Proceedings of the 2018 IEEE
International Conference on Robotics and Biomimetics
December 12-15, 2018, Kuala Lumpur, Malaysia

On Safe Robot Navigation among Humans as Dynamic Obstacles in Unknown Indoor Environments

Alireza Hekmati and Kamal Gupta, *Member, IEEE*

***Abstract*— In this paper, we rigorously test two conjectures in mobile robot navigation among dynamic obstacles in unknown environments: i) a planner for static obstacles, if executed at a fast update rate (i.e., fast re-planning), might be quite effective in dealing with dynamic obstacles, and ii) existing implemented planners have been effective in humans environments (with humans being dynamic obstacles) primarily because humans themselves avoid the robot and if this were not the case, robot will run into collisions with humans much more frequently.**

The core planning approach used is a Global path planner combined with a local Dynamic Window planner with repeated re-planning (GDW). We compare two planners within this framework: i) all obstacles are treated as static (GDW-S) and ii) predicted trajectories of dynamic obstacles are used to avoid future collisions within a given planning horizon time (GDW-D). The effect of humans avoiding robot (and other humans) is simulated via a simple local potential field based approach. We indicate such environments by a suffix +R (repulsion) for the corresponding planner. Hence there are four categories that we tested: GDW-S, GDW-D, GDW-S+R and GDW-D+R in different environments of varying complexity. The performance metrics used were the percentage of successful runs without collisions and total number of collisions. The results indicate that i) GDW-D planner outperforms GDW-S planner, i.e., conjecture 1 is false, and ii) humans avoiding robots does result in more successful runs, i.e., conjecture ii) is true. Furthermore, we've implemented both GDW-S and GDW-D planners on a real system and report experimental results for single obstacle case.

I. INTRODUCTION

Path planning for navigating mobile robots is a well-researched area in robotics and significant algorithmic work has been done for global planning approaches for static environments [1], and for dynamic environments [2]-[4]. However, in reality, due to uncertainties and lack of exact and complete world models, mobile robots require a combination of global planning with local avoidance that interleaves planning, sensing and execution, and repeated re-planning dynamically in order to navigate to the goal.

For static unknown environments, [5] implemented what they call Global Dynamic Window (GDW) approach. It repeatedly creates a global plan (i.e., a global path to the goal), based on current sensed map (with unknown treated as free), via a navigation function [1], which is then fed to the local planner, where the Dynamic Window Approach (DWA) [6] is used to obtain the locally optimal trajectory. Dynamic obstacles were treated as instantaneously static in this work.

*Research supported by NSERC Discovery and RTI Grants to Kamal Gupta.

Alireza Hekmati (ahekmati@sfu.ca) and Kamal Gupta (kamal@sfu.ca) are with the School of Engineering Science, Simon Fraser University, Burnaby, BC, V5A 1S6, Canada.

Fraichard [7] emphasized the importance of considering dynamic obstacles' future behavior to avoid inevitable collision states (among other criteria) and conjectured that for robotic systems operating in human environments (he surveyed 3 such systems Minerva [8], Rhino [9] and Robox [10]), collisions are avoided primarily because "people took care of the collision avoidance". Trautman and Krause [11] referred to a related issue, they called it Freezing Robot Problem (FRP) i.e., in dense crowds of people, due to high obstacle density, and motion uncertainty of humans, conventional path planners often fail to find a collision free path. They solve the FRP by incorporating joint collision avoidance, where humans (or other agents) cooperatively make room to create feasible trajectories for the robot.

In this paper, we rigorously test two conjectures that are based on empirical observations mentioned in the above discussion. The first conjecture is if a planner for static obstacles, executed at a fast enough update rate, is as effective in dealing with dynamic obstacles as a planner that incorporates the predicted future motion of obstacles. While one could argue that indeed incorporating dynamic obstacles should do better, and if that turned out to be true, we wanted to quantify the improvement, i.e., how much better, as well. The second conjecture is if planners are effective in dynamic environments (with humans being dynamic obstacles) primarily because humans themselves avoid the robot and if this were not the case, robot will run into collisions with dynamic obstacles much more frequently.

For fair comparison, all algorithms needed to be implemented on the same platform, and we chose ROS (Robot Operating System) [12] for this purpose. The standard navigation stack within ROS implements an elaborate version of Global Dynamic Window (GDW) type planner mentioned above. It treats all obstacles as static (i.e., even the dynamic obstacles are treated as instantaneously static), and we call this overall planner GDW-S for our purposes. We then modified the ROS code to incorporate collision checks with respect to the dynamic objects predicted future trajectory within the DWA (similar to [13]) for a specified time horizon. We call this version GDW-D. For simulating the obstacles, we used Stage [14] to run our simulations. Two types of worlds with moving obstacles were simulated – in one world moving obstacles do not avoid the robot and in the other, moving obstacles do avoid the robot via a simple potential field approach (denoted with +R). We then ran simulations with both planners for each of the worlds, resulting in 4 different sets of simulations – GDW-S, GDW-D, GDW-S+R, and GDW-D+R.

Two performance metrics were used: i) percentage of successful runs, i.e., where a successful run means that the robot was able to reach its goal without any collisions with the obstacles, static as well as dynamic, and ii) total

978-1-7281-0377-8/18/$31.00 © 2018 IEEE

number of collisions, where we let the simulations run for a longer time (to the goal and back to the start) and count the total number of collisions robot had with the obstacles. A two-tailed Z-test with p- value of 0.05 was used to determine if the results were statistically significant [15].

Our results indicate that i) GDW-S planner running at a faster re-planning rate (4Hz) still significantly under performs the GDW-D planner running at a slower re-planning rate (2Hz), i.e., conjecture 1 is false, and ii) humans avoiding robots does result in more successful runs, i.e., conjecture ii) is true, however, the relative improvement in GDW-S+R (as compared to GDW-S) is substantial (and is statistically significant). This provides rigorous evidence to support Fraichard's observation that humans avoiding robot is a significant factor since even a poorly performing planner in dynamic environments such as GDW-S is improved significantly when the dynamic obstacles themselves avoid the robot. However, the story is somewhat different for the relative improvement in GDW-D+R (as compared to GDW-D). While there is improvement, the relative magnitude of improvement is much smaller and moreover, it was found not to be statistically significant. Overall, GDW-D+R planner had the highest total percentage of success and outperformed GDW-S+R in a statistically significant manner (68% as compared to 51%). A key contribution of this paper is the definitive answers to the two conjectures in the mobile navigation field based on rigorous statistical testing. Furthermore, although the two algorithms we implemented are existing algorithms, significant effort was put in to implement them on the same hardware and then testing the simulations under exactly same conditions including the robot and the environment.

The organization of the paper is as follows. In Section II, we reviewed other related work on motion planning with replanning. In Section III, we give a brief background on the two core sub-planners – GDW-S and GDW-D as well as the repulsion force model added to the obstacles. In Section IV, we explain simulation details and discuss the results. In Section V, we explain the experimental setup and present preliminary experimental results for single moving obstacle case. Finally, conclusion and future work are discussed in Section VI.

II. Other Related Work

We divide the related work on motion planning (with re-planning) in dynamic environments into two categories: i) algorithmic works verified via simulations and ii) works implemented on real systems. In addition, there is substantial work in the fields of human robot interaction (and social robotics) literature that deals with issues such as comfort level of humans in proximity of robots, etc. [16]-[18]. The main focus of our paper is on the motion planning aspects; the human interaction aspects can be encompassed via techniques such as social force models (a variant of potential field). In category i), Petti and Fraichard [19] explicitly consider dynamic obstacles and call the concept of repeated re-planning as Partial Motion Planning (PMP), in which an updated model of the future obstacle trajectories is acquired and forward simulated over a certain time horizon, then the state-time space of the robot is searched using RRT to select the best safe partial trajectory based on a given criterion. The planner iteratively operates until the robot reaches a neighborhood of the goal state. Recently, [20] extended RRT to RRT^X that incorporates quick replanning. In related works from our Lab, [21] proposed a safety hierarchy framework for planning and execution of trajectories of a mobile robot in unknown indoor dynamic environments. The planner takes into account both the moving obstacles' future behavior in a safety hierarchy and robot's own dynamic constraints. It also uses an appropriate time horizon to plan and respects the timing constraints on various modules of the planner. [22] Incorporated human avoidance of robot within the planning algorithm.

In the second category, Trautman and Krause [11] incorporated joint collision avoidance via interacting Gaussian processes, which essentially model humans (or other agents) cooperatively making room to create feasible trajectories for the robot. They used overhead camera sensors to detect humans and their methodology requires training data.

Shiomi et al. [16] use a pedestrian model, essentially a version of the popular Social Force Model (SFM) [23] combined with a Dynamic Window planner that incorporates moving obstacles (similar to [13] and our GDW-D). A key focus of their work was that the robot motion should not only be safe but should also be perceived as comfortable by pedestrians and they demonstrated it via field trials. Fixed laser range finders installed around the environment were used to sense humans.

Mueller at al. [24] combine an A* planner with a people tracking system (uses a SICK LMS laser scanner and a Kalman filter-based multi-target tracker to detect and track people) that enables a robot to move with groups of people resulting in a more human-like way of navigation among people.

More recently, machine learning approaches have been investigated for robot navigation [25]. This methodology needs a huge training data set for robot in different situations. Kollmitz et al. [26] incorporate a social force element along with an A* search to plan collision-free paths that take human comfort into account. A Kalman filter based detection module estimates the position and velocity of the interacting pedestrian from scans of an on-board 2D laser range finder.

III. Global Path Planner with Local Dynamic Window – GDW-S and GDW-D

The GDW implementation available in ROS hydro navigation stack [27] was adapted to create the GDW-S Planner and further modified to create the GDW-D Planner. The overall high level description for both is shown in Figure 1. Using the current sensed map, the global planner (uses numerical navigation function [1]) is invoked to get a path for the robot from its current

position to the desired goal position. In this map, all the obstacles (both static and moving ones) are treated as instantaneously static. Suitably discretized points along the global path are then passed on to a local DW Planner. The DW planner searches for a locally optimal translational and rotational velocity (v,ω) (also called control command) while taking into account the dynamic constraints (maximum accelerations, both linear and rotational) of the robot. The search space is reduced to a local "dynamic window", centered on the current velocity (v_c,ω_c) and consists of the set of achievable velocities V_a, given its maximum and minimum acceleration constraints $\{a_{min},a_{max}\}$ and $\{\dot{\omega}_{min},\dot{\omega}_{max}\}$. This set is computed for a suitably discretized number of sample controls (v samples, and ω samples). Within this dynamic window, the set of velocities that would result in a collision with an obstacle are discarded and the remaining set is the set of admissible velocities, V_{ad}. The admissible velocity that maximizes an objective function is then chosen as the next control command. The objective function includes a measure of alignment with the goal direction, progress towards the goal location, the distance from the global path, and the distance from the provided global goal.

In the Hydro version of ROS package, the GDW-S planner implementation uses a set of "evaluating or critic functions filters" to determine the optimal control. Each filter corresponds to a component of the objective function. A filter can be a binary filter (e.g., a trajectory is in collision or free) that rejects the inadmissible trajectories and passes the admissible ones or a cost filter (e.g., progress toward goal measured via distance to the goal) that adds its portion of the cost to the cost of the trajectory. The control corresponding to the trajectory with the lowest cost is then chosen as the control command to be executed by the robot. For simplicity, we used only one critic function for the GDW-S planner - the distance to the goal. So the GDW-S planner prefers controls that lead it closer to the goal. An algorithm similar to that of Seder et al [13] was implemented to incorporate moving obstacles' dynamics into GDW-S planner in the ROS navigation stack and we call the resulting planner GDW-D. Essentially, it assumes that a module is available that predicts the trajectories of the dynamic obstacles (valid for a certain duration, Δ_o). The trajectories of moving obstacles as well as that of the robot over the planning horizon are forward simulated (for time Δ_p, the planning horizon, a planning parameter) and the collision check of the robot along an admissible trajectory is done with respect to the forward simulated motions of the dynamic obstacles. This was implemented as a new binary critic function – Dynamic Obstacle Critic - that rejects a robot trajectory if it collides with any moving obstacles' predicted trajectory. The overall high level schematic for both the GDW-S and GDW-D planners is shown in Figure 1.

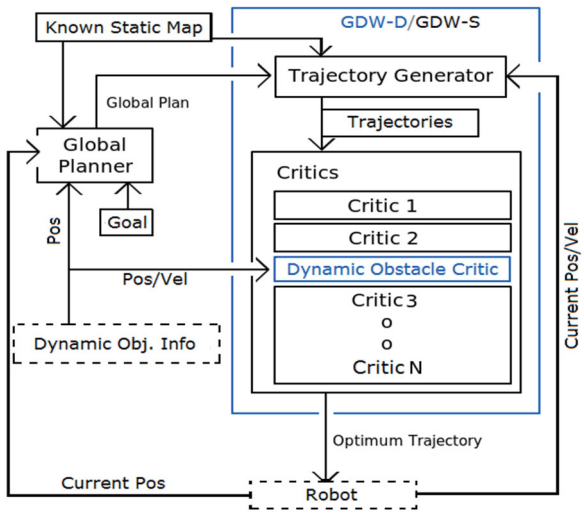

Figure 1. Overall high level schematic flowchart for ROS implementation of GDW-S and GDW-D Planners.

Interleaving of planning and execution, similar to that in [21] is shown in Figure 2. The robot's trajectory is represented by black lines. The solid line segments are the executed part of the robot's trajectory. The dotted line segments are the planned part of the robot's trajectory that is not executed because a newly re-planned trajectory is available for execution. The obstacle's trajectory is represented by blue lines (gray in the black and white version). The solid line segments are the actual obstacle's trajectory, while the dotted line segments represent the predicted trajectory that was used by the planner. The planning horizon, i.e., Δ_p was chosen to be 4 seconds in our simulations (ROS term for this is sim_time). Given the robot's parameters, and the fact there is uncertainty in real situations regarding predicted motion of humans, this was deemed to be a good compromise. The re-planning rate for the entire planning cycle was determined as follows (ROS term for this is controller frequency which corresponds to $1/\Delta_e$). It was chosen empirically so that the planner is able to plan a trajectory within this duration. GDW-D requires more computations due to collision checks with moving obstacles that require forward simulation of obstacle trajectories. For our hardware set-up (Intel i5 760 @ 2.8 GHz with 8 G RAM), it was observed that $\Delta_e \geq 0.5$ seconds which corresponds to a re-planning rate of $\leq$ 2Hz. So we chose 2Hz as the re-planning rate for GDW-D. For GDW-S, we could push the re-planning rate to twice as fast, i.e., to 4Hz. So we tested GDW-S at 4Hz, and for comparison purposes also at 2Hz.

The robot is a differential drive robot and maximum acceleration parameters were chosen to be consistent with the Powerbot mobile base in our Lab and the max linear velocity limit was chosen so that robot is moving at a "normal walking" speed. The robot and various planner parameters are listed in TABLE I below.

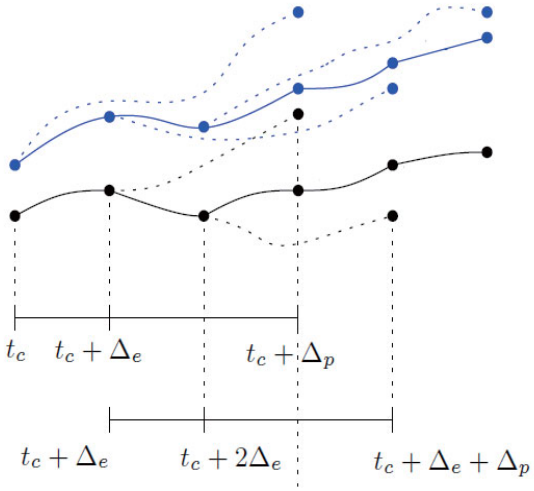

Figure 2. A schematic showing key planner parameters and re-planning timings.

TABLE I. Robot and Planner Parameters in Simulations

Robot's Dynamic Parameters		Planner Parameters	
Parameter	**Value**	**Parameter**	**Value**
v_{min}, v_{max}	0, 0.45 m/s	sim_time = Δ_p	4 s
ω_{min}, ω_{max}	0, 2 rad/s	# of v samples	5
a_{max}	6 m/s^2	# of ω samples	20
$\dot{\omega}_{max}$	4 rad/s^2	control cycle = Δ_e	0.5 s

IV. SIMULATIONS

A. Settings and Results

We used Stage [14], a 2D multiple-robot simulator as it facilitates simulating multiple fairly simple, computationally cheap agents rather than attempting to emulate any device with great fidelity in a two-dimensional bitmapped environment. In our simulations, we pass robot as well as dynamic obstacle position and velocities from Stage to the ROS planners via a stage interface node and these blocks are shown in dotted in Figure 1.The simulations were run in four different sets of static environments mimicking indoor environments: The first one is an open large hallway like environment bounded by walls with no static obstacles in it (Environment I), whereas the second environment simulates an office type situation and includes three large static obstacles inside its boundary walls (Environment II), while the third environment includes a narrow passage (Environment III), and finally the fourth environment includes multiple hallways and narrow passages (Environment IV). Due to lack of space, only Environments II and III are shown in Figure 3.

For environment I, the simulations were run with i) eleven and ii) with fifteen dynamic obstacles; and for environment II, the simulations were run with i) eight and ii) with eleven dynamic obstacles. The dynamic obstacles were spread out in the environment and their nominal motions were in horizontal or vertical (unless the repulsion force makes them deviate off course) and they keep moving back and forth. Finally, for giving the moving obstacles a capability to avoid collisions, we use a

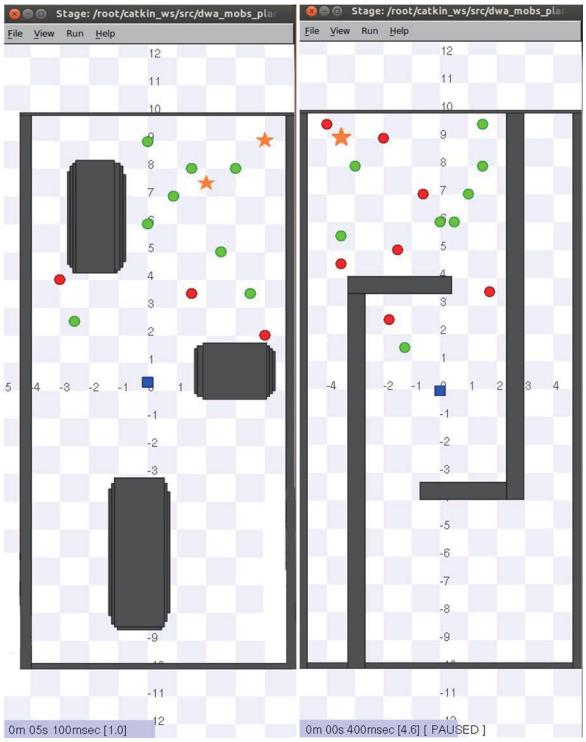

Figure 3. Static environments in Stage. The robot (small square) is shown in its initial position and the goal positions used in simulations are shown with stars. Dynamic obstacles are shown as circles where the green ones move vertically and the red ones move horizontally. See attached video for a sampling of the simulation runs.

simple potential field model and denote such environments with label +R. The model essentially applies a repulsive force on obstacle governed by (1) given below, where d is the distance between the robot and another agent (robot or another moving obstacle), d_{min} is the distance at which the repulsion force starts to be applied, and d_{coll} is the distance at which a collision is deemed to occur between O_i and the agent, and K is a scaling factor. With r_r = 0.3m being the robot radius and r_o = 0.3m being the obstacle radius, in our simulations, $d_{coll} = r_o + r_r$ = 0.6 m, and d_{min} = 1.4 m, and the constant K = 0.8.

$$F_{O_i} = \begin{cases} K\left(\frac{1}{d - d_{coll}} - \frac{1}{d_{min}}\right) & d_{coll} < d < d_{min} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Furthermore, we tested three different speeds - low, medium, and high - for the dynamic obstacles at 0.25, 0.5, and 0.75 (m/s), respectively. These speeds were chosen to mimic humans with slow, moderate and fast walking humans in indoor environments. Finally, we had two different goals for the robot to reach (shown as * in Figure 3). For each goal and for each speed of dynamic obstacles, we executed 5 runs. In each run, the start position of the dynamic obstacles along their nominal trajectories were varied. Two performance metrics were used: i) percentage of successful runs, i.e., where a successful run means that the robot was able to reach its goal without any collisions with the obstacles, static as

well as dynamic, and ii) total number of collisions, where we let the simulations run for a longer time (to the goal and back to the start) and count the total number of collisions robot had with the obstacles. For each metric, we combined all the results for each planner for all environments and for statistical significance, used the two-tailed Z-test with significance level of 0.05. We first tested two null hypotheses: i) there is no difference between the performance of GDW-S (run at 4Hz) and GDW-D (re-planning rate 2Hz), ii) there is no difference between performance of GDW-S+R (re-planning rate 4Hz) and GDW-D+R (re-panning rate 2Hz). These correspond to conjecture 1. We then tested two additional hypotheses that correspond to conjecture 2: iii) there is no difference between the performance of the GDW-S and GDW-S+R, and iv) there is no difference between the performance of GDW-D and GDW-D+R planners. A p-value < 0.05 indicates that the result is deemed significant and the null hypothesis is rejected.

B. Results and Discussion

It is impossible to include videos of all runs but the attached video gives a sampling of the runs. The overall simulation results are summarized in TABLE II . The statistical significance between paired planners is shown in TABLE III. The results in TABLE II clearly show that conjecture 1, i.e., a planner for static obstacles, if executed at a fast update rate, will be quite effective in dealing with dynamic obstacles, is false. The comparison between GDW-D and GDW-S planners illustrates that the overall success percentage for GDW-S (30%), even at a faster re-planning rate of 4Hz is substantially lower than for GDW-D (59%) running at the lower re-planning rate. Moreover, the corresponding p-value (row 1 in TABLE III) indicates that the better performance of GDW-D is statistically significant. Therefore, adding the capability of predicting the dynamic obstacles' trajectories and incorporating them in the planning process, results in a major improvement in the performance compared to the planner that treats moving obstacles as instantaneously static, even if it is running at a faster re-planning rate.

Our results also confirm that conjecture 2, i.e., planners are effective in dynamic human environments (with humans being dynamic obstacles) primarily because humans themselves avoid the robot and if this were not the case, robot will run into collisions much more frequently, is true. The simple avoidance behaviour of obstacles greatly increases the percentage of successful runs for GDW-S planner, i.e., GDW-S+R has much more successful runs (51%) than GDW-S (30%) and it is statistically significant.

On the other hand, GDW-D+R does have more successful runs (68%) than GDW-D (59%), however the improvement is much smaller and was not statistically significant (p-value > 0.05). Overall, GDW-D+R outperformed GDW-S+R (68% vs. 51% with p-value = 0.01 and hence statistically significant) which shows the importance of incorporating dynamic obstacle motions in the planning process even with environments where humans avoid the robot.

V. EXPERIMENTAL ANALYSIS

Although it is not the focus of this paper, we briefly outline our recent implementation of both GDW-D and GDW-S planners on real hardware, the mobile base in our lab, a Powerbot, and our preliminary experiments with a single person walking towards the robot. The experimental setup is shown in Figure 5. A SICK LMS 100 sensor mounted at the base of the Powerbot has been used to build the environment's map since it has a long range up to 30 (m), while a Velodyne HDL-32e sensor (mounted at about human eye level on a long cylindrical mount on the robot) was used to detect a moving human and his/her velocity as required by the GDW-D planner. Velodyne sensor range is about 6 m. Human torsos were detected using a basic segmentation of the point cloud data from the Velodyne and a Kalman filter was used to track the motion and the current velocity of the human obstacle was used to forward simulate the motion for planning purposes. The attached video shows running both GDW-D and GDW-S planners while a single person was walking towards the robot, and the results show that the GDW-D planner reacts to the upcoming human and starts turning while at a much larger distance (about 5 meters) from the human as compared to the GDW-S planner which starts turning much later (at about 1.8 meters) and in fact would have collided with the human, had the human not avoided the robot.

Figure 4. The experimental setup on Powerbot

VI. CONCLUSION AND FUTURE WORK

In this paper, we showed via simulation results that a planner for static obstacles, even if executed at higher re-planning rates shows poor performance measured by number of successful runs (robot reaches goal without any collisions) as compared to a planner that incorporates

dynamic obstacles' predicted trajectories in the planning process. However, if the dynamic obstacles have the ability to avoid the robot (as humans do), then the performance of the static planner greatly improves, and the improvement is statistically significant. Nevertheless, the planner incorporating predicted trajectories of dynamic obstacles still outperforms the static planner in the environments where dynamic obstacles have the ability to avoid the robot and other obstacles as humans do. The static and dynamic planners were implemented within ROS and were run under identical common parameters.

For future work, we are planning to run extensive experiments with the mobile robot moving around amongst multiple humans in several indoor environments.

REFERENCES

[1] J. C. Latombe, "ROBOT MOTION PLANNING," Springer, 1990.

[2] P. Fiorini, and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," Int. Journal of Robotics Research, vol. 17, no. 7, pp. 760-772, 1998.

[3] M. Phillips, and M. Likhachev, "Sipp: Safe interval path planning for dynamic environments," *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[4] R. Vatcha, and J. Xiao, "Perceived CT-space for motion planning in unknown and unpredictable environments," *in 8th Workshop on Algorithmic Foundations of Robotics*, December 2008.

[5] O. Brock, and O. Khatib, "High-Speed Navigation Using the Global Dynamic Window Approach," *IEEE International Conference on Robotics and Automation*, 1999.

[6] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, pp 23-33, March 1997.

[7] T. Fraichard, "A short paper about motion safety," *IEEE International Conference on Robotics and Automation*, 2007.

[8] S. Thrun, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Haehnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "Minerva: A second generation mobile tour-guide robot," *in Proc. Of the IEEE Int. Conf. on Robotics and Automation*, Detroit, MI (US), May 1999.

[9] W. Burgard, A. Cremers, D. Fox, D. Haehnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "Experiences with an interactive museum tour-guide robot," *Artificial Intelligence*, vol. 114, no. 1-2, 2000.

[10] R. Philippsen, and R. Siegwart, "Smooth and efficient obstacle avoidance for a tour-guide robot," *in Proc. of the IEEE Int. Conf. on Robotics and Automation*, Taiwan, September 2003.

[11] P. Trautman, and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[12] http://www.ros.org/wiki/

[13] M. Seder, and I. Petrovic, "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles," *IEEE International Conference on Robotics and Automation*, 2007.

[14] R. Vaughan, "Massively Multiple Robot Simulations in Stage", Swarm Intelligence 2(2-4):189-208, 2008. Springer.

[15] R. C. Sprinthall, "Basic Statistical Analysis (9th ed.)," Pearson Education, 2011.

[16] M. Shiomi, F. Zanlungo, K. Hayashi, and T. Kanda, "Towards a socially acceptable collision avoidance for a mobile robot navigating among pedestrians using a pedestrian model", Int J Soc Robot 6:443–455, 2014.

[17] AK. Pandey , R. Alami, "A framework towards a socially aware mobile robot motion in human-centered dynamic environment", In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2010.

[18] J. Rios-Martinez, A. Renzaglia, A. Spalanzani, A. Martinelli, and C. Laugier, "Navigating between people: a stochastic optimization approach", In IEEE International Conference on Robotics and Automation (ICRA), 2012.

[19] S. Petti, and T. Fraichard, "Safe motion planning in dynamic environments," *in Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Edmonton, AB (CA), August 2005.

[20] M. Otte, and E. Frazzoli, "RRTX: Asymptotically Optimal Single-Query Sampling-Based Motion Planning with Quick Replanning", The International Journal of Robotics Research, 2015.

[21] B. L'Espérance, and K. Gupta, "Safety Hierarchy for Planning With Time Constraints in Unknown Dynamic Environments," *IEEE Transactions on Robotics*, Vol. 30, No. 6, pp. 1398 – 1411, December 2014.

[22] S. Oli, B. L'Espérance, and K. Gupta, "Human Motion Behaviour Aware Planner (HMBAP) for path planning in dynamic human environments", In Advanced Robotics (ICAR), 2013.

[23] D. Helbing, and P. Molnar, "Social force model for pedestrian dynamics," *Physical Review E*, 51:4282, 1995.

[24] J. Mueller, C. Stachniss, K. Arras, and W. Burgard, " Socially inspired motion planning for mobile robots in populated environments", *In Proc. of International Conference on Cognitive Systems*, 2008.

[25] M. Kuderer, H. Kretzschmar, and W. Burgard, "Teaching mobile robots to cooperatively navigate in populated environments", In: IEEE international conference on intelligent robots and systems, 2013.

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

[27] http://wiki.ros.org/navigation, last accessed on September 15, 2017.

TABLE II. Overall Results for GDW-S, GDW-D, GDW-S+R and GDW-D+R

Planner	**GDW-S**		**GDW-D**	**GDW-S+R**		**GDW-D+R**
	@ 2 Hz	**@ 4 Hz**		**@ 2 Hz**	**@ 4 Hz**	
% Success for all combined runs	36/120 = 30%	36/120 = 30%	71/120 = 59%	61/120 = 51%	68/120 = 57%	82/120 = 68%
Total # of collisions for all runs	422/90	416/90	141/90	99/90	93/90	65/90

TABLE III. P-values for Paired Planners

Paired Planners	**p-value for % success (M1)**	**p-value for # collisions (M2)**
GDW-S (4 Hz) vs. GDW-D (2Hz)	0.00	0.00
GDW-S vs. GDW-S+R (both 2Hz)	0.00	0.00
GDW-D vs. GDW-D+R (both 2Hz)	0.14	0.00
GDW-S+R vs. GDW-D+R (both 2Hz)	0.01	0.01
GDW-S+R (4 Hz) vs. GDW-D+R (2Hz)	0.06	0.02