Make Way Please

By Bonny Mahajan, Anna Wang

Abstract - If robots are to be expected to function in normal everyday environments, it is imperative for them to communicate with humans in order for them to complete their tasks successfully and more effectively. Navigation is one such task. As human beings we actively communicate while moving, to ensure fluid and hassle free movement. This should serve as model for our robots. Currently the robots in the BWI lab lacked the ability to communicate with people while navigating - if they were obstructing its path. Through our project we try to mitigate this problem by teaching the robot to tell people to 'Get out of the way' if they are obstructing its motion.

I. INTRODUCTION

As of right now, when the robots in the BWI Lab are moving to a specified goal and people get in the robot's path, the robot tries to replan its path and if it can't find another way - aborts its plan. We thought it would be useful if the robot could tell people that they were hindering it's movement - so that they could move out of the robot's path - and the robot could continue to it's navigation goal. We try to mitigate this problem in our project.

II. BACKGROUND/RELATED WORKS

Through our readings, we found out how different research groups were approaching this problem of successful robot navigation in crowds. In general a lot of researchers try to solve the problem of an obstacle being in the robot's path by talking about obstacle avoidance. Very few of them actually try to talk to the obstacle to get it out of the way. There, in our opinion, are situations when simply communicating with humans rather than avoiding them can be more useful in navigation.

Unlike our method- which relies on human cooperation, authors of "Human -friendly robot navigation in dynamic environments" [1] discuss a method for autonomous navigation that implements the same heuristics for mutual avoidance that human beings use when navigating. According to them because these paths were similar to how a human would move in a crowded room, these paths could be intuitively predicted by humans, hence leading to lesser confusion and more efficient navigation.

Another method presented by Borenstein, J., and Koren Y in "Real-Time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environments" [2] was using Vector Field Histogram (VFH). This method uses a two-dimensional Cartesian histogram grid as the world model. Similarly, to the cost map method, this world model is updated frequently in real time. However, unlike the cost map, the VFH method develops a one dimensional polar histograms around the robot's current location. This means the histograms will consist of obstacle densities and the robot will move towards the least dense direction. The difficulties of implementing this method would be determining the threshold needed before the robot will move to a certain location.

Similarly in 'UnfreeZing the Robot : Navigation in Dense, Interacting Crowds'[3] by Peter Tratman and Andreas Krause, they suggest the robots engage in mutual collision avoidance like humans do in crowded rooms.The authors point out how people engage in joint collision avoidance - by adapting trajectories that make room for the other to move.So the authors develop IGP (Interacting Gaussian Process) - which describes a probabilistic interaction between multiple navigating entities.Their tests show that IGP leads to shorter and safer paths than those taken by observed pedestrians.

In "Motion Planning in Dynamic Environments Using Velocity Obstacles"[4] , the authors Fiorini, Paolo, and Shiller, Zvi suggested using the concept of velocity obstacle for dealing with dynamic obstacles.For computing avoidance maneuvers, the robot computes a reachable avoidance set at discrete time intervals using a tree. The nodes correspond to positions of the robots and branches correspond to the avoidance maneuvers at those positions.

Again most of these research ideas try to find ways to avoid the 'dynamic obstacle' in the environment rather than interacting with it - which is the approach that we advocate.

II. TECHNICAL APPROACH

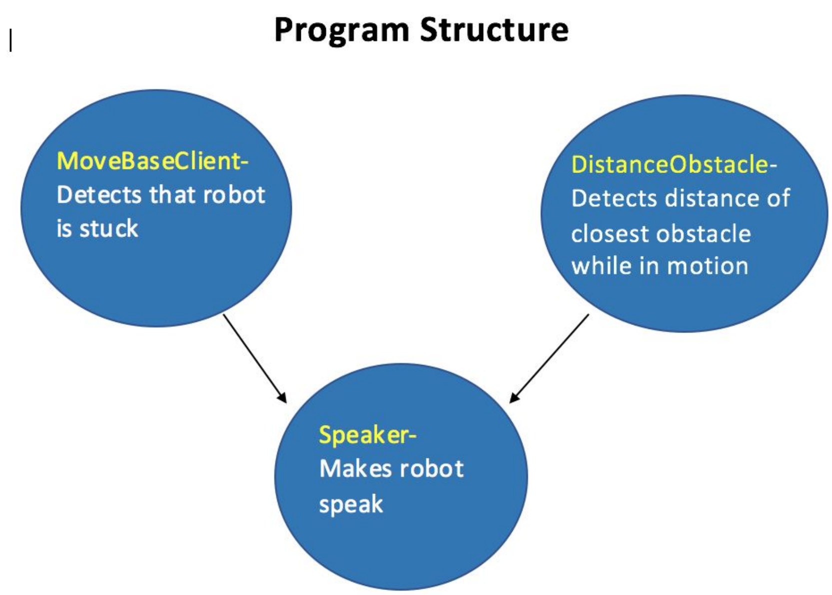

Figure A) Diagram representing the approach to the project. Program consists of three steps with MoveBaseClient, Speaker, and DistanceObstacle

Our project consisted of three main steps :

1. Detecting that the robot was stuck and then
2. Finding the closest obstacle and
3. If required conditions are satisfied, the robot will speak.

<u>Detecting when the robot was stuck :</u> We had a node called move_base_client that was publishing data to any of its subscribers about whether the robot was stuck or not.For detecting when the robot was stuck we used the following approach : Whenever the robot is given a navigation goal , the ROS Topic '/move_base/EBandPlannerROS/global_plan' is used to publish a plan for the robot to follow to get to it's destination. This plan is given by a vector of positions(given by x,y coordinates in the map) that the robot needs to be at in order for it to reach it's final goal. So we gained access to this plan by subscribing to this topic. With time we calculated the robot's distance from the next intermediate goal point in the plans vector we received above. If the distance increased or stopped decreasing for some reason - before the robot had reached it's final destination - indicated to us that the robot was stuck. The robot - when stuck - also often spins once when in recovery mode and trying to localize itself. We took this state of executing recovery behavior as another instance of when the robot was stuck.

<u>Using the robot's laser data to detect obstacles while moving</u> : We made another node called 'Distance Obstacle' that detected and reported information about the obstacles in front of the

robot. For this we subscribed to a topic called '/scan' that returns a vector containing data from the robot's laser readings. This data is actually the distances for each point in the array in front of the robot that the laser light hit an obstacle. We compared all these distances reported by the laser, to find the distance of the obstacle closest to the robot. If this min distance was in a given range then our obstacle could be a possible hindrance in our robot's way. So our node would publish that the robot is being blocked by something in it's path. This node, before publishing this data also ensured that the robot was moving or spinning while detecting minimum Obstacle distances as obstacle detection was only important when the robot was moving. For this we subscribed to a topic called '/cmd_vel' that reported the robot's moving speeds at the specified time interval.

<u>Making the robot speak</u> : So our main program subscribed to the 2 nodes :

1. Move_base_client [which detected if the robot was stuck]
2. DistanceObstacle [which detected if there was an obstacle within a given minimum distance from the robot's body while the robot was moving]

If either one of these publishers reported true, then our robot could tell whatever was obstructing its path to 'Get out of its way'. For making the robot speak, we used a ros node called sound_play .The SoundPlay topic had a C++ API which is what we used - to create a SoundClient() and to then use SoundClient's in-built function 'say' - which can be used to make the robot say something.

IV. RESULT/ EXPERIMENTS

These videos outline our results :
https://drive.google.com/file/d/0B2YKJdj5jKyicWNqU040X2ZPckE/view?usp=drivesdk
https://drive.google.com/file/d/0B2YKJdj5jKyibm9MN3RSN0tSSXc/view?usp=drivesdk
https://github.com/bmahajan/friSpring2017 **(Link to code)**
We tested our robot in both single and multiple person environments. It worked in both of the above. We also tested our robot with only inanimate obstacles like chairs, which is a shortcoming we talk about in the 'Possible Shortcomings' section below.

V. FURTHER APPLICATIONS

In events like exploreUT when the robot is in environments when it's path to it's destination could be easily obstructed ,telling the human being/beings that are around it , that they are in the way seems to encourage cooperation in the system - which is imperative in the future for robots that are aimed to function in human filled environments. We hope telling people to get out the way will clear the robot's path - hence allowing it to achieve any navigation goals that it may have lined up.

V. POSSIBLE SHORTCOMINGS

Although we got our our core idea to work, our approach does fail to address some problems.

First : Like in Paper Number [3] , our approach also assumes that the robot has the ability to control the crowd which it really does not. There is no way of ensuring that the person - when told to get out of the way will actually execute the action of 'getting out' and clearing out the robot's path.

Another possible shortcoming of the approach we adopted is our system's failure to differentiate between different kind of obstacles, specifically animate and inanimate objects. Because we rely solely on laser reading for getting information about obstacle distances, we don't characterize what the closest obstacle in our way actually is. It would only make sense for the robot to talk when the obstacle facing it is a person not when it is a chair. Due to time constraints, Our current model can't do that. But one possible solution to that is using OpenCV's person detector when working with closest obstacle finder. Though this proposes an entirely new set of challenges as openCV won't be as fast and could lead to a lag in the program. Hence making it unusable for real life purposes.

V. CONCLUSION

Overall we were successful in our achieving our goal of getting the robot to communicate with whatever was obstructing its path. This we feel is at times, a better approach to navigation than always simply relying on obstacle avoidance - which a lot of research seems to concentrate on.

REFERENCES

[1] Guzzi, Jerome, et al. “Human-friendly robot navigation in dynamic environments.” *Institute of Electrical and Electronics Engineers Explore* (2013): 1050-4729. http://ieeexplore.ieee.org/abstract/document/6630610/

[2] Borenstein, J., and Koren Y. “Real-Time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environments.” *Institute of Electrical and Electronics Engineers Explore* (1989): 1179-1189. http://ieeexplore.ieee.org/abstract/document/126042/

[3] Trautman, Peter, and Andreas Krause. “Unfreezing the Robot: Navigation in Dense, Interacting Crowds.” *Institute of Electrical and Electronics Engineers Explore* (2010): 797-803. http://ieeexplore.ieee.org/abstract/document/5654369/

[4] Fiorini, Paolo, and Shiller, Zvi. “Motion Planning in Dynamic Environments Using Velocity Obstacles” *The International Journal of Robotics Research* 17.7 (2016): 760 – 772. http://journals.sagepub.com/doi/pdf/10.1177/027836499801700706