Title: Autonomous Trash collection and Disposal Robot Simulated in ROS/Gazebo Environment.

Name: Dipesh Budhathoki

Abstract:

The concept of smart homes is gradually becoming a reality and the need of service robots for domestic purposes is increasing with it. The goal of this project is to create an autonomous robot that can collect trash from a certain area of a household and place it in the dumpster. The robot then makes its way back to the starting point and does its chores as scheduled. One of the schedules cycles is demonstrated in this project. To successfully navigate, avoid obstacles, and accomplish its goal, the robot used a hybrid control system that processed the sensor data from laser, IMU, and odometry. Also in this project, the contrast is shown in the navigation of the robot with and without the path planning algorithm. This article details the construction of such a robot within the ROS/Gazebo simulation environment.

Introduction:

Automation of household tasks improves sanitation and efficiency, especially about waste management. The concept presents a robotic system to handle the chore of moving water from an indoor area to a disposal location on its own. This project contributes to the rapidly expanding field of home automation systems and falls under the umbrella of improving residential living through robotics.

The ROS-Gazebo-simulated robot combines a few sensors for obstacle avoidance and navigation. The system has the potential to be used in the real world because it is made to be reliable, effective, and able ot self-navigate in a changing environment. This project develops the fundamentals ideas of path planning and robotics to create an autonomous robot that can pick up trash by navigating a residential area. Finding the best route from the starting point to a goal while avoiding obstacles is known as path planning, and it is an essential component of robotics. The robot must perform this difficult task in real time, making judgements based on pre-programmed algorithms and sensor inputs. The procedure comprises a detailed examination of the surrounding conditions, the forecasting of dynamic shifts, and the development of an adaptable navigation plan.

To navigate the robot, our method combines reactive strategies and deterministic algorithms. The deterministic part entails placing waypoints along a predefined track to direct the robot toward the dumpster and then the trash pickup station. When the robot comes across unanticipated barriers, the reactive component kicks in, utilizing sensor data to dynamically modify its path. This approach works best when waypoints specify the overall course and local modification are done in response to urgent environmental inputs. But in our project, we have intensely utilized the reactive strategies and only fixed the goal point and the starting point. We tackle the more general problem of robotics autonomy—that is, a machine's capacity to carry out activities without the assistance of a human – by creating this robot. For a robot to be considered autonomous, it must be able to move around, make decisions. Adapt to new circumstances and draw lessons from its past. These skills are essential in a home environment because it is less controlled and more variable than in an industrial setting.

The methods for the robot's path planning and navigation, the incorporation of sensors for environmental interaction, and the specifics of the implementation that lead to the task of collecting rubbish autonomously will be covered in the following sections.

Approach:

The robots control system is built upon a PID like control approach with sensor data combined with potential field path planning algorithms. The attractive potential field guides the robot towards the goal while the repulsive field prevents the robot from colliding with obstacles. The hybrid control employed in the robot is described briefly below:

- **Attractive force:** The robot is attracted towards the waypoints using a force that is proportional to the distance from the target, transitioning to a conical potential beyond a threshold distance. A robot performs has better performance if we define the waypoints throughout the path but for this project, we wanted to exploit the dependency on different sensors to find its optimal path in real time. Thus, only the starting point and the goal point was defined.

$$F_{\mathrm{att}}(q) = -\nabla U_{\mathrm{att}}(q) = \begin{cases} -\zeta(q - q_{\mathrm{final}}) & : \quad \rho_f(q) \le d \\ -\dfrac{d\zeta(q - q_{\mathrm{final}})}{\rho_f(q)} & : \quad \rho_f(q) > d \end{cases}$$

Figure 1: Equation for attractive field.

- **Repulsive Force:** A repulsive force is proportional to the inverse square of the distance from the nearest obstacle and becomes active within a certain range. For this project, we have focused on the dynamic detection of the sensor rather than defining the obstacle.

$$F_{\mathrm{rep}}(q) = \begin{cases} \eta\left(\dfrac{1}{\rho(q)} - \dfrac{1}{\rho_0}\right)\dfrac{1}{\rho^2(q)}\nabla\rho(q) & : \quad \rho(q) \le \rho_0 \\ 0 & : \quad \rho(q) > \rho_0 \end{cases}$$

Figure 2: Equation for the repulsive force.

- **Laser Sensor**: It is used for real-time obstacle detection, and it provides the distance measurement necessary for the repulsive force calculation. The laser sensor we used in the ROS/Gazebo environment is a Hokoyu sensor.
- **IMU**: It provides orientation data to maintain the robots heading and helps in trajectory correction.
- **Odometry**: It delivers accurate positional information for navigation towards goal and ensures the robot returns to its starting position.

For this project, our main goal was to drive our robot from a predefined point near the household to the location of the dumpster which is located outside the premise of the household. We first tried to use the data from the laser sensor and odometry for achieving our goal by using a simple obstacle avoidance algorithm that takes a reactive approach. It uses laser scan data primarily to drive a robot between a starting point and a goal location while avoiding obstacles. Following initialization, odometry and IMU callbacks are used to continuously update the robot's state, which includes its location and orientation. The robot moves towards what is initially designated as the goal. The robot executed an avoidance maneuver, rotated away from the object depending on the observed angle, and moved ahead at a decreased speed to maintain a safe distance when laser scans identify an obstacle within a predefined threshold. Depending on which side of the robot the obstacle is on, the avoidance angle is calculated in a binary manner, turning the robot left or right. The algorithm for this process is as follows:

1. Initialize:
   a. Set Start_Position with x,y coordinates of the start
   b. Set Goal_Position with x,y coordinates of the goal
   c. Set Target to Goal_Position
   d. Set At_Goal to False
   e. Set Obstacle_Threshold to the minimum distance for obstacle avoidance
   f. Set Safe_Distance to the minimum safe distance to maintain from obstacles
2. Start Robot navigation loop:
   While True:
   a. Update Position and Yaw from odometry
   b. Update Nearest_Obstacle_Distance from laser scan

   i. If the distance to Target < goal threshold:
      A. If At_Goal is false:
         1. Set Target to Start_Position
         2. Set At_goal to true
      B. Else:
         1. Stop the robot
         2. Signal shutdown and exit the loop
   ii. Else if Nearest_Obstacle_Distance < Obstacle_Thresshold:
      A. Calculate the angle to the nearest obstacle Angle_To_Obstacle
      B. Determine Avoidance_Angle based on Angle_To_Obstacle
      C. Set angular velocity for avoidance maneuver
      D. Reduce linear velocity to Safe_Speed
   iii. Else:
      A. Calculate the heading to Target as Heading_To_Target
      B. Compute heading error as Heading_Error
      C. Adjust angular velocity based on Heading_Error
      D. Set linear velocity to normal cruising speed

   c. Publish linear and angular velocities for robot motion

3. End loop

The robot did a good job on a comparatively easier world setup, but it struggled while a more complicated setup was brought in. On our second approach, we employed path planning algorithm, potential field, integrated with the sensor data to achieve our goal. This method steers the robot by applying a combination of repulsive and attractive forces that are obtained from real-time sensor data. It moves toward an objective by determining an attractive force based on how far away the target is from it. At the same time, when impediments are found within a predetermined range, as indicated by laser data, it generates a repulsive force. By employing this tactic, the robot may instantly modify its course and avoid obstacles without relying on intricate path planning algorithms. This time the robot did a really good job avoiding the obstacle and reaching its goal and successfully coming back to its starting point. The entire code is submitted with this report. The motion of the robot would have been a lot smoother if we had employed the PID controller in it, but we wanted to apply a different approach as taught in the class, so we just used path planning algorithm and sensor data. The plots and results of our experiment are discussed below.

Results:

When the robot was deployed in the Gazebo world, it was able to find its way from the start point to the dumpster, the trash collecting station, and back again. The robot's ability to avoid obstructions by a wide margin proved how successful the sensor fusion plan and the control system were. Plots of trajectory tracking, and the yaw angle was generated, demonstrating the robot's autonomous ability to complete the objective.

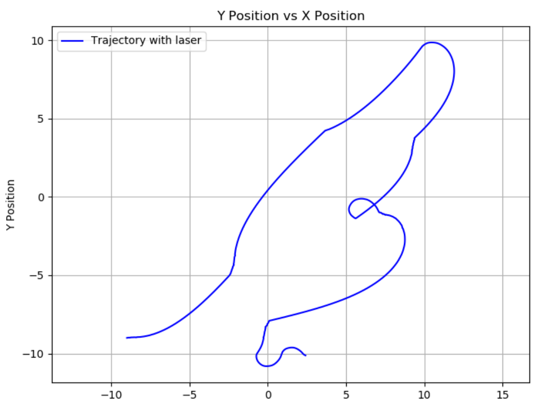

Fig 3: Trajectory on final world without path planning.

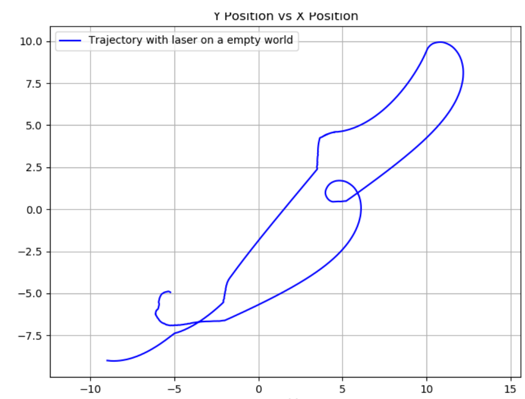

Fig 4: Trajectory on empty world without path planning.

In figure 3, we can see the trajectory of the robot running in the final world, with household and real obstacles, and in figure 4 we can see the trajectory of the same robot in an empty world with random obstacles. We can see that the robot did a good job reaching the goal but while returning to the starting position, it lost its way. We tried tuning the parameters, but the robot struggled every time. I think the robot could have done a better job if we employed a PID controller instead, but for this project, we limited our scope to using sensors and path planning algorithms only.

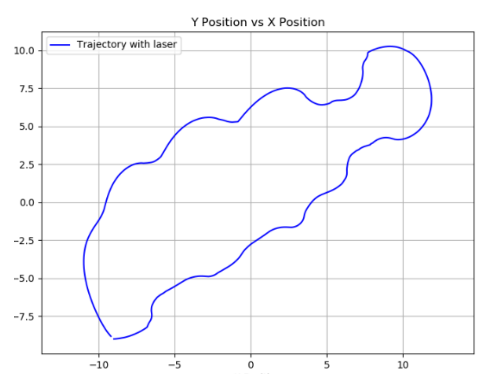

Fig 5: Trajectory on empty world with path planning.

Fig 6: Trajectory on empty world with path planning.

In figure 5, the robot is running in an empty world with randomly placed obstacles and in figure 5, the robot is running in the final world that we have designed with a household and real life obstacles. As we can see, the robot have done quite an impressive job avoiding the obstacles both while reaching the goal position and while returning back to the starting point. Also the motion of the robot is filled with less swift motions and the robot is able to use the shortest path avoiding the obstacles. The attractive force of the goal point is what driving the robot towards the dumpster and the repulsive force of the obstacles is repelling the robot away from the obstacles. It is apparent that implementing the pathplanning algorithm significantly improves the performance of the robot. This is also apparent by the plot of yaw angle with respect to time as below:

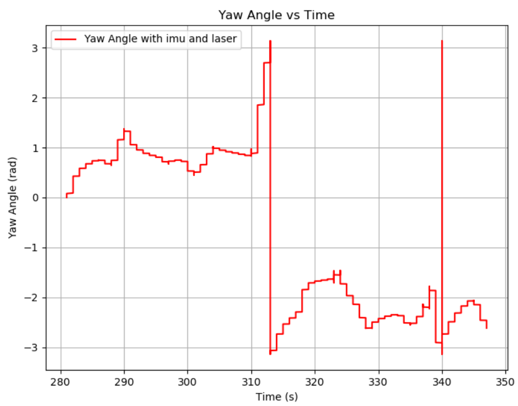

Fig 7: Yaw angle with path planning implemented.

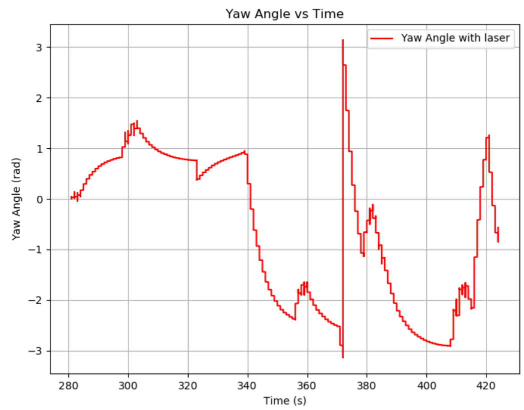

Fig 8: Yaw angle without path planning implemented.

As we can see in the plots, the motion of the robot is filled with lot less swift yaw angle changes while the pathplanning algorithm is implemented than while the robot is running depending on the laser sensor only. All the videos of the robot running in different environments is uploaded along side this report.

Conclusion:

The main objective of the priject, to automate the garbage pickup in a residential environment, was accomplished. Practical home robots must have the capability of retuning to the starting position upon task completion, which prepares the robot for repeated cycels. Even though the simulation results show the task was completed, there is still a lot of improvements. We could employ a PID controller to make the motion of the robot jerk free. The parameters of the controller can be further tuned to make the robot find its path in a more complicated environment. Potential future imprivements might be computer vision-based trash recognition, dynamic pathplanning, and integration with home automation systems for coordinated chores. The robot's ability to function in unstructured situations would be greatly enhanced by the addiotn of SLAM (Simultaneous Localization and Mapping). Deep learning algorithms may also be used for improved objct detection and decision-making.