# Target-Driven Navigation of ROS Robot with Object Detection using Deep Learning

**Athira T[1], Rajasree R[2]**

[1]*Student, Department of Computer Science & Engineering, College of Engineering Trivandrum*
[2]*Assistant Professor, Department of Computer Science & Engineering, College of Engineering Trivandrum*

---------------------------------------------------------------------***---------------------------------------------------------------------

**Abstract -** Target-driven navigation of robot has the task of navigating in an environment to reach a target specified by the user. Mapping, localization and planning are the major challenges in accomplishing the task of target driven robots. To overcome these challenges, this paper proposes a system integrated with Hector SLAM, Adaptive Monte Carlo localization (AMCL), A* and Dynamic Window Approach (DWA). In order to make robot able to navigate through complex environments, surrounding map has to be created. Simultaneous Localization and Mapping (SLAM) is a widely used technique for creating map of an unknown environment and localizing the robot at the same time. Once a map is created, the problem of localization of robots in the map arises. This issue of localization is solved in this system by integrating AMCL. The map created by SLAM algorithm is utilized by path planning algorithm to reach the final coordinates specified by the user. The proposed system uses A* algorithm for global path planning. But global path planning algorithm alone cannot handle new or dynamic obstacles in the path of navigation of robots. To deal with such obstacles, a local path planning algorithm called Dynamic Window Approach is also used in this system. In applications such as indoor service robots, rescue robots etc., the use of object detection algorithms enhance the performance of robots in fulfilling the task. A technique called ensemble method, which combines the results of several models is utilized in this system for object detection and identification. The combined use of Hector SLAM, AMCL, A* and DWA along with ensemble object detection algorithm improves the performance of target-driven navigation of robots. This paper studies the simulation of the proposed system in an open-source framework called Robot Operating system (ROS).

***Key Words*:** SLAM, AMCL method, A* algorithm, DWA algorithm, Ensemble method

## 1. INTRODUCTION

Target-driven navigation of robots have gained more attention in recent years. It is essential for many applications such as indoor rescue robots, home service robots, people assistance robots etc. The three basic components in autonomous navigation of a target driven robot in an indoor environment are mapping of environment, robot localization, and path planning. The integration of these components for a robot is a challenging task. Mobile robots with SLAM technology for creating maps of unknown environments have achieved much success in many scenarios. Many researches have been carried out to study path planning for autonomous navigation of robots with map constructed using SLAM technology. Zhang et al.[1] studied different Lidar-based SLAM mapping and path planning for rescue robots in an indoor environment. Recently many studies were done by combining SLAM and AMCL (Adaptive Monte Carlo Localization)[2] algorithm for autonomous navigation and localization of robot . In [3], a method is proposed to enhance SLAM algorithm by combining with Convolutional Neural Networks (CNNs) for an autonomous home service robots. The integration of SLAMs with object detection algorithm [4,5,6,7] proved to have enhanced the performance of autonomous robots in various applications. This paper discusses the effectiveness in fulfilling the task of target-driven navigation of ROS (Robot Operating System) robot incorporated with Hector SLAM mapping, AMCL localization, A* path planning and Dynamic Window Approach (DWA) technique. Object detection is another important area of research in the field of robotics in recent years. Many deep learning object detection models are now available. This paper also studies the effect of combining some existing object detection models to improve the accuracy of object detection. This technique of combining models called ensemble algorithm [8] is used in the proposed system to detect and identify the objects in the path of navigation of target-driven ROS robots. This paper is organized as follows: Section II presents the flowchart of the proposed system; Section III briefly discusses the two commonly used Lidar-based SLAM algorithm; AMCL is briefly given in section IV; The path planning algorithms A* and DWA are explained in section V; Object detection algorithms used in ths proposed system are given in section VI; Section VII provides the simulation results; Section VIII gives the conclusion..

## 2. PROPOSED SYSTEM

The simulation part of the proposed system is studied in this paper. Simulation is done in Robot Operating System (ROS).The proposed system is composed of mapping, localization, path planning and object detection. Fig. 1 shows the basic flowchart of the proposed system. The inputs from different sensors such as Lidar, IMU are used by SLAM for simultaneous mapping and localization of the robot. The SLAM will construct the 2D map of the environment. The map once created will be used by the robot for autonomous navigation. For target-driven navigation of any robots, path planning and localization play a key role. Here, the target is the destination location where the robot has to reach finally. Once a target location is given by the user, the robot has to locate itself in the constructed map. The proposed system uses AMCL algorithm for localization. Then an optimal path is estimated from the current location to the target location. This is done using the combined effort of A* and DWA algorithm.

Objects in the path of navigation is also detected using a technique called ensembling, a method of combining different models for object detection.

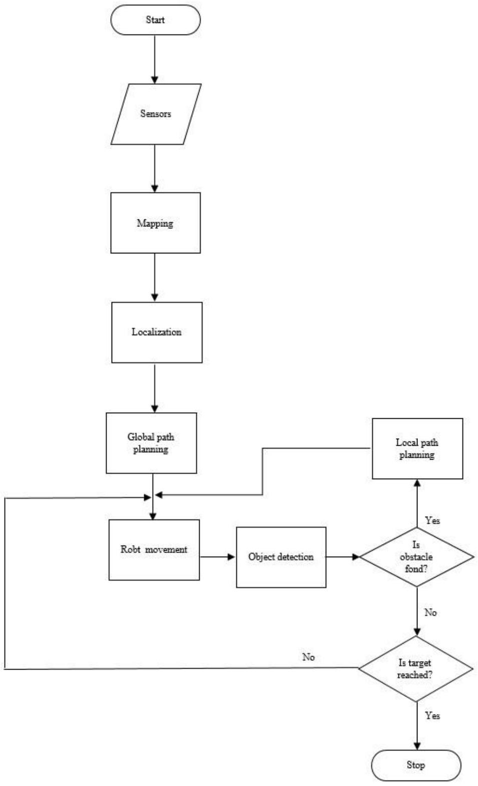


**Fig. 1**: Basic flowchart of the proposed system

## 3. SLAM ALGORITHMS

SLAM involves both localization and mapping by using data continuously measured from various onboard sensors. The data used in this system are Lidar scan and odometry. The result of SLAM is the 2D map of an environment. Today, different types of SLAM are available. But in this paper, we will focus only on Gmapping and Hector SLAM. These two SLAM algorithms are two widely used Lidar-based SLAM.

### 3.1 Gmapping

Gmapping is a laser-based SLAM algorithm for creating 2D occupancy maps of an environment. It is based on RaoBlackwellized particle filter algorithm. In this approach, each particle carries an individual map of the environment. A good mapping effect can be obtained with large number of particles. But, as the number of particle increases, the computational cost and execution time also increases. These particles are reduced by several adaptive techniques to learn the 2D grid maps. The most common technique in particle filter is Sampling Importance Resampling (SIR). As shown in Fig. 2, Gmapping provides grid mapping and position of robot simultaneously. The most widely used open source SlAM software package in ROS is gmapping. as Lidar, IMU are used by SLAM for simultaneous mapping.

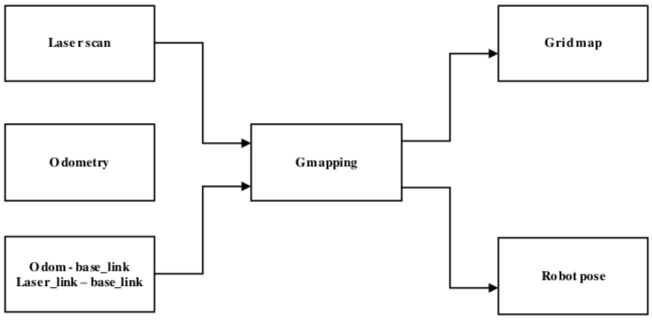


**Fig. 2**: Block diagram of Gmapping

### 3.2 Hector SLAM

Hector SLAM[9] is also a laser-based SLAM algorithm which can localize and construct 2D map of an environment. The advantage of hector SLAM over Gmapping is that the computational cost is smaller as it creates the map of an environment without odometry. This algorithm is based on scan matching technique. In scan matching technique, scan matcher searches for landmarks within a laser scan. Once the landmarks are identified it is used to determine the Lidar's position. Hector mapping is the open source SLAM software package in ROs for implementing hector SLAM. The block diagram representation of hector SLAM algorithm is shown in Fig. 3.

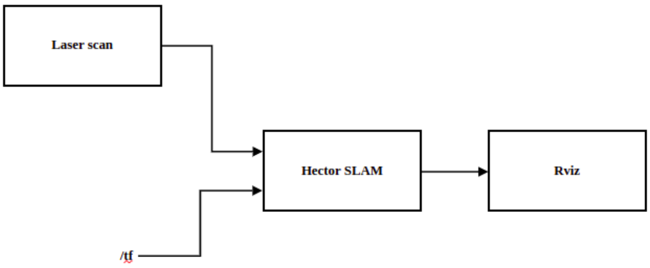


**Fig. 3**: Block diagram of Hector SLAM

## 4. AMCL ALGORITHM

After constructing a 2D of the environment, the robot should localize itself on it. AMCL is an algorithm that uses particle filter method to keep track of the pose of robot with respect to the generated map. Given a map of the environment, AMCL algorithm determines the position and orientation of a robot moving through the environment. The algorithm starts with uniform distribution of particles over the environment. When robot takes a step, it predicts the new pose of the robot. Based on recursive Bayesian estimation, the particles are resampled. Finally, the particles converge with the actual pose of the robot. AMCL algorithm is a variant of Monte Carlo Localization algorithm (MCL). In this work, AMCL algorithm is used for localization of robot as it converges faster than MCL. This algorithm is implemented in ROS using amcl package.

## 5. PATH PLANNING ALGORITHM

In three main challenges of path planning algorithm are:
- To plan a path to reach the target location
- To avoid obstacles in the path of navigation
- To find an optimum path

To implement path planning, a global path planner and local path planner are considered in this paper. Global path planning algorithms are used for planning a shortest path whereas local path planning are used mainly for obstacle avoidance. The proposed system uses A* algorithm as global path planner and DWA as local path planner. A* algorithm finds a shortest path from the current location of robot to the target location with the help of the constructed map. A* algorithm takes into account only the objects in the environment at the time of mapping. This algorithm selects a path having minimum cost function. The cost function of A* algorithm is calculated as below.

$$f(n) = g(n) + h(n) \tag{1}$$

where f(n) represents the total cost function of current node n, g(n) represents the actual cost from the starting node to the current node, and h(n) denotes the estimated cost of the current node to the destination node. The local path planner, DWA method is a velocity based technique. This algorithm gives the robot an optimal collisionfree velocity to reach the target. The two important phases of this algorithm are velocity search space calculation and select an optimal velocity. At first, a search space is constructed from the set of velocities that allows the robot to stop before colliding. Finally, an optimum velocity is determined to maximize robot's clearance, velocity and also to obtain the heading closest to the goal.

## 6. OBJECT DETECTION ALGORITHM

The information given by the camera used in the proposed robotic system is used by object detection algorithms for identifying objects. Currently there are many successful deep learning models for object detection. The two basic classifications in object detection algorithms are one stage detector and two stage detector. In one stage detectors such as Single Shot Detector (SSD) and YOLO (You Only Look Once), the input image is divided into regions and given to CNNs to obtain the detection. But in two stage detectors like R-CNN, Faster R-CNN, Mask R-CNN etc. features are extracted from input image and region of interests (ROI) for the detection of objects. The proposed system uses some models in TensorFlow 2 Detection Model Zoo[10] which is pre-trained on COCO 2017 dataset for object detection. In this paper, EfficientNet, centerNet, SSD MobileNnet and SSD ResNet models are compared based on their ability of detecting objects. Then two models having better object detction capability are selected and combined. The proposed system uses a combination of models called ensemble models to detect objects. In this technique, same object may be detected by more than one model. This may lead to multiple bounding boxes around the same object. To avoid these redundant boxes Non Maximum Suppression (NMS) technique is also used. The combination of different models is expected to improve the performance of object detection.

## 7. SIMULATION EXPERIMENTAL RESULTS

### 7.1 SLAM simulation

A simulation environment shown in Fig. 4 is first created in Gazebo platform to test SLAM algorithms. Different SLAM algorithms are available. The selection of SLAM depends on the purpose of the map. For our purpose, we decide to test Gmapping SLAM and Hector SLAM algorithms. Fig. 5 shows the scanning of environment and mapping in Rviz using Gmapping. The final 2D grid map of the environment created using Gmapping is shown in FIg. 6.

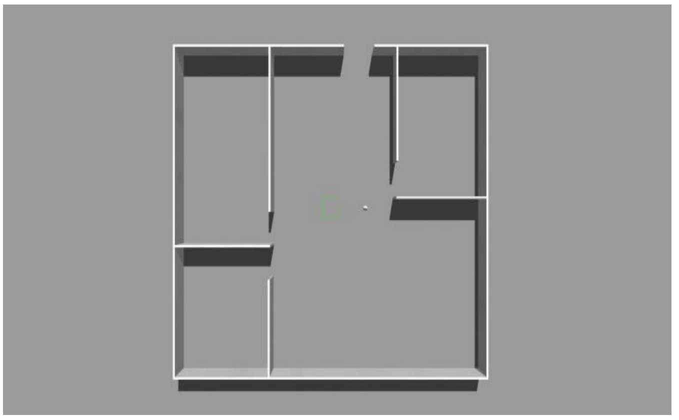

**Fig. 4**: Simulated environment in Gazebo

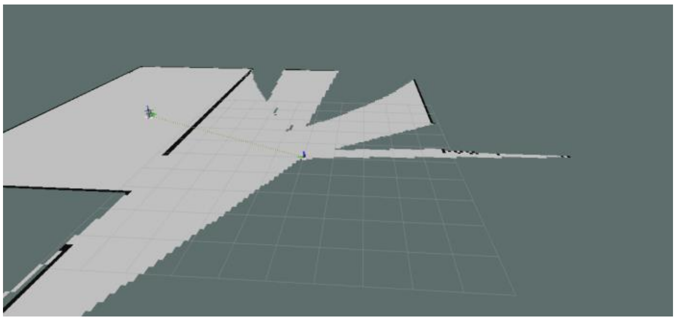

**Fig. 5**: Mapping of simulated environment using Gmapping

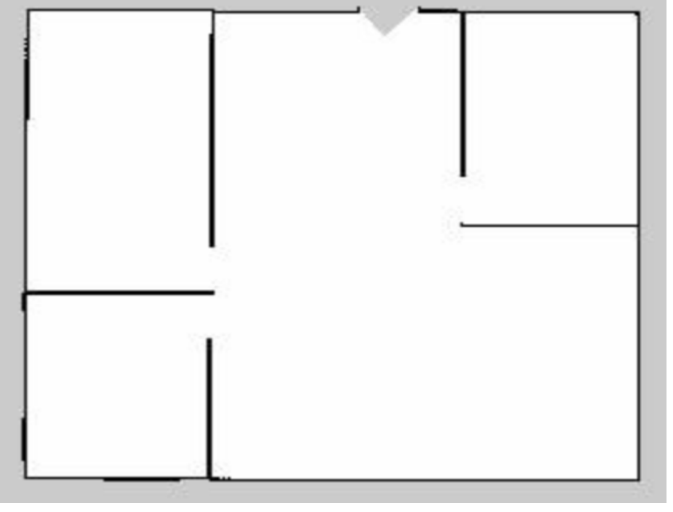

**Fig. 6**: Final 2D map obtained using Gmapping

In Fig. 7, the robot simulation process using Hector SLAM is shown. Fig. 8 shows the complete 2D map constructed using Hector SLAM.

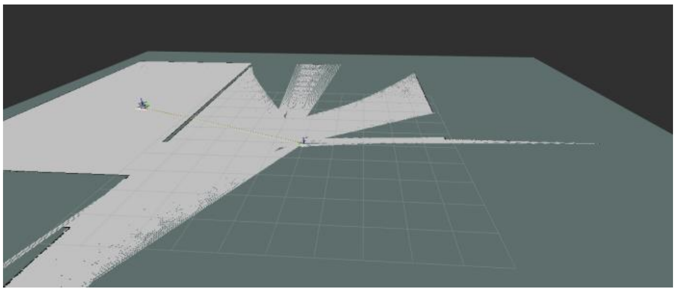

**Fig. 7**: Mapping of simulated environment using Hector SLAM

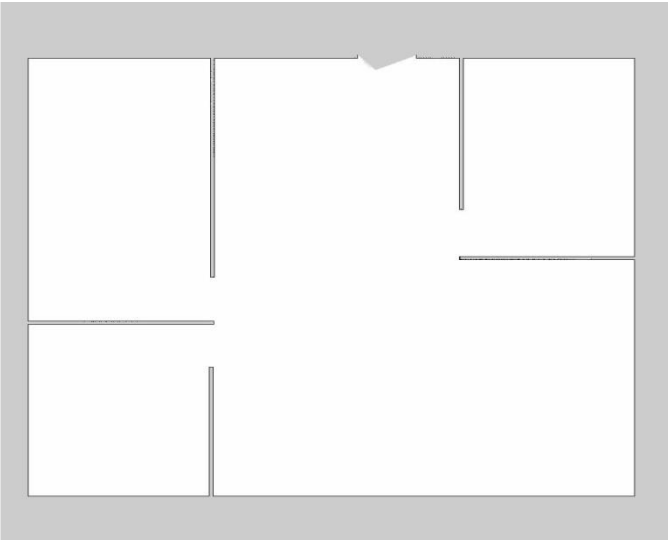

**Fig. 8**: Final 2D map obtained using Hector SLAM

The results of mapping clearly show that a perfect map of the simulated environment is obtained using Hector SLAM technique. So in this work, Hector SLAM is considered for target driven navigation of robot. An experiment with ten trials is conducted to evaluate the performance of Gmapping and Hector SLAM. In this experiment, the robot has to reach the target location given by the user. The time taken for each trial is calculated for Gmapping and Hector SLAM. The result of this experiment is shown in table-1.

### 7.2 Localization

Localization of robots in a known map is essential for the smooth autonomous navigation of robots. This paper focuses on the implementation of AMCL algorithm which is based on particle filter method. Using this algorithm the pose of robot is estimated within the created map. Fig. 9 shows the implementation of AMCL algorithm to estimate the position and orientation of robot in the map created using Hector SLAM. The green arrows around the robot represent the particles carrying information about the pose of the robot.

**Table-1:** Comparison of gmapping and hector slam

| **Trial No.** | **Gmapping** (time in secomds) | **Hector SLAM** (time in seconds) |
|---|---|---|
| 1 | 105 | 85 |
| 2 | 122 | 105 |
| 3 | 110 | 125 |
| 4 | 131 | 122 |
| 5 | 116 | 90 |
| 6 | 112 | 92 |
| 7 | 90 | 78 |
| 8 | 102 | 92 |
| 9 | 108 | 82 |
| 10 | 90 | 98 |
| Average | 108.6 | 96.9 |

### 7.3 Path Planning

The task of target-driven robot is to navigate in an environment to reach the target specified by the user. The purpose of the proposed system is to navigate autonomously and reach a target location given by the user. Given a map of the environment, the navigation of robot from current position to the goal location requires a proper path planning algorithm. In this work, A* algorithm is used for global path planning. This algorithm gives the shortest path from the current location of robot to the goal location based on the constructed map. Fig. 10 shows the target location (pink arrow). The path given by A* algorithm from the current location to target location is shown in Fig. 11.

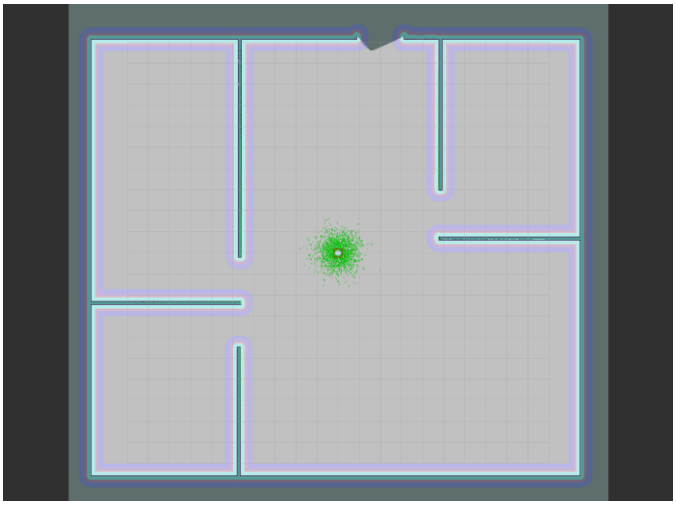

**Fig. 9**: Implementation of AMCL visualization in Rviz

The results show that a shortest path to reach the target location is obtained by using A* algorithm. But if a new or dynamic obstacle appears in the path of navigation of robot, then A* will fail to guide the robot to the destination. This issue is solved by incorporating DWA algorithm, a local path planning algorithm for finding a path to avoid the new obstacle coming to the navigation of robot. Fig. 12 shows an block placed as an obstacle in the environment which was never seen by the robot during mapping. Fig. 13 shows how the robot avoids the new obstacle placed in its path of navigation using DWA algorithm.

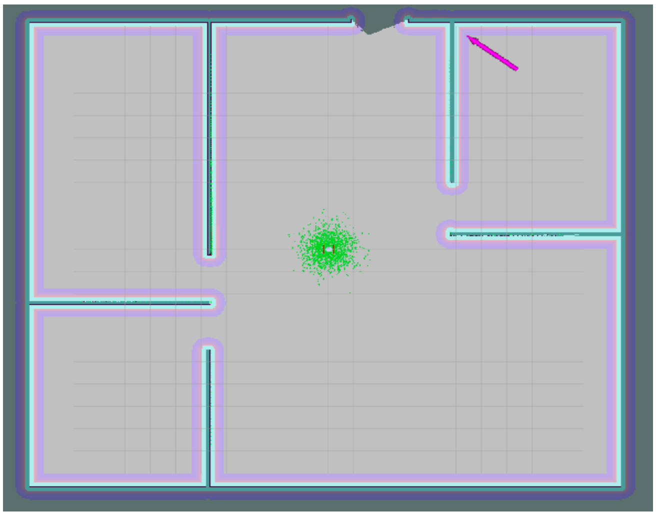

**Fig. 10**: Target location given by the user

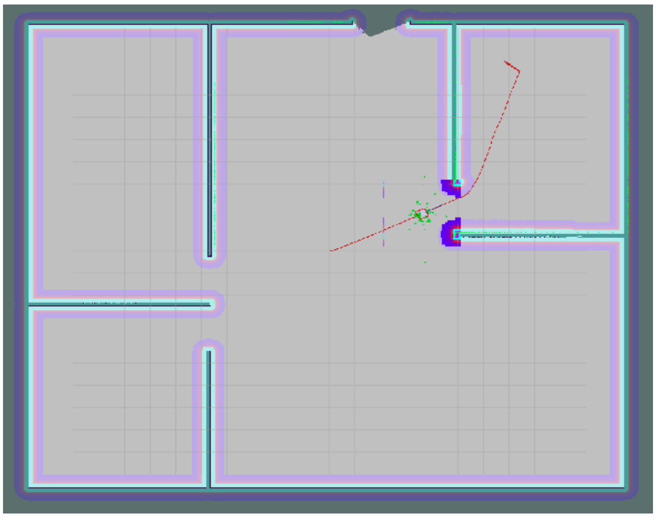

**Fig. 11**: Path planned by A* algorithm to reach the target location

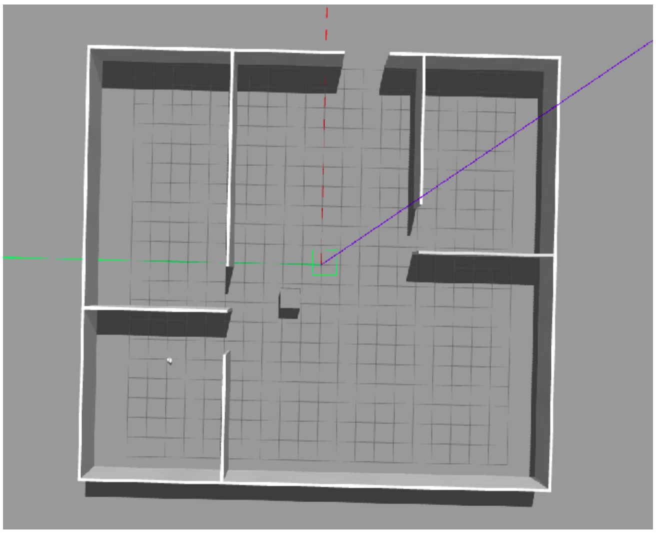

**Fig. 12**: A new obstacle placed in the simulated environment

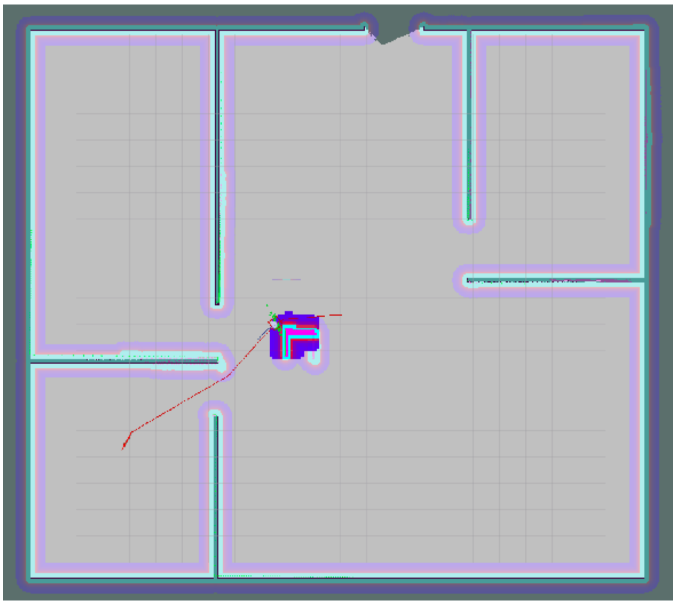

**Fig. 13**: Navigation of robot by avoiding a new obstacle using DWA algorithm

## 8. OBJECT DETECTION

The proposed system is also capable of detecting objects in the path of navigation. In this paper, four models COCO 2017 dataset from TensorFlow 2 Detection Model Zoo pretrained on COCO 2017 dataset were selected for study. Table-2 shows the selected model with their speed and mean average precision (mAP).These models were analyzed based on the ability of detecting multiple objects in a given input image and the confidences scores.

**Table-2:** Comparison of TensorFlow 2 detection model

| Model Name | Speed(ms) | COCO mAP |
|---|---|---|
| SSD MobileNet v2 | 19 | 20.2 |
| CenterNet Resnet50 V2 | 27 | 29.5 |
| EfficientDet D0 | 39 | 33.6 |
| SSD ResNet152 V1 FPN | 111 | 39.6 |


**Fig. 14**: Output of SSD MobileNet

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 06 | June - 2022 Impact Factor: 7.185 ISSN: 2582-3930

Fig. 14-17 shows the outputs of SSD MobileNet, CenterNet, EfficientNet, SSD ResNet respectively for a given input image.The results suggest that EfficientDet and SSD ResNet detected many number of objects. When the outputs of EfficientNet and SSD ResNet are analyzed, it is clear that all objects are not detected. Also models have different confidence scores for the same detected objects. But the combination of results these two models give a better result. So in this paper, an ensembling model is implemented by combining EffiecientNet and SSD ResNet. This gives a better performance as shown in Fig. 18.

**Fig. 15**: Output of CenterNet

**Fig. 16**: Output of EfficientDet

**Fig. 17**: Output of SSD ResNet

**Fig. 18**: Output of ensemble method

Now the ensemble model which combines EffiecientNet and SSD ResNet are added with the proposed system to identify the objects. For this, some objects were placed in the simulated environment and a target is specified by the user. While navigation through the planned path, the camera in the robot captures real time frames and is given to the ensemble model. Fig. 19 shows the detection made by ensemble model in a frame captured during the navigation of robot towards the target location.

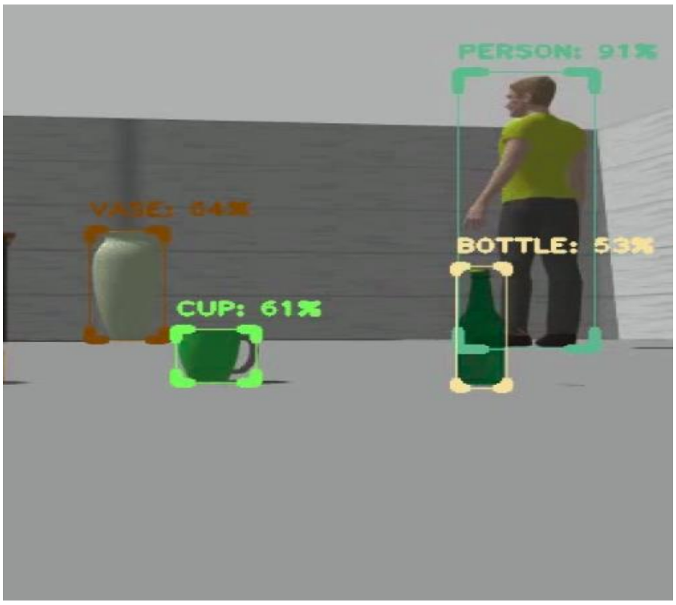


**Fig. 19**: Object detection using ensemble model during the navigation of robot toward goal location

## 9. CONCLUSION

In this paper, we introduced a system that integrates Hector SLAM, AMCL, A*, DWA algorithm and object detection capability using ensemble algorithm for a target-driven ROS robot. The simulation of this system in ROS gave a good performance result. This system succeeded in finding an optimal path and reached the target by avoiding all obstacles. Also, the use of ensembling technique improved the object detection capability of robots. In the future, this system can be modified and extended to develop indoor service robots.

## REFERENCES

[1] Xuexi Zhang, Jiajun Lai, Dongliang Xu, Huaijun Li and Minyue Fu, ”2d Lidar-Based Slam and Path Planning for Indoor Rescue using Mobile Robots”, *Journal of Advanced Transportation*, 2020.

[2] Zhang, Baoxian, Jun Liu and Haoyao Chen. “AMCL based map fusion for multi-robot SLAM with heterogenous sensors” IEEE *International Conference on Information and Automation (ICIA)*,2013, 822-827. in factory environment”, *Procedia Manuf.*, 2019, 38, 1524–1531.

[3] P. Maolanon, K. Sukvichai, N. Chayopitak and A. Takahashi, ”Indoor Room Identify and Mapping with Virtual based SLAM using Furnitures and Household Objects Relationship based on CNNs,” 2019 *10th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*, 2019, pp. 1-6, doi: 10.1109/ICTEmSys.2019.869596

[4] R. Raveendran, S. Ariram, A. Tikanmaki and J. R ¨ oning, ”De- ¨ velopment of task-oriented ROS-based Autonomous UGV with 3D Object Detection,” 2020 IEEE International Conference on Realtime Computing and Robotics (RCAR), 2020, pp. 427-432, doi: 10.1109/RCAR49640.2020.9303034. *Journal of Advanced Transportation*, 2020.

[5] Harapanahalli S, Mahony N.O, Hernandez G.V, Campbell S, Riordan and Walsh J, ”Autonomous Navigation of mobile robots in factory environment*”, Procedia Manuf.*, 2019, 38, 1524–1531.

[6] P. Jensfelt, S. Ekvall, D. Kragic, and D. Aarno, “Integrating slam andobject detection for service robot tasks,” in *IROS 2005 Workshop on Mobile Manipulators: Basic Techniques, New Trends and Applications* Edmonton, Canada, 2005.

[7] M. Kulkarni, P. Junare, M. Deshmukh and P. P. Rege, ”Visual SLAM Combined with Object Detection for Autonomous Indoor Navigation Using Kinect V2 and ROS,” 2021 IEEE *6th International Conference on Computing, Communication and Automation (ICCCA),* 2021, pp. 478- 482, doi: 10.1109/ICCCA52192.2021.9666426.

[8] Casado-Garcıa and J. Heras, ”Ensemble methods for object detection”, *Proceedings of the ECAI European Conference on Artificial Intelligence*, pp. 2688-2695, 2020.

[9] Stefan Kohlbrecher, Oskar von Stryk, Johannes Meyer and Uwe Klingauf, ”A flexible and scalable SLAM system with full 3D motion estimation [J]”, 2011 IEEE *International Symposium on Safety Security and Rescue Robotics*, December 2011.

[10] Hongkun Yu, Chen Chen, Xianzhi Du, Yeqing Li, Abdullah Rashwan, Le Hou, Pengchong Jin, Fan Yang, Frederick Liu, Jaeyoun Kim, and Jing Li (2020)TensorFlow Model Garden [Source code] https://github.com/tensorflow/models