Robotics Algorithms in Action with Gazebo and a Simulated TurtleBot

İsmail Alperen AKAR – Umut KORKMAZ

Supervisor
Dr. Berkan DÜLEK
Electrical and Electronics Engineering, Hacettepe University

Introduction

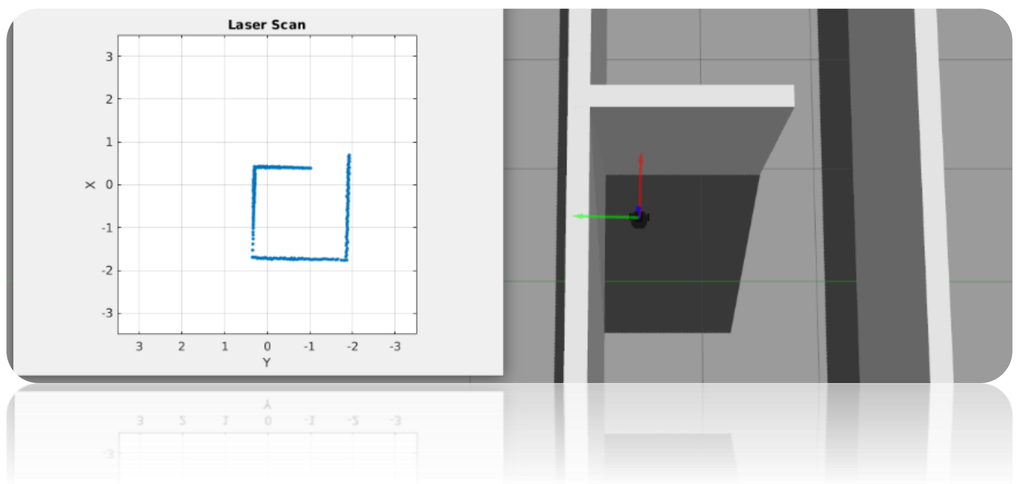
- In this project, obstacle avoidance, decision make on crossroads, hallway correction and SLAM algorithms have been created and tested in complete virtual environment Gazebo with realistic scenarios. ROS toolbox provides an interface connecting MATLAB with the Robotic Operating System. Enabling you to create a network of ROS nodes.
- This project's algorithms created in Matlab, simulated on Gazebo, used ROS for robotic actions and all these works were done on the Linux operating system.

Solution Methodology

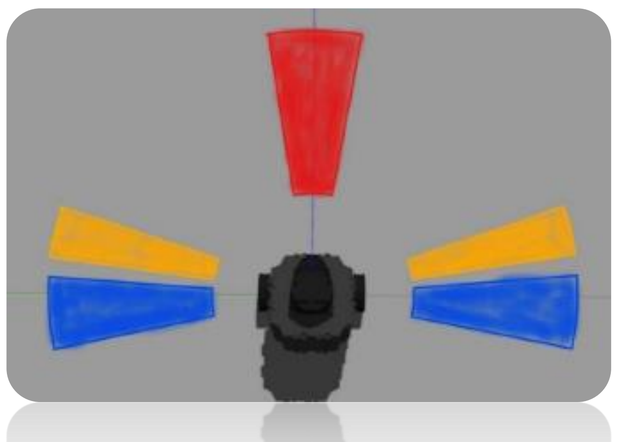
❖ The Project has two main objectives. First is to have an Obstacle Avoidance as movement algorithm and seconds is performing SLAM.

❑*OBSTACLE AVOIDANCE*

❖ Obstacle avoidance algorithm based on 2-D LIDAR.



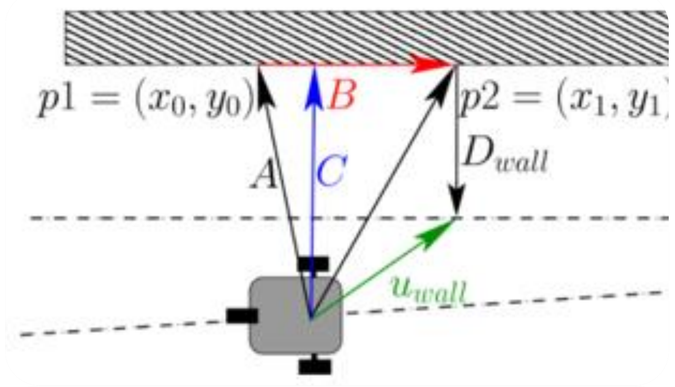
❖ Lidar data was segmented according to distances and angles. The TurtleBot moves according to this data segments.



❖ Made following improvements so that the algorithm can make more logical decisions for certain situations.

➢ *Going Straight in Hallway*

❖ To make TurtleBot go straight in hallway, we made an error calculation with logarithmic function for going straight.

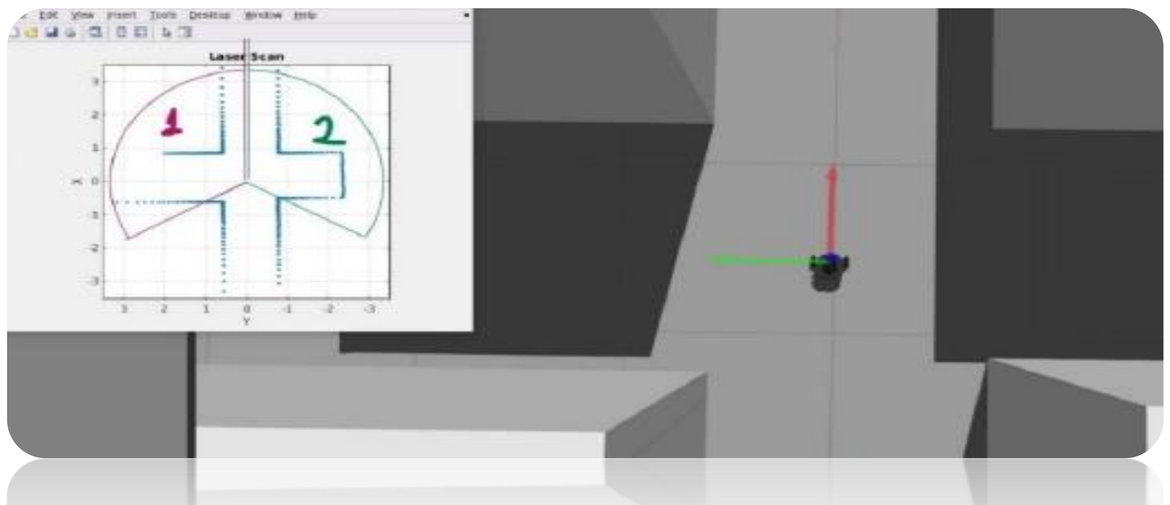


$$velMsg.Angular.Z = 0.01 * log2(abs(20 * diff + 1))$$

❖ After TurtleBot3 understands that if it is in hallway, checks LIDAR data to find closest point and if closest point's index greater than 180 means referring wall is the right side and makes index 90 with given formula. If index lower than 180, it will make index value 90. With this calculation TurtleBot3 avoids zig zag and goes straight in hallway.

➢ *Decision on Crossroads*

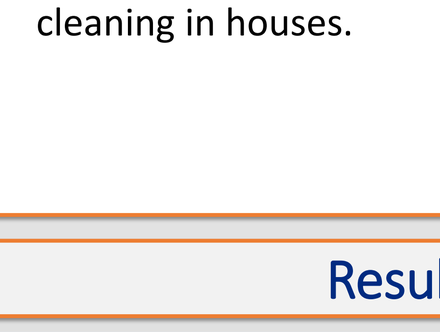
❖ In crossroads TurtleBot3, even there is an available path on front of it, stops and chooses random path. Thus, TurtleBot3 avoids missing alternative paths and creates a possibility to go different locations.



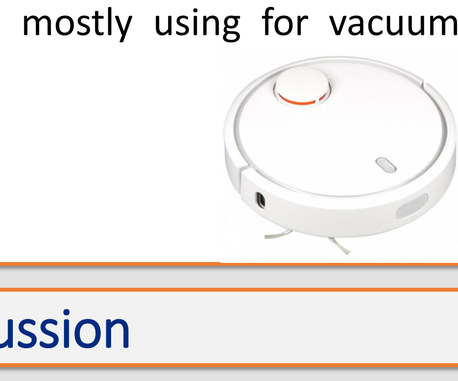
❑*Simultaneous Localization and Mapping (SLAM)*

❖ SLAM is computational problem of building a map of an unknown environment while keeping the track of mobile robot at the same time.
❖ SLAM consists of multiple parts; Landmark extraction, data association, state estimation, state update and landmark update. Slam is applicable for both 2-D and 3-D motion.
❖ Extended Kalman Filter is used for map building.
❖ Our algorithm basically scans as it goes forward, but when it starts mapping, the robot will stop. After mapping is done, robot will start going forward and continues to scan.

$$p(x_t, m \mid z_t, u_t)$$



Where,
x_t is the pose at time t,
$\boldsymbol{m}$ is the map,
Z_t is the measurements,
U_t is the controls.

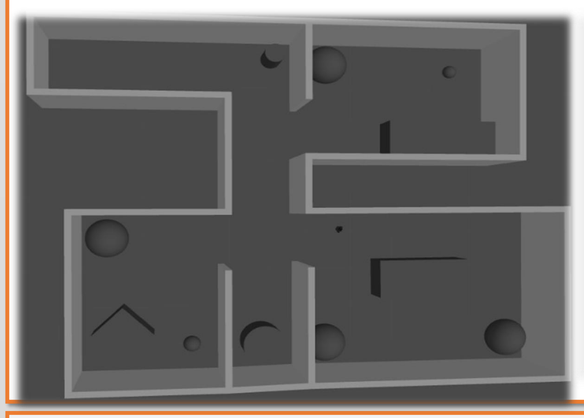


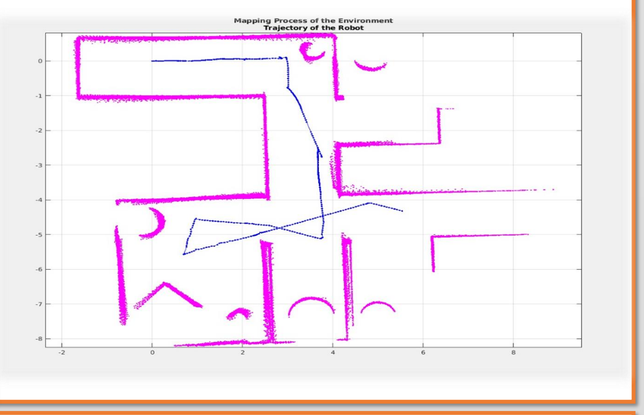
Application Areas

❖ These kinds of algorithms with LIDAR mostly using for vacuum cleaning in houses.

Results and Discussion

❖ As a result, TurtleBot3 successfully performed SLAM without any collusion.
❖ SLAM map is not built as same as environment but with small dispersions due to movement of TurtleBot3.





Acknowledgements

❖ This project was completed within the context of ELE401-402 Graduation Project courses in Hacettepe University, Faculty of Engineering, Department of Electrical and Electronics Engineering.
❖ We thank Dr. Berkan DÜLEK for his invaluable contributions to our project.