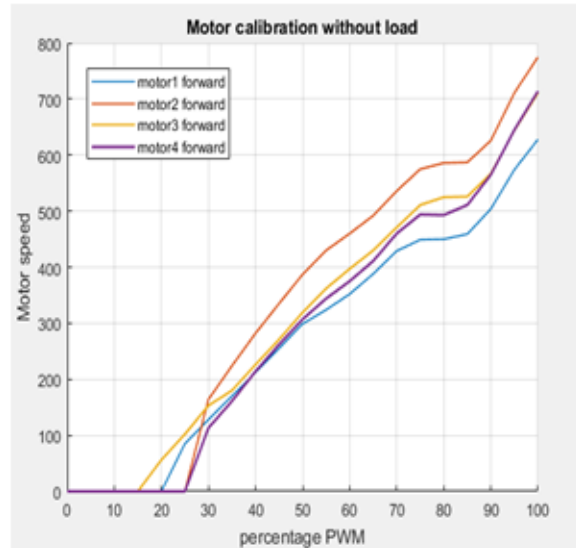


Figure 2(a): Motor speed graph at different values of PWM at no load

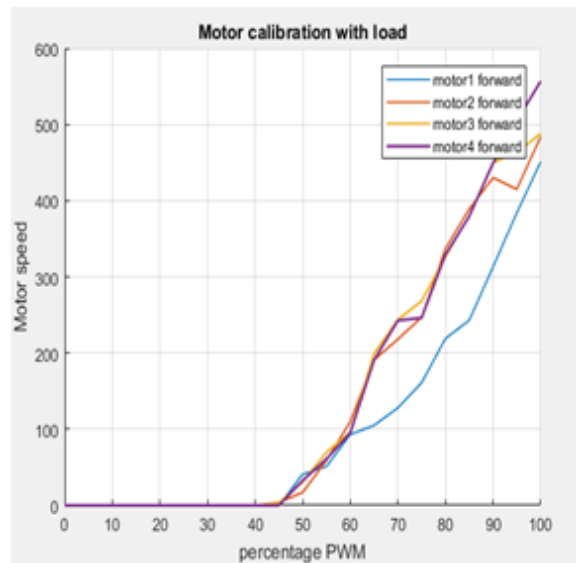


Figure 2(b): Motor speed graph at different values of PWM on full Load

To remove this error, a PID controller was implemented on the motors with the algorithm shown in following diagram:

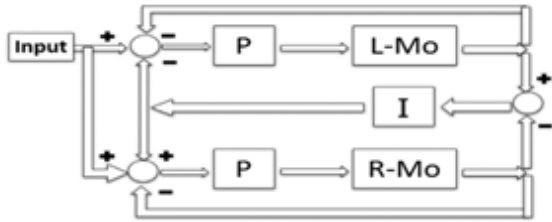


Figure 2:PI Controller Implementation

This PID controller keeps an eye on each rotating motor at a given target speed by controlling the motors individually and if somehow a motor still doesn't achieve the target speed due to various reasons i.e. heavily overloaded, jammed, its supply path cut-off etc. PID controller will slow down all the other motors to have a speed equal to one with lowest speed.

3.2 No rotation about axis

When the robot model was four wheeled, there were two circles of rotations for the mobile robot to rotate about its axis shown in Figure4. It can be clearly seen that there was not even a little intersection between these circles, mean there was no central point of rotation for the robot i.e. why it was not rotating. To counter this problem, the motors were closely placed so that there could be an intersection between these two circles and robot could have point of rotation. Thus, we were somehow able to rotate around its axis.

3.3 Skid-steering

Although in second step, the robot was able to rotate about its axis, but this was not a perfect rotation as the robot now had too many points of rotation. As a result, the wheels were not rotating alone instead they were dragged as well during the rotation creating a problem called skid-steering. This led to inaccurate measurement of encoder ticks and was unable to measure that how much degree the robot had been rotated which causes the failure in trajectory which it was following. The only way to avoid this error was to have only one point of rotation which was not possible by intersection of two different circles. So, it was required to re-design the robot from four-wheeled to two-wheeled mobile robot with two caster wheels.

Our new robot model which is differential drive robot can be seen in Figure 6 it doesn't have the above mentioned issued and it is now perfect for the navigation of indoor environment however it also has drawback of not capable of carrying much weight and slippage problems.

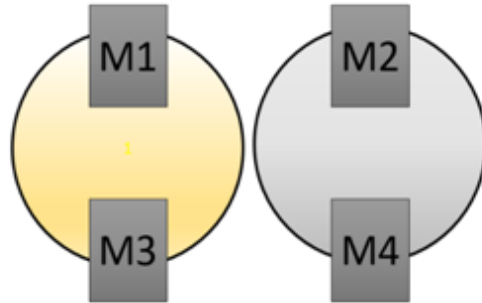


Figure 4:Multiple point of rotations

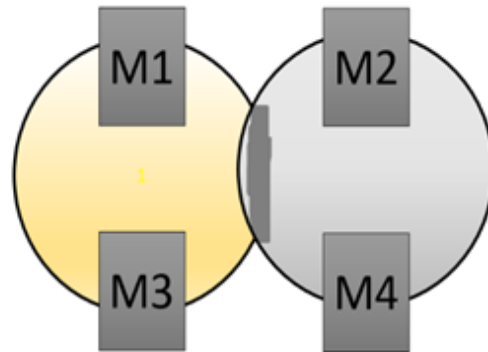


Figure 4: No common center of rotation



Figure 5: Differential drive robot

4. Results and Discussion

4.1 Localization with Adaptive MCL

Direct map is obtained from external sensors in a standardized format. In our system we are using a sensor model to generate map by the mobile robot which is taking reference from vertical planes so that a map update is also performed. For localization, our main concern is to find the current pose of base frame of robot in map frame. It locates itself in the map of unknown environment by using Adaptive Monte Carlo Localization (AMCL) [10]. AMCL matches the laser-scans with the generated map to detect if there is any drift in the pose estimate based on data of odometry coming from mobile robot. This algorithm reduces the errors in the position estimation of robot in the map very effectively. After mapping and localization our goal is path planning. For obtaining the map of indoor environment ROS based Gmapping intakes the transform from /laser → /base_link and uses the odometry information. We run Map_server to save the generated map, and dimensions of map was stored in the yaml file format. Then for estimating the pose of mobile robot in that map of indoor environment, run the adaptive Monte Carlo localization with particle filter and at the same time change in the Rviz window can be noticed. While moving the robot with path planners, the position of the robot can be estimated by the AMCL technique. Arrow in the below figure is

showing the 2D Pose Estimate and result is shown in Figure 7.

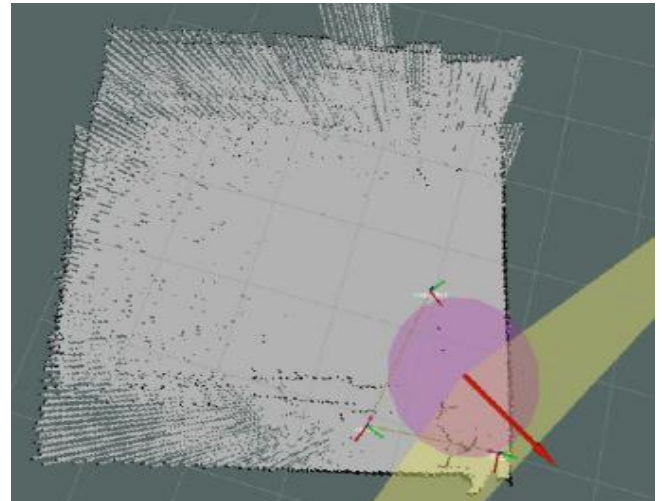


Figure 6: Pose Estimation.

4.2 Path Planning

For the global path planning we used the Dijkstra algorithm and for local path planning we used trajectory rollout planner. Global Path planner provides the trajectory of whole path considering the environment, starting, and ending point and the static obstacles while local path planner only provides trajectory of close spaces of robots mainly used for avoiding obstacle.

Holding the velocity constant and varying the steering angle gives candidate set of trajectories.

$$X_n = \sum_{i=0}^{n-1} v_i \cos(\theta_i) \Delta t = X_{n-1} + v_{n-1} \cos(\theta_{n-1}) \Delta t$$

$$Y_n = \sum_{i=0}^{n-1} v_i \sin(\theta_i) \Delta t = Y_{n-1} + v_{n-1} \sin(\theta_{n-1}) \Delta t$$

$$\theta_n = \sum_{i=0}^{n-1} v_i \frac{\tan(\gamma_i) \Delta t}{L} = \theta_{n-1} + v_{n-1} \frac{\tan(\gamma_{n-1}) \Delta t}{L}$$

Figure 8 is the implementation of the Global planner as well as the local planner. Blue hexagonal shape is the robot footprint and around that footprint is the robot space. Black lines are the walls or static obstacles and beside the walls is the cost that decrease as we move far from the walls or obstacles. Red arrow is showing the direction / pose / face of the Robot. Green line on the map is the global plan that's obtain using the Dijkstra algorithm from starting position of the robot to the destined position while orange line is the local plan that obtain by using the Trajectory Rollout algorithm. This planed path by planners can change if robot faces some dynamic obstacle like humans. Axis's below

the footprint of the robot is the transformation of the Robot frame to odom frame and then into map frame. Map frame is the arbitrary frame that we assume anywhere on the map generally we take below left corner. Odom frame is the Robot frame at $t=0$. Map frame and odom frame both are stationary frames while robot frame is the moving frame that moves corresponding to the robot.



Figure 7: Global & local Plan

4.3 Real Time Verifications

We verify the result by implementing in real time environment, we performed experiment on different types of floors like cement floor, tiles floor, bricks floor and finally carpeted floor. The purpose of checking on different floor is to verify the odometry of the Encoder motors. As on different floor there is possibility of getting different problems of slips in odometry. If it is not nullified, it starts accumulating that cause to fail the robot to reach desire position. By performing different experiment, we observe that maximum slip occurs on bricks floor due to the not smoothness of the floor. But it is mainly built to handle the COVID-19 Pandemic Situation. So, mostly we have tiles floor or carpeted floor. That's why we ignore the slip problem on the brick floor. Figure 9 showing the motion of the robot on the carpet floor assuming COVID patient is on the bed in his room. In the Figure9 (a) left side is the real time snaps while right side is the cost map of the room on the RVIZ. In figure 9 (a), it is taking the goal from the doctor or caretaker assuming him outside the room and wants to give medicines to the patients and set the goal near the patient. So, in figure 9(b) it starts moving toward the goals by using the global and

local planner. In figure 9(c), Robot reach the desired locations and then robot similarly takes its way back, after supplying medicine and other things in this COVID-19 pandemic. This robot is freely navigating in the indoor environment (specially Home and hospitals) by planning the most suitable path from starting to the goal position to use for the Medical Supplies to COVID-19 patients.

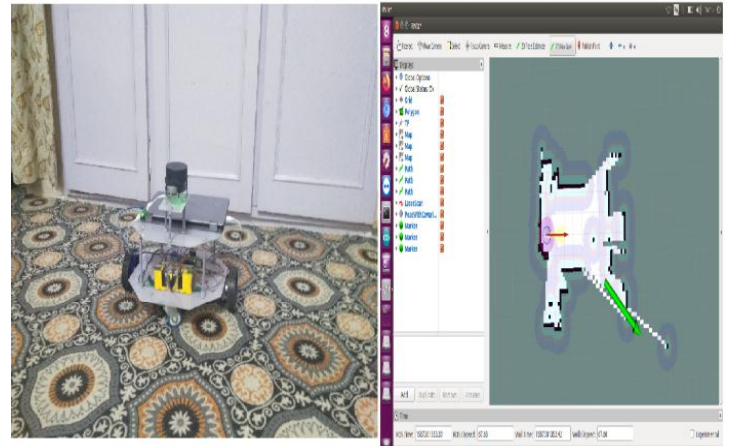


Figure 8(a): Getting the Goal for Robot

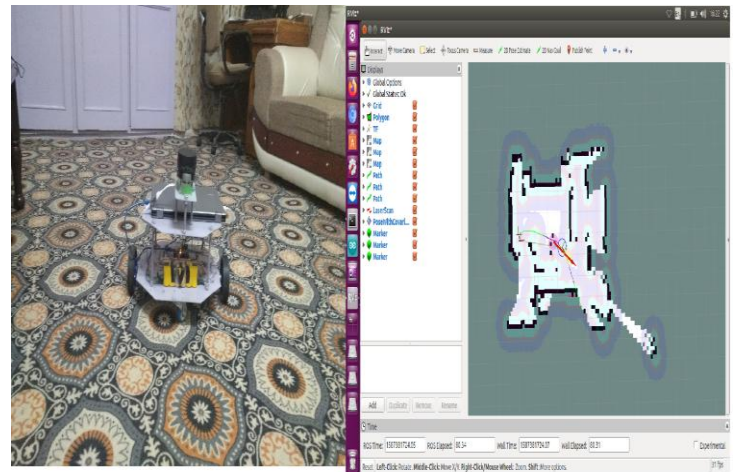


Figure 8(b): Navigation towards the goal

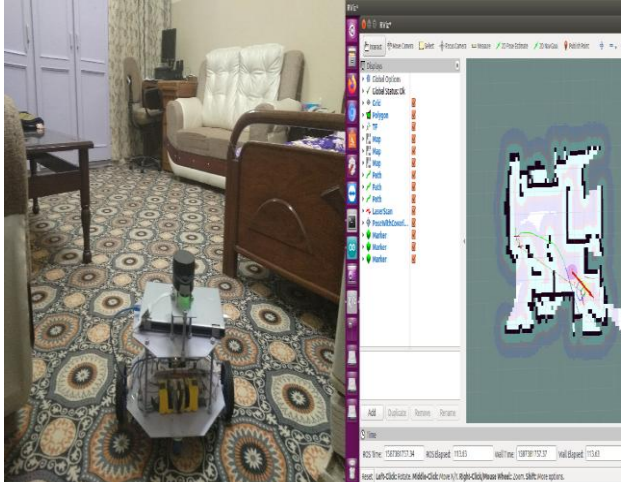


Figure 8(c): Reached at the goal

4.4 Error Calculation

In error calculation we compared the velocity data from real time calculations of local planner with the real time velocities calculated from the encoders because ideally both should be equal in order to achieve a perfect navigation and making it sure that encoders are following the local planner but due to some practical limitations we got some errors. So, these errors need to calculate in order to remove or compensate for odometry calculations. If it is not removed it can start accumulating and robot will finally fail to reach the desired goal.

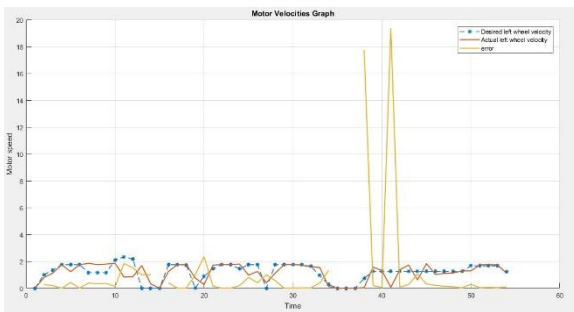


Figure9: Error calculation of two velocities

5. Conclusion

In this paper we are presenting the model of the differential drive robot and locating its position by SLAM algorithm based on pose estimation technique known as Adaptive Monte Carlo localization with particle filter in map of indoor environment to avoid the robot kidnapping problem in which robot loses track of its location. We have obtained the map by ROS based Gmapping algorithm as it is very efficient and quick method for map generation among other map generating algorithms. Most suitable local and

global path planners were integrated to provide the robot a shortest and collision free path in indoor environment. This robot is freely navigating in the indoor environment (specially Home and hospitals) by planning the most suitable path from starting to the goal position mainly use for supplying medicine and other thing in this COVID-19 pandemic. Future aspects that we can predict about this robot is that it can formulate to plan its path without disturbing human space for example if a person is taking photograph of another person or two persons are embracing each other so it should not cross these human spaces and should plan its path by avoiding obstacles and humans as well.

6. References

- [1] Vega, A., Cintas, R., Manso, L. J., Bustos, P., & Núñez, P. (2019, November). Socially-accepted path planning for robot navigation based on social interaction spaces. In *Iberian Robotics conference* (pp. 644-655). Springer, Cham. Interaction Spaces." Iberian Robotics conference. Springer, Cham, 2019.
- [2] Rosete, A., Soares, B., Salvadorinho, J., Reis, J., & Amorim, M. (2020, February). Service robots in the hospitality industry: An exploratory literature review. In *International Conference on Exploring Services Science* (pp. 174-186). Springer, Cham.
- [3] Gómez, J. V., Mavridis, N., & Garrido, S. (2013, November). Social path planning: Generic human-robot interaction framework for robotic navigation tasks. In *2nd Intl. workshop on cognitive robotics systems: replicating human actions and activities*.
- [4] Chan, A. P. H., & Tung, V. W. S. (2019). Examining the effects of robotic service on brand experience: the moderating role of hotel segment. *Journal of Travel & Tourism Marketing*, 36(4), 458-468.
- [5] Ramoly, N., Bouzeghoub, A., & Finance, B. (2018). A framework for service robots in smart home: an efficient solution for domestic healthcare. *IRBM*, 39(6), 413-420.
- [6] Lafaye, J., Gouaillier, D., & Wieber, P. B. (2014, November). Linear model predictive control of the locomotion of Pepper, a humanoid robot with omnidirectional wheels. In *2014 IEEE-RAS International Conference on Humanoid Robots* (pp. 336-341). IEEE.
- [7] Lafaye, J., Gouaillier, D., & Wieber, P. B. (2014, November). Linear model predictive control of the locomotion of Pepper, a humanoid robot with omnidirectional wheels. In *2014 IEEE-RAS*

International Conference on Humanoid Robots (pp. 336-341). IEEE.

[8] Nie, W., Li, Q., Zhong, G., & Deng, H. (2017, October). Indoor 3D path planning using a Kinect V2 sensor. In *2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC)* (pp. 527-531). IEEE.

[9] O'Callaghan, S. T., & Ramos, F. T. (2012). Gaussian process occupancy maps. *The International Journal of Robotics Research*, 31(1), 42-62.

[10] Ayawli, B. B. K., Chellali, R., Appiah, A. Y., & Kyeremeh, F. (2018). An overview of nature-inspired, conventional, and hybrid methods of autonomous vehicle path planning. *Journal of Advanced Transportation*, 2018.

[11] Zhang, L., Sun, L., Zhou, L., Zhang, X., & Liu, J. (2016, December). Path planning for the mobile robots in the environment with unknown obstacles. In *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (pp. 1153-1158). IEEE.

[12] Peng, G., Zheng, W., Lu, Z., Liao, J., Hu, L., Zhang, G., & He, D. (2018). An improved AMCL algorithm based on laser scanning match in a complex and unstructured environment. *Complexity*, 2018.

[13] Sun, L., Liu, X., & Leng, M. (2006, June). An Effective Algorithm of Shortest Path Planning in a Static Environment. In *International Conference on Programming Languages for Manufacturing* (pp. 257-262). Springer, Boston, MA.

[14] Fadzli, S. A., Abdulkadir, S. I., Makhtar, M., & Jamal, A. A. (2015, December). Robotic indoor path planning using dijkstra's algorithm with multi-layer dictionaries. In *2015 2nd International Conference on Information Science and Security (ICISS)* (pp. 1-4). IEEE.

[15] Gerkey, B. P., & Konolige, K. (2008, May). Planning and control in unstructured terrain. In *ICRA Workshop on Path Planning on Costmaps*.

[16] Taketomi, T., Uchiyama, H., & Ikeda, S. (2017). Visual SLAM algorithms: a survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9(1), 1-11.

[17] Xin, G. X., Zhang, X. T., Wang, X., & Song, J. (2015, December). A RGBD SLAM algorithm combining ORB with PROSAC for indoor mobile robot. In *2015 4th International Conference on Computer Science and Network Technology (ICCSNT)* (Vol. 1, pp. 71-74). IEEE.

[18] kanta Choudhury, Rama, Sanjay Kumar, and Chandra Kanta Samal. "Mobile Robot Path Planning Approach: A."

[19] Vaskevicius, N., Pathak, K., Ichim, A., & Birk, A. (2012, May). The jacobs robotics approach to object recognition and localization in the context of the icra'11 solutions in perception challenge. In *2012 IEEE International Conference on Robotics and Automation* (pp. 3475-3481). IEEE.

[20] Wang, D., Liang, H., Mei, T., Zhu, H., Fu, J., & Tao, X. (2013, June). Lidar Scan matching EKF-SLAM using the differential model of vehicle motion. In *2013 IEEE Intelligent Vehicles Symposium (IV)* (pp. 908-912). IEEE.

[21] Abdelrasoul, Y., Saman, A. B. S. H., & Sebastian, P. (2016, September). A quantitative study of tuning ROS gmapping parameters and their effect on performing indoor 2D SLAM. In *2016 2nd IEEE international symposium on robotics and manufacturing automation (ROMA)* (pp. 1-6). IEEE.

[22] Fox, D., Burgard, W., Dellaert, F., & Thrun, S. (1999). Monte carlo localization: Efficient position estimation for mobile robots. *AAAI/IAAI, 1999*(343-349), 2-2.

[23] Thrun, S., Fox, D., Burgard, W., & Dellaert, F. (2001). Robust Monte Carlo localization for mobile robots. *Artificial intelligence*, 128(1-2), 99-141.