Human Aware Robot Navigation

Aniket Patil
Robotics Engineering
Worcester Polytechnic Institute
Worcester, USA
apatil2@wpi.edu

Denny Boby
Robotics Engineering
Worcester Polytechnic Institute
Worcester, USA
dboby@wpi.edu

Nihal Suneel Navale
Robotics Engineering
Worcester Polytechnic Institute
Worcester, USA
nsuneelnavale@wpi.edu

***Abstract*—As robots grow their influence in our lives, it is important to develop methods which allow robots to work and move along-side humans more naturally without compromising our safety. In this paper, we work towards implementing motion planning techniques and their comparative analysis for human aware robot navigation. Our approach uses predicted human trajectories and a cost function to plan collision free paths while taking human/social comfort into account, our approach generates paths that is more like how a human would navigate, such as waiting for the other person to move, navigating along side another person, and making way when blocking the person's path. We will be conducting experiments for the robot navigation in hospital environment using Unity simulation and ROS.**

***Index Terms*—Human aware robot navigation, motion planning, Unity simulation, ROS**

GitHub: https://github.com/dennyboby/human_aware_robot_navigation

I. Introduction

Robot motion planning has been at the fore-front of robotics research for many decades, and over the past few years as robots start to make their way into our lives, human-robot interaction has become a growing research interest. Robots are no longer machines that are bound to a specific area or zone where humans couldn't collaborate efficiently, In recent years, robots work alongside humans, such examples would be Collaborative robots: robots working with humans in a shared environment, iRobot: a service based mobile robot which vacuums our homes and Warehouse robots: fleet of mobile robots which work in-sync, another important and growing research field is Autonomous Vehicles where a vehicle uses motion planning techniques to navigate complex real world scenarios.

Robot navigation can be defined as Where am I?, Where am I going? and How do I get there?. These questions are related to environment model, localization of the mobile robot and real-time path planning. Path-planning is finding a optimal or sub-optimal path from the start/current position to the target position, and this can be divided into global and local path planning. The global path planner generates a path off-line based upon a known map between the start and goal positions. In contrast, the local path planner generates a local path for obstacle avoidance and revises it in real-time.

In Section II, we look into the related work; Section III elaborates on our path planning approach, Section IV and Section V presents the simulation experiments and preliminary results; Section VI describes our project schedule and future work to be done.

II. Related Work

In the paper titled **Time Dependent Planning on a Layered Social Cost Map for Human-Aware Robot Navigation (2015)** by Marina Kollmitz, Kaijen Hsiao, Johannes Gaa and Wolfram Burgard, they proposed a novel planning based approach for social robot navigation. It uses predicted human trajectories and a social cost function to plan collision free paths that take human comfort into account.

Their approach generates paths that exhibit properties similar to those used in human-human interaction, such as waiting for a human to pass before continuing along an intended path, avoiding getting too close to another human's personal space, and moving out of the way when blocking a human's path.

The novel approach to human-aware navigation combines time dependent, search-based path planning with dynamic-social cost maps containing costs based on predicted human trajectories and Gaussian cost model. The A* search algorithm optimizes the costs along the planned path according to a cost function over navigation actions.

Experiments were conducted in real-life using Turtlebot2 platform to test navigation performance in path crossing situation, avoiding an approaching person and evaluated the navigation behaviour for a situation where the robot rests at a conflicting area.

In **2D Map Building and Path Planning Based on LiDAR (2017)** by Chi Zhang, unzheng Wang, Jing Li and Min Yan, they proposed a method of map building and path planning in an unknown environment. The method employs LiDAR to construct the 2D grid map, on which AD* and DWA (Dynamic Window Approach) are used for path planning. The global path planning is generated by AD* and DWA is responsible for dynamic obstacle avoidance.

For Global Map building they employ SLAM, and for localization they employ AMCL (Adaptive Monte Carlo Localization approach) to track robot's attitude in an indoor environment. For outdoor circumstance, the Global Positioning System (GPS) is exploited to provide an accurate position.

They have performed Simulations in ROS, Gazebo to validate the real-time robustness to plan an optimal path in a dynamic environment. For real-world Experiments they have used Voyage II robot with VLP-16 LiDAR, IMUs and GPS as experimental platform. They conducted two experiments, the first is an indoor environment where the robot is required to navigate through and office space with chairs being set up as static and dynamic obstacles. In the second experiment which is conducted outdoors, a red box is given as the goal and the bicycles and chairs are set up as obstacles, however due to the inaccuracy of localization, the robot can only reach the position near the target point.

In the paper **Human-aware Robot Navigation based in Time-dependent Social Interaction Spaces: a use case for assistive robotics** by Vega-Magro, Calderita, Bustos and Nunez, the service robot is set in a complex environment with elderly people, clinical staff or companions. This article presents a first approach to the idea of time-dependent social mapping, where the planning of route takes social variables such as time and schedule of certain spaces into consideration. This work uses a shared representation of the environment (DSR: Deep State Representation) and the CORTEX cognitive architecture. The social path is planned by using a classical Djisktra algorithm to find the shortest route between an initial position and a target to which the robot must travel. The social map is built by the social navigation agent according to the DSR, which has been previously provided by two different agents: the human-observer and the object recognition agent.

In order to evaluate the proposed navigation approach, they use the following evaluation metrics:

(i) average minimum distance to a human during navigation
(ii) distance traveled
(iii) navigation time
(iv) cumulative heading changes
(v) personal space intrusions

The first experiment evaluates the approach described in this paper for building time-dependent social interaction spaces (a room with a TV, circular table and a stretcher).

The second test represents a more complex scenario of different rooms, objects and people, and where the activity scheduling is also modified and the robot plans a path from its current pose to a pose in the other room.

In the paper **MRPB 1.0: A Unified Benchmark for the Evaluation of Mobile Robot Local Planning Approaches** by Jian Wen, Xuebo Zhang, Qingchen Bi, Zhangchao Pan, Yanghe Feng, Jing Yuan, and Yongchun Fang, they do a detailed comparison between DWA and TEB local planner for a differential drive robot. The robot they use a Pioneer 3-DX mobile robot. In this paper they setup following scenarios for testing:

1. Indoor scenarios
2. Narrow space scenarios
3. Partially unknown scenarios
4. Dynamic scenarios

They evaluate the the performance of TEB and DWA planner in the following aspects:

A. *Efficiency*

Based on evaluation they concluded that DWA needs to forward simulate and evaluate each pair of sampled velocities and does not take into account the environmental information during sampling. Therefore, considerable time is wasted in generating infeasible trajectories. While TEB takes the global path as the initial guess and obtains the local optimized trajectory through several iterations. As a result, planning with TEB is 8.94 times faster than planning with DWA.

B. *Safety*

Based on experimental results shown in paper TEB achieves better security performance in most cases because of the better clearance from obstacles but the total travel distance of TEB is relatively longer than that of DWA.

C. *Flexibility*

The evaluation results indicate that TEB performs better flexibility than DWA especially in scenarios like mazes that require robots to turn continuously.

III. PROPOSED METHOD

We are working with navigation of a mobile robot in an indoor environment, which combined with the presence of humans makes the environment very cluttered. Therefore, we employ various tools and algorithms to consider different aspects of indoor robot navigation. We need a local planner which can handle dynamic obstacles and needs separate distance thresholds for static and dynamic obstacles such as humans.

A map of the environment is needed as an input for trajectory planning in our hospital environment. This map can be obtained using slam_gmapping, which is a ROS package for Simultaneous Localization and Mapping (SLAM) of the environment. We first map and save the environment locally, which can be later included in the ROS map server.

After we include the map, we need a global and local planner for navigation. There are various local planner packages available for mobile robot navigation using ROS. In our implementation, we will use TEB local planner and the global_planner package of ROS, which internally uses Dijkstra and A* algorithms. This planner utilizes the Static Obstacle and Inflation Layer Costmap available in ROS. This entire architecture will act as the baseline for comparison with our implementation.

We added a Social Costmap Layer which will consider humans, their velocities and direction of motion in the environment during runtime. This costmap layer includes boundaries based on our specifications of what constitutes as personal space, social space and public space. Using this costmap, we integrated Time Based Lattice Planner (global planner) and Timed Path Follower (local planner) which provides better navigation around humans.

A. *Base Global Planner : Global Planner*

This planner is a package in ROS that uses Dijkstra algorithm to compute the goal from current robot pose to goal location. It uses all costmap layers, robot pose and goal pose to find the shortest path to goal. In figure 1 we can see the global path planned by Dijkstra algorithm(yellow in color).

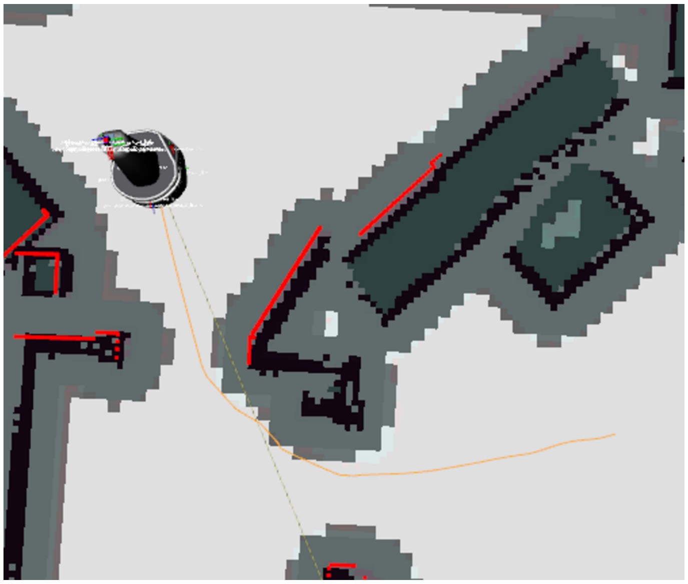

Fig. 1. Base Global Planner

B. TEB planner : Local Planner

The ”Timed Elastic Band” approach optimizes robot trajectories by subsequent modification of an initial trajectory generated by a global planner. In this trajectory optimization the objectives include but not limited to execution time, path length, separation from obstacles, passing through intermediate waypoints and compliance with robots dynamics, kinematic and geometric constraints. TEB explicitly considers the dynamic motion of obstacles and the robot such as velocities and accelerations. Simulations and experiments with a real robot show that this implementation is robust and computationally efficient. Figure 2 shows the TEB planner trajectories amongst three obstacles.

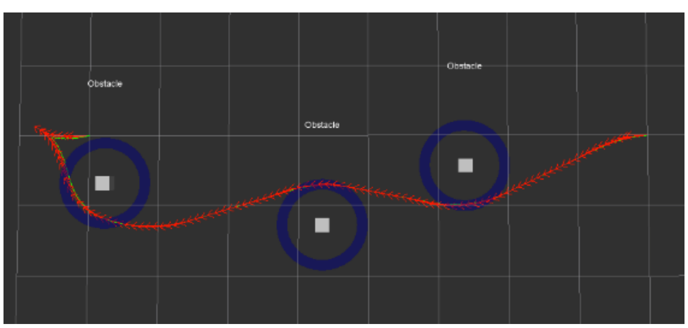

Fig. 2. TEB planner

In TEB we can optimize the following parameters for better path planning and obstacle avoidance. A section of parameters are shown in figure 3.

C. Time-based Lattice Planner : Global Planner

Lattice planner is a global planner where the planner will generate a path from the robot’s current position to a desired goal pose. The paths are generated by combining a series of

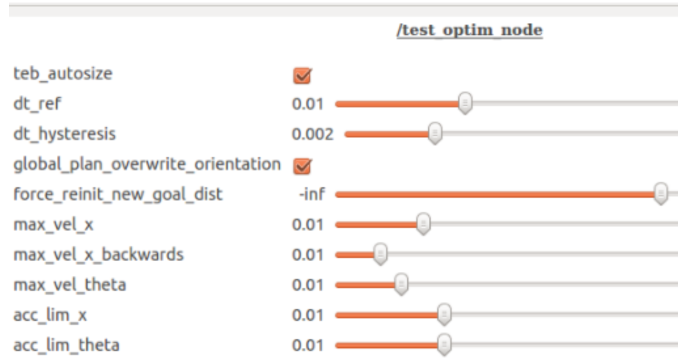

Fig. 3. TEB planner parameters

”motion primitives” (figure 4)which are short, kinematically feasible motions. Planning is done in x,y,theta dimensions, resulting in smooth paths that take the robot orientation into account.

In our project we use time dependent, deterministic A* to reason about the spatial relationship of the robot and humans with respect to time and by adding time to the planning space introduces additional complexity. The planned, time dependent trajectories have to be executable to ensure that the robot can follow the planned trajectory in time. The reachable robot configurations within a time interval depend upon current pose and velocity and its kinematic and dynamic constraints.

There are various parameters that we can tune to get better performance, some of the parameters are below:

allow_unknown: whether the planner is allowed to expand into unknown map regions

xy_goal_tolerance: the Euclidean goal tolerance distance in meters

yaw_goal_tolerance: the rotational goal tolerance in rad

time_resolution: time resolution of the discretized configuration space in seconds **collision_check_time_resolution**: time increment to slice trajectories for collision checking in seconds

time_steps_lookahead: maximum number of time steps for planning

planning_timeout: timeout for the planning after which the best solution found so far is returned

passive_navigation: flag for planning dynamically for the max number of time steps, not only until the goal is found

publish_expanded: flag for publishing the expanded search tree for visualization

D. Timed Path Follower : Local Planner

In our project we use the Timed Path Follower as our local planner, timed_path_follower package provides a local planner plugin for the ROS navigation stack. The local planner implements a controller for robots with differential drive constraints to follow time dependent, dynamically feasible navigation paths. It is based on a feedback motion controller for differential drive robots. Note that the local planner is only

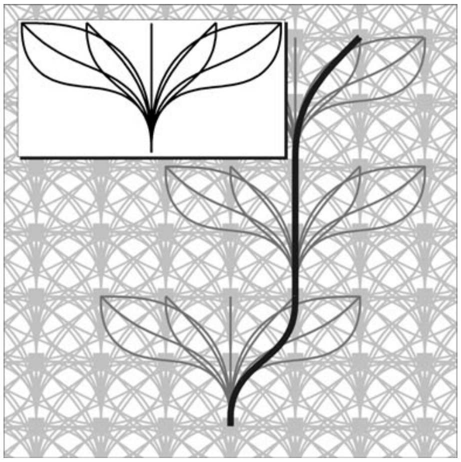

Fig. 4. Lattice Planner - motion primitives

concerned with following a given robot trajectory as closely as possible, it does not perform any collision checking.

The following parameters configure the behavior of the path follower:

allow_backwards: allows backwards motion when trying to follow a trajectory.

xy_goal_tolerance: euclidean goal distance threshold in meters.

yaw_goal_tolerance: rotational goal distance threshold in radians.

IV. Experiment

We conducted three experiments to demonstrate and evaluate our method of social robot navigation. Our experiments are conducted in the Unity simulator using a Gopher Presence robot set in a hospital environment provided by Hiro Lab, WPI.

We mapped the hospital environment using the LiDAR sensor present on the Gopher presence robot and slam_gmapping package. After our initial mapping, we get a map with non-existent walls, to resolve the issue of non-existent walls we edit the map in a GNU-Image Manipulation Software. We use this map for global path planning. The map is shown in figure 5.

Our first experiment tests robot navigation in a intersection space where humans are crossing the robot's path. This experiment is illustrated in figure 6. For this scenario we intend to make the robot wait till the human passes by before re-planning to move towards its goal.

The second experiment tests robot navigation in a corridor with humans walking parallel to and opposite to the robot's navigation direction. This experiment is illustrated in figure 7. For this scenario we intend to make the robot give enough space for the human to pass by and not make the human uncomfortable.

The third experiment demonstrates our robot's global path planning capabilities and compare TEB and Timed Path Followers ability to re-plan for dynamic-moving obstacles and

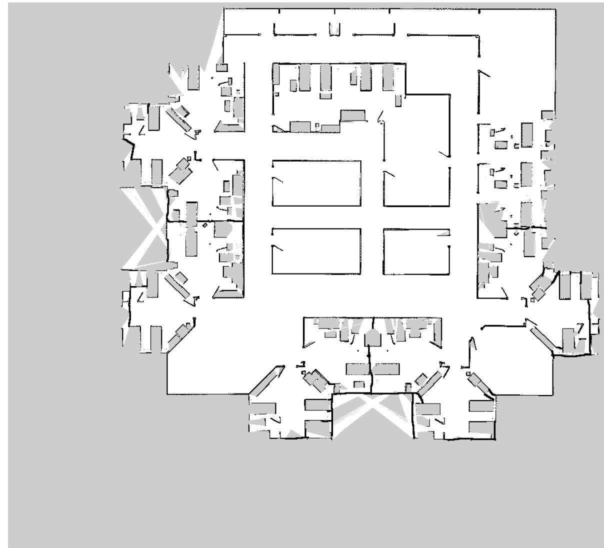

Fig. 5. Map of hospital environment

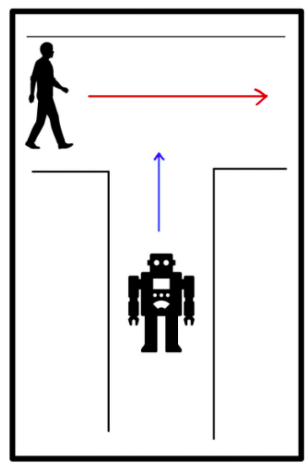

Fig. 6. Illustration of Experiment 1

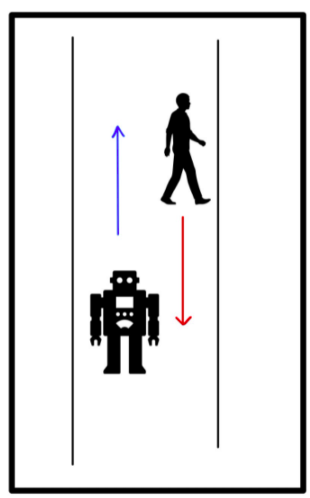

Fig. 7. Illustration of Experiment 2

new static obstacles in the environment. This experiment is illustrated in figure 8.

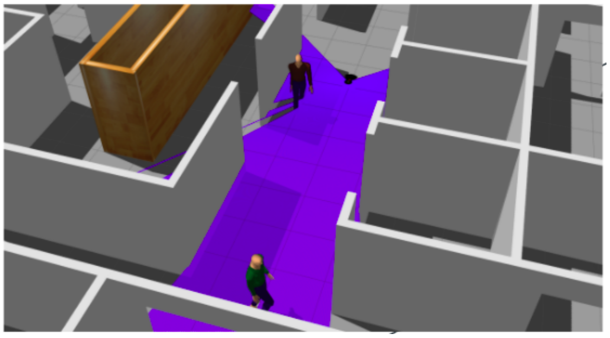

Fig. 8. Illustration of Experiment 3

To evaluate the performance of our human aware navigation we choose our evaluation metrics as follows:

1. Optimality of the global path planned.
2. Optimality of the local planner to plan a new path.
3. Obstacle avoidance.
4. Human-Social aspects.

V. Results

A. Workspace Setup

A repository on Github is setup called ”Human Aware Robot Navigation” which is present in the following link (https://github.com/dennyboby/human_aware_robot_navigation). Figure 9 shows the repository setup. Setting this up helps us to coordinate on the same project well, track changes and revert back to previous commit whenever something breaks. We also setup the hospital simulation in Unity with gopher presence robot with all the parameters of the plugins set according to our use case. Figure 10 shows the hospital scene in unity. As part of learning we understood GitHub workflow and command, the working of ROS, Unity and different plugins in Unity. And finally we understood how Unity communicates with ROS.

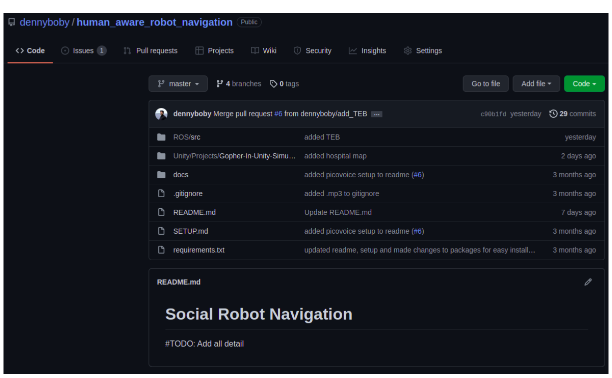

Fig. 9. Github Repository

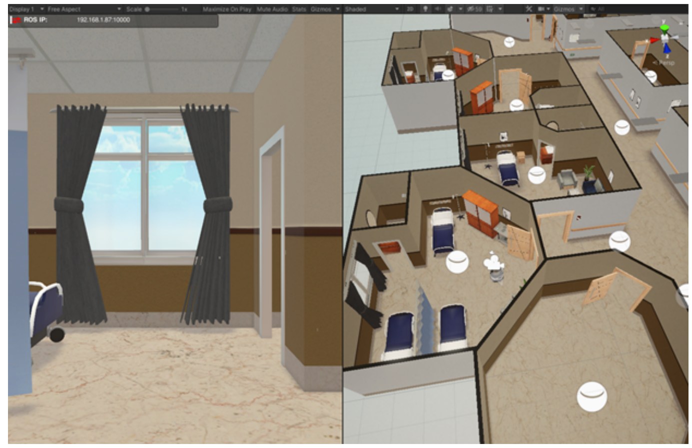

Fig. 10. Hospital scene in unity simulation

B. Mapping

After setting-up the simulation the next step is to map the entire hospital scene using slam_gmapping. In order to map the environment the slam_gmapping library needs transformation between odom frame and robot base_link and the laser scans as input and as output it generates the map as we move the robot in the environment. Finally once we have mapped the area we can save the map for loading it later for navigation. The mapped map is show in figure 5. As part of learning we understood the working of slam_gmapping and the different parameters that can be tuned for slam_gmapping.

C. Navigation

For navigation we have done literature review of papers that do a comparative analysis of DWA and TEB local planners performance for differential drive robot. Based on the review we have decided to use TEB local planner because the paths planned more flexible, safe and efficient. So we setup TEB local planner for dynamic obstacle avoidance and global_planner (A* planner) for global path planning as our baseline for comparison with our human aware robot navigation.

As mentioned before, In our project we employ a social navigation stack, by the means of this stack we intend to form a social cost map on the human and also on the predicted path. The global planner we use in this stack is a time-based Lattice Planner from the tb_lattice_planner package, and timed-path planner for local dynamic planning.

Figure 11 shows the planned global path towards goal (in yellow) and local path (in red) as it navigates around the obstacles by using the costmap layer (in light blue).

D. Experiment 1: Intersection

As discussed before, in this experiment we intend to employ TEB-local planner and the results are a follows, we can see that the TEB-local planner needs to be optimized for our application, after successfully optimizing the TEB-local planner, we can see the local path being planned in real-time in the Rviz window, and when a human is detected, a path is generated to avoid collision. In figure 12, we can see from

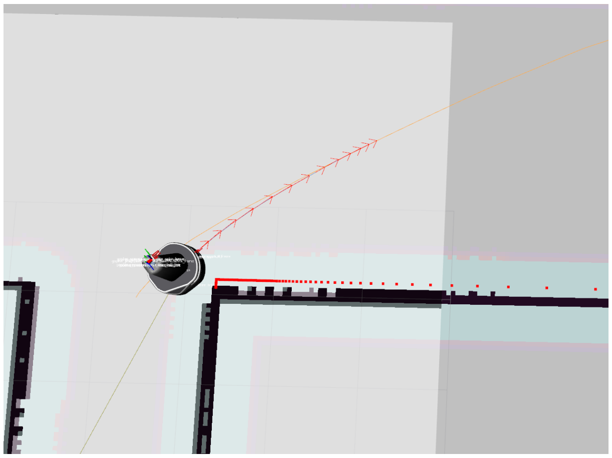

Fig. 11. Global (in yellow) and local (in red) path

the Rviz visualization that the human is detected and a path is generated to avoid them. The global path is shown in orange, while the local path is shown in blue.

This experiment fails when the human makes a sharp reverse and gets in way of the robot and the robot is unable to re-plan fast enough, it also fails when the human walks from the blind spot of the robot, here the blind spot maybe the sides and rear of the robot where the Lidar scanner is not integrated for.

Fig. 12. In the left is Rviz visualization, In the right is Unity simulation.

In figure 13, we can see the path planned by the Time-based Lattice planner and Timed Path Planner in orange, the social cost map layer is the blue cloud of points around the human(red points). When we use Time-based Lattice Planner and Timed Path Planner in this experiment, we can observe that the robot stops and waits for the human to pass by before resuming traversal on its path.

We have conducted 10 different test cases to compare the TEB planner and Timed Path Planner as shown in Table 1. From this table we can see that our approach performs far better than TEB planner, the number of collisions is far fewer compared to TEB and the path length and min. distance between the robot and human is maintained sufficiently.

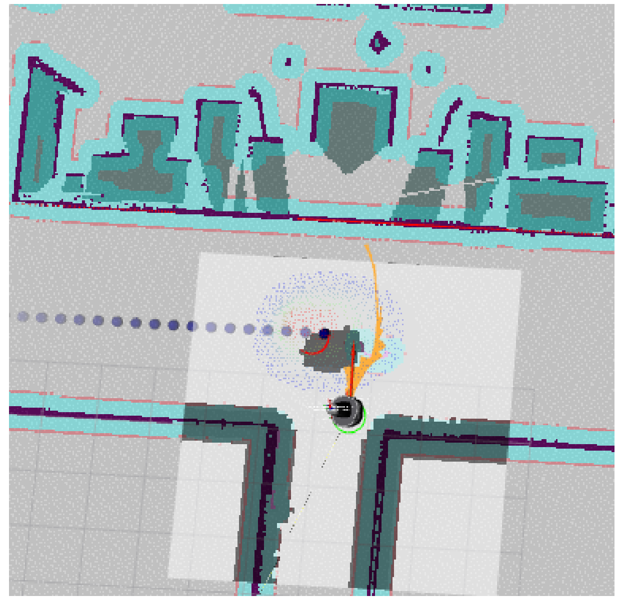

Fig. 13. Experiment 1 with Social Costmap and Timed Lattice Planner

E. Experiment 2: Corridor

To evaluate this experiment where the robot needs to navigate with respect to the human's movement, we make the robot move in direct path of the human as shown in figure 14. Here, we can see that the global path planned (orange) is in collision with the human but TEB-local planner path (blue) avoids collision with the human.

This experiment fails if the robot is too slow to avoid the human, or if the human's movement is faster than the local planner's ability to re-plan.

Fig. 14. In the left is Rviz visualization, In the right is Unity simulation.

In figure 15, we can see the robot replanning to avoid imminent collision with the human. Similar to experiment 1 we have conducted 10 different test cases to compare the TEB planner and Timed Path Planner as shown in Table 2. From this table we can see that our approach has fewer collisions and since timed path planner is used for social aspects to navigation, we see that the min. distance between the robot and human is higher than TEB, this results in a longer path taken by the robot but our approach is more social.

TABLE I
RESULTS OF EXPERIMENT 1

TEST CASE	TEB Planner			Timed Path Planner		
	Collision	**Path Length**	**Min. Dist to Human**	**Collision**	**Path Length**	**Min. Dist to Human**
1	**No**	4.942	1.233	**No**	4.858	1.916
2	**Collision**	5.019	0.634	**No**	4.892	1.244
3	**No**	5.079	1.241	**No**	4.720	1.382
4	**Collision**	5.074	0.684	**No**	4.879	1.217
5	**Collision**	5.180	0.587	**No**	4.695	1.470
6	**No**	4.957	1.545	**No**	4.922	1.331
7	**No ***	4.808	0.649	**No**	4.957	1.270
8	**No**	5.098	1.584	**No**	4.717	2.123
9	**Collision**	5.355	0.812	**No**	4.918	1.268
10	**Collision**	5.586	0.587	**No**	4.683	1.179
Average		5.109	0.955		**4.824**	**1.440**

TABLE II
RESULTS OF EXPERIMENT 2

TEST CASE	TEB Planner			Timed Path Planner		
	Collision	**Path Length**	**Min. Dist to Human**	**Collision**	**Path Length**	**Min. Dist to Human**
1	**No**	19.069	0.777	**No**	19.531	0.928
2	**No**	19.133	0.780	**No**	20.387	0.957
3	**Collision**	19.390	0.573	**Collision**	19.50*	0.90*
4	**No**	19.323	0.776	**No**	19.896	1.045
5	**No ***	19.099	0.634	**No**	19.387	0.872
6	**Collision**	19.231	0.646	**No**	19.405	0.886
7	**No**	19.036	0.724	**No**	19.175	0.828
8	**No**	19.391	0.834	**No**	19.548	0.717
9	**No**	19.190	0.719	**No**	19.492	0.874
10	**No**	19.305	0.616	**No**	20.345	1.114
Average		**19.216**	0.707		19.666	**0.912**

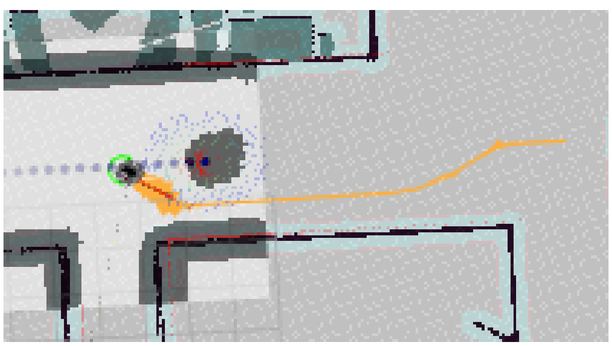

Fig. 15. Experiment 2 with Social Costmap and Timed Lattice Planner

F. Experiment 3: General navigation

To evaluate this experiment we make the robot navigate the environment to avoid collisions with the humans and new static obstacles such as chairs, IV stands, trashcans and more as shown in figure 16.

While conducting this experiment we came across one particular case where the robot collides with the IV stands and Lamps, upon further experimentation and analysis using Rviz, we found a shortcoming with the Lidar setup, the Lidar which is present in the robot is at a certain height and due to this height the Lidar maps the IV stand/Lamp's post as a pixel or point in Rviz and does not consider the width of the bases and the costmap generated is for the pixel and not the base, hence when navigating among such obstacles the robot collides.

Table 3 describes the 3 test cases where we compare the overall performance of TEB with Timed Path Planner, we make the robot navigate through 3 paths namely, a-b, b-c, c-d, and we can clearly see that our approach is better than TEB in collision avoidance and being more socially acceptable.

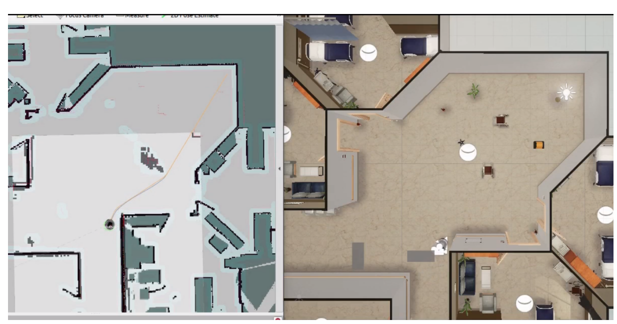

Fig. 16. In the left is Rviz visualization, In the right is Unity simulation.

TABLE III
RESULTS OF EXPERIMENT 3

TEST CASE	TEB Planner			Timed Path Planner		
	Collision	**Path Length**	**Min. Dist to Human**	**Collision**	**Path Length**	**Min. Dist to Human**
1						
a-b	**No**	17.217	1.073	**No**	16.128	0.800
b-c	**No**	11.527	2.558	**No**	11.737	1.69
c-d	**Collision**	20.555	0.560	**No**	20.528	0.795
2						
a-b	**No**	17.199	1.251	**No**	16.509	0.825
b-c	**No**	11.637	2.514	**Collision**	12.024	0.582
c-d	**No**	19.986	0.617	**No**	19.821	0.649
3						
a-b	**No**	17.258	3.120	**No**	16.423	0.688
b-c	**Collision**	**12.287**	0.584	**No**	12.770	0.994
c-d	**Collision**	19.302	0.578	**No**	20.042	0.877

VI. DISCUSSIONS AND CONCLUSION

By experimenting with TEB, we came across the following issues:

1. Blindspot: The robot is unable to navigate or avoid the human, if the human approaches the robot from its blindspot. To address this issue we can employ a 360 Lidar sensor to grasp the scene.

2. Optimization: If the parameters of the planners are not optimized properly, then the planner fails.

3. Velocity of the robot: After satisfactory optimization of the planners, the robot would still collide with the human if the robot doesn't have enough velocity and acceleration to traverse the non-collision path. We need to optimize the planner while considering the maximum velocity and acceleration of the robot, as these parameters are dependent on the hardware of the robot.

4. Sensors: As discussed in Experiment 3, the type of sensor and their position/orientation does contribute to the mapping and obstacle avoidance, one way of solving this issue is by using different/multiple sensors.

In our project, to employ the social cost-map on the nurse in Unity, we subscribe to the nurse's position and velocity published from Unity, however this would not be possible in a real-world application as we would be required to estimate them real-time, to estimate them we can employ concepts of computer vision and deep learning to develop a model which detects humans in the scene and estimate their position and velocities.

To address most of the issues faced with TEB local planner we employ the timed path follower to our robot. When we use the timed path follower, we can see that our robot waits for the human to pass and when the path is clear the robot resumes it's navigation, and by employing the time-based lattice planner we get smooth planned path, and the social cost-map helps the robot navigate better around the human, which results in better human aware navigation.

TABLE IV
TIMELINE & TASK DIVISION

Week Number	Task Division		
	Aniket	***Denny***	***Nihal***
Week 1-2	Literature Review		
Week 3-4	Unity Simulation Setup, Familiarize with ROS		
Week 5-6	Generate Map, Integrate Global and Local Planner for base comparison, Project Proposal Report and Slides		
Week 6-7	Implement lattice planner(global planner) for the generated map		
Week 8-9	Implement timed path follower(local planner) for dynamic obstacle avoidance and social navigation		
Week 10-11	Experiment 1	Experiment 2	Experiment 3
Week 12	Buffer Week		
Week 13	Achieve Extended Goals		
Week 14	Report, PPT and Presentation		

VII. SCHEDULE (TASK DIVISION & TIMELINE)

REFERENCES

[1] M. Kollmitz, K. Hsiao, J. Gaa and W. Burgard, "Time dependent planning on a layered social cost map for human-aware robot navigation," 2015 European Conference on Mobile Robots (ECMR), 2015, pp. 1-6.

[2] C. Zhang, J. Wang, J. Li and M. Yan, "2D Map Building and Path Planning Based on LiDAR," 2017 4th International Conference on Information Science and Control Engineering (ICISCE), 2017, pp. 783-787.

[3] Wen, J., Zhang, X., Bi, Q., Pan, Z., Feng, Y., Yuan, J. and Fang, Y. (2021). MRPB 1.0: A Unified Benchmark for the Evaluation of Mobile Robot Local Planning Approaches. 2021 IEEE International Conference on Robotics and Automation (ICRA), 8238-8244.

[4] A. Vega-Magro, L. V. Calderita, P. Bustos and P. Núñez, "Human-aware Robot Navigation based on Time-dependent Social Interaction Spaces: a use case for assistive robotics," 2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), 2020, pp. 140-145.

[5] A. Kushleyev and M. Likhachev, "Time-bounded lattice for efficient planning in dynamic environments," 2009 IEEE International Conference on Robotics and Automation, 2009, pp. 1662-1668, doi: 10.1109/ROBOT.2009.5152860.

[6] C. Rösmann, F. Hoffmann and T. Bertram: Integrated online trajectory planning and optimization in distinctive topologies, Robotics and Autonomous Systems, Vol. 88, 2017, pp. 142–153.

[7] C. Rösmann, F. Hoffmann and T. Bertram: Planning of Multiple Robot Trajectories in Distinctive Topologies, Proc. IEEE European Conference on Mobile Robots, UK, Lincoln, Sept. 2015