

HuNavSim: A ROS 2 Human Navigation Simulator for Benchmarking Human-Aware Robot Navigation

Noé Pérez-Higueras , Roberto Otero , Fernando Caballero , and Luis Merino , *Member, IEEE*

Abstract—This work presents the Human Navigation Simulator (*HuNavSim*), a novel open-source tool for the simulation of different human-agent navigation behaviors in scenarios with mobile robots. The tool, the first programmed under the ROS 2 framework, can be used together with different well-known robotics simulators like Gazebo. The main goal is to facilitate the development and evaluation of human-aware robot navigation systems in simulation. In addition to a general human-navigation model, *HuNavSim* includes, as a novelty, a rich set of individual and varied human navigation behaviors and an comprehensive set of metrics for social navigation benchmarking.

Index Terms—Human-aware motion planning, performance evaluation and benchmarking, simulation and animation.

I. INTRODUCTION

THE evaluation of the robot skills to navigate in scenarios shared with humans is essential to envisage new service robots working along people. The development of such mobile social robots poses two main challenges: first, real experimentation with people is costly, as well as being difficult to perform and to replicate except for very limited, controlled scenarios. In addition, human participants may be at risk, mainly during the initial stages of development. Therefore, simulating realistic human navigation behaviors for the development of robot navigation techniques is necessary. And secondly, the evaluation not only requires the assessment of the navigation efficiency, but also the safety and comfort of the people. This latter requirement is a human feeling which is difficult to quantify through mathematical equations. Thus, there is not solid agreement in the research community on a suitable set of metrics for human-aware navigation.

Most state-of-the-art simulation approaches are based on models of crowd movement to control the behavior of simulated human agents. Whereas this is valid to obtain a collective

Manuscript received 4 May 2023; accepted 4 September 2023. Date of publication 15 September 2023; date of current version 25 September 2023. This letter was recommended for publication by Associate Editor I. Karamouzas and Editor A. Bera upon evaluation of the reviewers' comments. (*Corresponding author: Noé Pérez-Higueras.*) This work was supported in part by Programa Operativo FEDER Andalucía 2014–2020, in part by Consejería de Economía, Conocimiento y Universidades under Grant DeepBot, PY20_00817, in part by the Projects NHOA under Grant PLEC2021-007868, in part by NORDIC under Grant TED2021-132476B-I00, in part by MCIN/AEI/10.13039/501100011033, and in part by the European Union through NextGenerationEU/PRTR.

The authors are with the School of Engineering, Pablo de Olavide University Seville, 41013 Seville, Spain (e-mail: noeperez@upo.es; rotegal@alu.upo.es; fcaballero@upo.es; lmercab@upo.es).

This letter has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2023.3316072>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2023.3316072

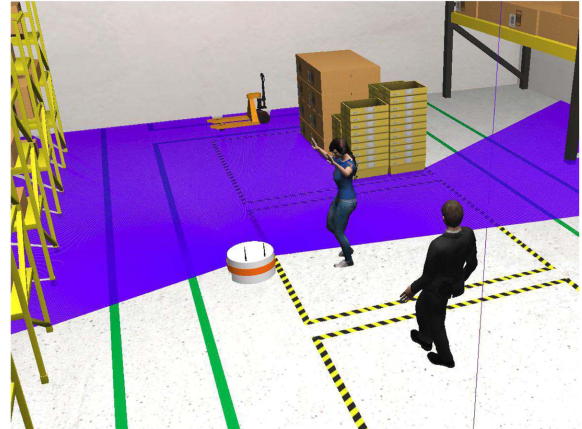


Fig. 1. Capture of HuNav agents in the Gazebo Simulator.

behavior of the agents, it loses realism at the local level since the behavior of all individual agents is exactly the same for the same scenarios. Here, we propose a set of individual human behaviors related to reactions to the presence of a robot.

Another issue is the evaluation of the human-aware navigation through metrics. Each benchmarking tool usually presents its own set of metrics. While research in new realistic “social” metrics is needed, the absence of common well-known metrics hinders the comparison of different techniques for social robot navigation.

With the *HuNavSim* we aim to contribute to the solutions of the two problems mentioned: providing a set of different behaviors for individual agents closer to the reality, and presenting a compilation of metrics employed in the literature. In a nutshell, we present the following contributions:

- 1) An open-source and flexible simulation tool of human navigation under the ROS 2 framework [1] that can be used together with different robotics simulators.
- 2) A rich set of navigation behaviors of the human agents, which includes a set of frequent individual reactions to the presence of a robot.
- 3) A wide compilation of metrics into the tool from the literature for the evaluation of human-aware navigation, which is configurable and extensible.
- 4) A wrapper to use the tool along with the well-known Gazebo simulator used in Robotics (see Fig. 1).

II. RELATED WORK

Different simulators and benchmarking tools for human-aware navigation problems can be found in the literature. We provide a brief review of existing related software and highlight the differences and similarities with our approach.

*PedSimROS*¹ is an adaptation to ROS of a basic library that uses Social Force Model (SFM) [2] to lead the crowd movement. *MengeROS*² [3] is a more comprehensive tool that uses A^* and potential fields for global path planning, and allows for the selection among a set of collision-avoidance strategies such as different variations based on SFM, Optimal Reciprocal Collision Avoidance (ORCA) [4] and Pedestrian Velocity Obstacle (PedVO) [5]. However, both *PedSimROS* and *MengeROS* are deprecated software integrated into unmaintained versions of ROS 1. Also, they do not incorporate any option for navigation evaluation. *CrowdBot*³ [6] and *SEAN (Social Environment for Autonomous Navigation)*⁴ [7], [8] are more recent, advanced and ambitious tools. They share similar features. Both are based on the game engine Unity and ROS 1, and both aim at becoming the standard for evaluating robot navigation in populated environments. In contrast to them, *HuNavSim* provides a more flexible approach that allows using the tool along with different simulators and provides a set of individual and more realistic human reactions to the presence of a robot. Moreover, these tools present a closed set of metrics while *HuNavSim* includes a larger compilation of metrics that is easily configurable and extendable.

Another interesting simulator is the *Intelligent Human Simulator (InHuS)* [9], [10]. This simulator is meant to control the movement of human agents in another simulator. It also includes a small set of human individual behaviors, as *HuNavSim* does. In contrast to *InHuS*, our simulator employs a powerful and flexible tool, behavior trees [11], to define and drive particular human behaviors. These trees can also be easily modified and extended. Moreover, *InHuS* is based on ROS 1 and it mainly employs a navigation system devised for robots, *HATEB2* [12], to control human movements. This could lead to more unrealistic crowd movement than simulators based on specific crowd movement models.

*SocNavBench*⁵ [13] follows a different approach. It is a simulator-based benchmark with pre-recorded real-world pedestrian data replayed. The main drawback of this tool is that pedestrians' trajectories are replayed from open-source datasets and, therefore, the effects of robot motions in pedestrians' paths are not considered. That makes it difficult to obtain a realistic evaluation of the human-aware navigation.

Other approaches are devised for training, testing, and benchmarking learning-based navigation agents in dynamic environments. For instance, *Arena-Rosnav* [14] allow for the training of obstacle avoidance approaches based on Deep Reinforcement

Learning. It makes use of *PedSim* library to guide pedestrians' movements. Then, *Arena-Rosnav* was extended to include navigation benchmarking, *Arena-Bench* [15], and a larger set of utilities for development and benchmarking (*Arena-rosnav 2.0* [16]). For the simulation and training of reinforcement learning agents in spaces shared with humans, we found *SocialGym 2.0* [17], [18], which also uses the *PedSim* library to simulate human navigation. These tools do not include particular human behaviors nor a specific set of metrics for human-aware navigation.

Finally, there are interesting evaluation tools like *BARN (Benchmark for Autonomous Robot Navigation)*⁶ [19] and *Bench-MR*⁷ [20]. However, those are oriented to the general problem of navigation in cluttered environments without considering the "social" component of spaces shared with humans.

III. HUMAN NAVIGATION SIMULATOR (HUNAVSIM)

A. Architecture of the Simulator

The general architecture of the simulator can be seen in Fig. 2. *HuNavSim* is in charge of properly controlling the pose in the scene of the human agents spawned in another base simulator such as Gazebo, Morse, or Webots, which also must simulate the scenario and the robot. Therefore, a wrapper to communicate with the base simulator is required.

Initially, the number and characteristics of the agents to be simulated (like individual behavior, list of goals, etc) must be provided by the user. It can be specified through a configuration `yaml` file, or through a graphic user interface based on a ROS 2 RViz panel.

Then, at each execution step, the simulator wrapper sends the current human agents' status to the *hunav_manager* module through ROS 2 service calls. According to the current state, the system decides the next state of the agents which are returned to the wrapper, and thus updated in the base simulator.

Finally, the *hunav_evaluator* module records the data of the experiment and computes the evaluation metrics at the end of the simulation. The information about each simulation and the desired set of metrics to be computed can be specified through a `yaml` file or through a list of checkboxes shown in a ROS 2 RViz panel. The module generates a set of output result files with the simulation info and the names and values of the computed metrics.

The complete documentation and code of the *HuNavSim* is available at https://github.com/robotics-upo/hunav_sim

B. Human Navigation Modeling

The *HuNavSim* is primarily based on the use of the well-known Social Force Model [2] and its extension for groups [21], [22], to lead the human agents movement similar to other crowd simulators. The simplicity and good results of this model allowed us to easily extend it to provide a set of different and individual navigation reactions of the agents to the presence

¹[Online]. Available: https://github.com/srl-freiburg/pedsim_ros

²[Online]. Available: https://github.com/ml-lab-cuny/menge_ros

³[Online]. Available: <http://crowdbot.eu/CrowdBot-challenge/>

⁴[Online]. Available: <https://sean.interactive-machines.com/>

⁵[Online]. Available: <https://github.com/CMU-TBD/SocNavBench>

⁶[Online]. Available: https://www.cs.utexas.edu/~attruong/metrics_dataset.html

⁷[Online]. Available: <https://github.com/robot-motion/bench-mr>

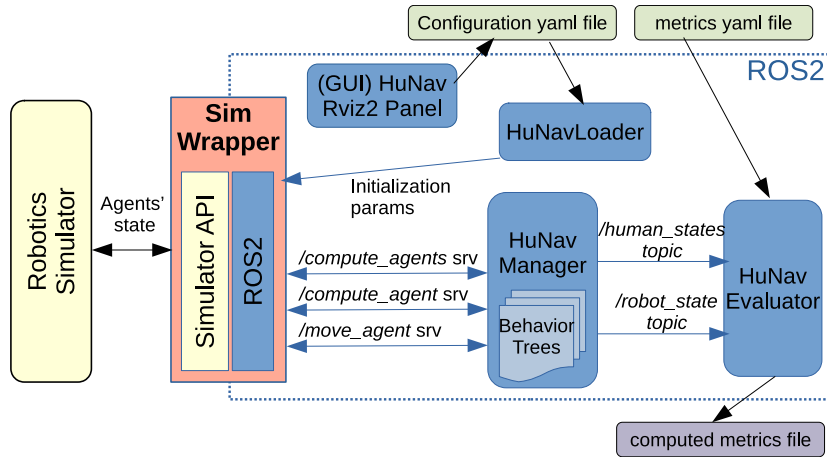


Fig. 2. Diagram of *HuNavSim*. The core modules of *HuNavSim* are in blue. The base robotics simulator modules are depicted in light yellow color. Green boxes indicate the configuration input data. Output evaluation metrics are indicated in light purple color.

of the robot. That allows for the enrichment of the navigation scenarios and to challenge the navigation algorithms with more diverse and realistic human behaviors. The set of behaviors included are the following:

- *regular*: this is the regular navigation based on SFM. The robot is included as another human agent, so the human treats the robot like another pedestrian.
- *impassive*: in this case, the robot is not included as a human agent in the SFM. Therefore, the human deals with the robot as an obstacle.
- *surprised*: when the human sees the robot, the current goal navigation is stopped. The human stops walking and starts to look at the robot (heading to the robot position).
- *curious*: the human abandons his/her current navigation goal when the robot is detected. Then, the human approaches the robot slowly. After a short time (30 seconds by default) the human loses interest and goes back to the original navigation goal.
- *scared*: when the human agent detects the robot, it tries to stay far away from it. In this case, we add to the agent an extra repulsive force from the robot and we decrease the maximum velocity. This way we simulate a scared human behavior.
- *threatening*: when the agent sees the robot, it approaches the robot and tries to block the robot’s path by walking in front of it. A goal in front of the robot is continuously computed and commanded. After some time (40 seconds by default), the agent gets bored and continues its original navigation.

An interesting feature is that all these behaviors are controlled by behavior trees [11]. They make possible efficient structuring of the switching between the different tasks or actions of the autonomous agents. Moreover, they are easily programmable, and new behaviors can be added or modified with minimal effort. We use the engine *BehaviorTree.CPP*⁸ for behavior creation and editing.

⁸[Online]. Available: <https://github.com/BehaviorTree/BehaviorTree.CPP>

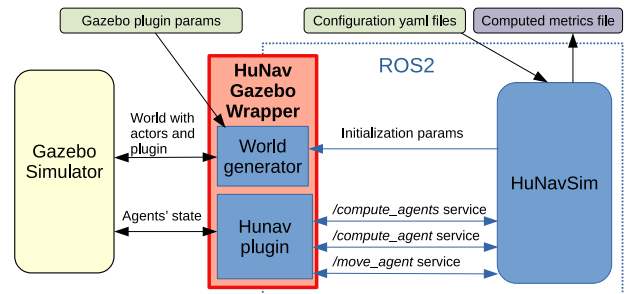


Fig. 3. Diagram of the Gazebo wrapper for *HuNav Simulator*. The red square encapsulates the two modules of the wrapper, which intermediates between the Gazebo simulator and the *HuNavSim*.

C. Wrapper for Gazebo Simulator

A wrapper to use *HuNavSim* along with the Gazebo simulator is also provided. Fig. 3 shows the wrapper modules (red square) and the communication with the Gazebo simulator and *HuNavSim*.

The control of the agent movement has been programmed as a Gazebo world plugin that must be included in the Gazebo world file. Also the human agents are included as a Gazebo actor in the same file. For that reason, we have programmed the following two modules:

- A *world_generator* module which is in charge of reading the configuration parameters of the human agents (defined in the agents’ configuration *yaml* file), and to write the proper plugin and agents in the base Gazebo world file.
- Then, Gazebo loads this world file and the *HuNav plugin* communicates with *HuNavSim* to update the agents’ status (model pose in the world) during the execution of the simulation.

A set of typical scenarios shared with humans is included in the wrapper: a cafeteria, a warehouse, and a house. Moreover, we have developed a new set of different 3D human models. The human 3D model and the associated behavior can be selected by the user. Since *HuNavSim* only determines the agents’ global position and orientation in the Gazebo world, we try to provide an

TABLE I
 AVAILABLE METRICS IN *HuNavSim*

Metric	Source	Description	units
time_to_reach_goal	Pérez et al.[23]	Time spent to reach the goal	s
path_length	Pérez et al.[23]	Length of the robot path	m
cumulative_heading_changes	Pérez et al.[23]	Added absolute heading angle differences between successive waypoints	rad
avg_dist_to_closest_person	Pérez et al.[23]	Distance to the closest person at each time step on average	m
min_dist_to_people	Pérez et al.[23]	The minimum distance to any person along the trajectory	m
intimate_space_intrusions	Pérez et al.[23]	Time percentage the robot spent in a person intimate space. See Proxemics[24]	%
personal_space_intrusions	Pérez et al.[23]	Time percentage the robot spent in a person personal space [24]	%
social+_space_intrusions	Pérez et al.[23]	Time percentage the robot spent in a person social or public space [24]	%
group_intimate_space_intrusions	Pérez et al.[23]	Time percentage the robot spent in a group intimate space	%
group_personal_space_intrusions	Pérez et al.[23]	Time percentage the robot spent in a group personal space	%
group_social+_space_intrusions	Pérez et al.[23]	Time percentage the robot spent in a group social or public space	%
completed	SEAN2.0 [7]	Whether the robot reached the goal or not	-
min_dist_to_target	SEAN2.0 [7]	The closest the robot passes to the target (or goal) position	m
final_dist_to_target	SEAN2.0 [7]	Distance between the last robot position and the target (goal) position	m
robot_on_person_collisions	SEAN2.0 [7]	Number of times robot collides with a person	-
person_on_robot_collisions	SEAN2.0 [7]	Number of times person collides with a robot	-
time_not_moving	SEAN2.0 [7]	Seconds that the robot was not moving	s
avg_robot_linear_speed	SocNavBench[13]	Average linear speed of the robot along the trajectory	m/s
avg_robot_angular_speed	SocNavBench[13]	Average absolute angular speed of the robot along the trajectory	rad/s
avg_robot_acceleration	SocNavBench[13]	Average acceleration of the robot along the trajectory	m/s ²
avg_robot_jerk	SocNavBench[13]	Average jerk (time derivative of acceleration) of the robot along the trajectory	m/s ³
avg_pedestrians_velocity	Katyal et al.[25]	Average linear velocity of all the pedestrian along the trajectory	m/s
avg_closest_pedestrian_velocity	Katyal et al.[25]	Average linear velocity of the closest pedestrian along the trajectory	m/s
social_force_on_agents	SFM-based[2]	Sum of the modulus of the social forces provoked by the robot on the human agents along the trajectory	m/s ²
social_force_on_robot	SFM-based[2]	Sum of the modulus of the social forces provoked by the agents on the robot along the trajectory	m/s ²
obstacle_force_on_agents	SFM-based[2]	Sum of the modulus of the obstacle forces on the agents along the trajectory	m/s ²
obstacle_force_on_robot	SFM-based[2]	Sum of the modulus of the obstacle forces on the robot along the trajectory	m/s ²
social_work	SFM-based[2]	The sum of the social force on the robot, the obstacle force on the robot, and the social force on the agents	m/s ²

agent’s natural body movements by using different animations to better express the different agent behaviors visually. However, producing a very realistic body movement is beyond the scope of this work.

The wrapper documentation and code are publicly available at https://github.com/robotics-upo/hunav_gazebo_wrapper

IV. NAVIGATION EVALUATION

With the aim of tackling the problem of selecting the best metrics for evaluation of the human-aware navigation, we decided to keep the evaluation system as open as possible.

First, we reviewed the literature in order to collect most of the metrics applied to the issue. Then, we let the user select the metrics to be computed for each simulation. Finally, the system also allows for the easy addition of new metrics to the evaluation. The input of all the metric functions consists of two arrays with the poses, velocities, and other data of the *HuNavSim* agents and the robot for each time step.

Besides the computation of the metrics for all the agents in the scenario, the evaluator also performs the calculation regarding the particular behavior of the agents. In this way we can obtain interesting metrics for the situations in which the agents show a specific behavior to the presence of the robot.

The main objective here is to validate the usefulness of *HuNavSim* for comparing planners under the given model. The main contribution is the possibility of including individualized

behaviors, which therefore allows one to evaluate the same planners in a wider range of situations.

Thus, *HuNavSim* presents a reliable and flexible evaluation system in contrast to the fixed evaluation systems currently found in the literature.

A. Metrics Description

At the time of writing this work, the metrics implemented were those employed in our previous work [23], the *SEAN* simulator [7], some of the *SocNavBench* [13], and some metrics related to the Social Force Model (SFM) employed in different works according to the compilation of Gao et al. [26]. This results in a total of 28 metrics, and it represents, to the best of our knowledge, the most extensive compilation of metrics. The list of available metrics can be seen in Table I. A detailed description of the metrics along with their mathematical equations can be reviewed on the Github page of the *HuNavEvaluator*⁹

B. Output Metrics Data

As output, the *HuNavEvaluator* will generate a set of metrics files. They are formed as text files starting with headers indicating the metrics names, followed by the data organized in a tabular form of rows and columns:

⁹[Online]. Available: https://github.com/robotics-upo/hunav_sim/tree/master/hunav_evaluator

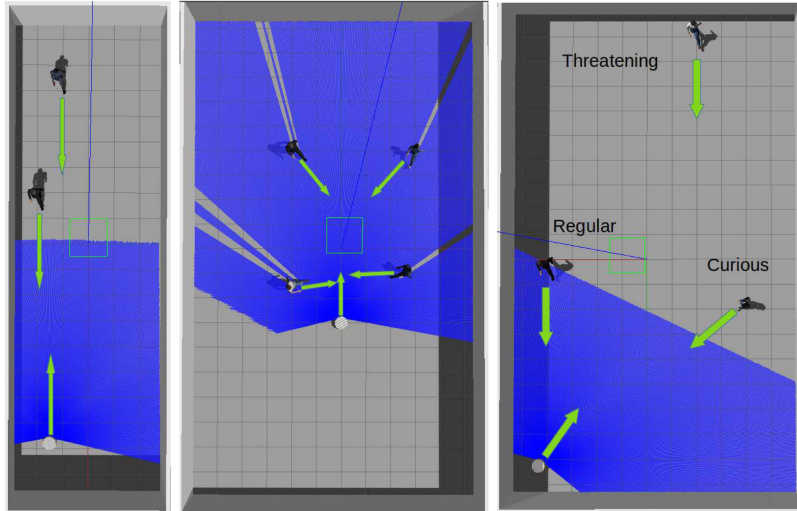


Fig. 4. Scenarios for evaluation. Left image: passing scenario in a corridor, center image: crossing scenario, right image: combined scenario with different human navigation behaviors. Green arrows indicate the direction of the navigation goal for the agents and robot.

- General metrics file, with the values of the final metrics computed.
- List of metrics values computed for each time step in the simulation. This is very useful to plot and to visualize the behavior of some metrics over the simulation time or at specific moments.
- Set of files with the final metrics but only considering the groups of people with the same behavior. We will obtain as many files as particular behaviors we use in the simulation.
- Finally, the tool will generate the metrics computed at each time stamp for each previous behavioral group.

V. VALIDATION EXPERIMENTS

To assess the *HuNavSim* as a benchmarking tool, we employ three different planners in different socially-aware navigation situations. Then, the metrics produced by the tool are used to compare the social compliance of the navigation.

A. Scenarios

According to Gao et al. [26], the scenarios commonly used in evaluating human-aware navigation are: *Passing*, *Crossing*, *Overtaking*, *Approaching*, *Following/Accompanying* and a *Combination* of those. In this evaluation, we employ the basic scenarios for *Passing* and *Crossing*, and a *Combined* scenario with humans showing different individual behaviors. In Fig. 4 we can see the *Passing* scenario in the left image, in which the robot must reach the other side of the room while two people are walking in the opposite direction. In case of the *Crossing* scenario, center image of Fig. 4, the robot must again reach the opposite side of the room while crossing with four people: two of them walk perpendicularly in the robot's direction, and the other two walk diagonally. Finally, the right image presents the *Combined* scenario with three people showing different reactions to the presence of the robot: a "regular" navigation, which deals with the robot as a another human; a "curious" reaction, in which the human agent carefully approaches the robot; and a "threatening"

reaction, in which the human agent will insistently try to block the robot from passing.

B. Planners

In the previous social-navigation scenarios, we will compare the metrics obtained by three planners with different features:

- *DWB*. The basic ROS 2 navigation stack using the *DWB* local planner¹⁰ which is based primarily on the *Dynamic Window Approach* [27] to reach a goal efficiently without "social" constrains.
- *SCL*. The same system using a *Social Costmap Layer* (a custom ROS 2 version of the social navigation layers implemented in ROS 1¹¹). It includes a cost around the people according to the distance and orientation.
- *SFW*. A social local planner, developed by us, based on the *Dynamic Window Approach*.¹² It uses the *Social Force Model* as a human pose predictor and a trajectory scoring function which adds the social work as a new term. That means that in addition to distance from people, the velocity vectors of the robot and people are also considered.

C. Results

1) *Results for Regular Navigation Behavior*: First, we show the general metrics obtained by the planners in the *Passing* and *Crossing* scenarios. We have repeated the trajectory 10 times for each planner in both scenarios. Tables II and III show the average values of the metrics obtained.

We can draw some conclusions from these results. First, the *DWB* planner does not consider any social component, so its objective is to reach the goal as soon as possible without

¹⁰[Online]. Available: https://github.com/ros-planning/navigation2/tree/main/nav2_dwb_controller

¹¹[Online]. Available: https://github.com/robotics-upo/nav2_social_costmap_plugin

¹²[Online]. Available: https://github.com/robotics-upo/social_force_window_planner

TABLE II
 METRICS FOR THE Passing SCENARIO

Passing	DWB	SCL	SFW	units
time_to_reach_goal	25.5	26.6	35.7	s
path_length	15.2	15.7	15.2	m
cumulative_heading_changes	0.47	1.89	1.89	rad
avg_dist_to_closest_person	6.36	7.95	7.49	m
min_dist_to_people	0.40	0.90	0.55	m
intimate_space_intrusions	3.66	0	0	%
personal_space_intrusions	19.7	5.8	14.7	%
social+_space_intrusions	17.0	26.7	21.4	%
completed	100	100	100	%
min_dist_to_target	0.18	0.19	0.11	m
final_dist_to_target	0.18	0.19	0.11	m
robot_on_person_collisions	0	0	0	-
person_on_robot_collisions	0	0	0	-
time_not_moving	0.00	1.04	1.78	s
avg_robot_linear_speed	0.51	0.51	0.38	m/s
avg_robot_angular_speed	0.02	0.09	0.09	rad/s
avg_robot_acceleration	0.05	0.05	0.08	m/s ²
avg_robot_jerk	0.08	0.06	0.08	m/s ³
avg_pedestrian_velocity	0.47	0.44	0.34	m/s
avg_closest_pedestrian_velocity	0.57	0.51	0.39	m/s
social_force_on_agents	0.52	0.97	0.49	m/s ²
social_force_on_robot	0.52	0.92	0.49	m/s ²
obstacle_force_on_agents	8.14	0.13	0.81	m/s ²
obstacle_force_on_robot	0.00	0.00	0.00	m/s ²
social_work	1.04	1.89	0.98	m/s ²

TABLE III
 METRICS FOR THE Crossing SCENARIO

Crossing	DWB	SCL	SFW	units
time_to_reach_goal	10.1	14.6	14.0	s
path_length	4.9	5.7	4.9	m
cumulative_heading_changes	2.15	6.68	2.03	rad
avg_dist_to_closest_person	1.45	1.37	1.51	m
min_dist_to_people	0.23	0.21	0.49	m
intimate_space_intrusions	13.7	14.6	1.6	%
personal_space_intrusions	28.5	25.2	44.9	%
social+_space_intrusions	57.9	60.2	53.4	%
completed	100	100	100	%
min_dist_to_target	0.19	0.22	0.13	m
final_dist_to_target	0.19	0.22	0.13	m
robot_on_person_collisions	0	0	0	-
person_on_robot_collisions	0	0	0	-
time_not_moving	0.00	0.00	0.38	s
avg_robot_linear_speed	0.35	0.31	0.28	m/s
avg_robot_angular_speed	0.44	0.55	0.25	rad/s
avg_robot_acceleration	0.09	0.12	0.14	m/s ²
avg_robot_jerk	0.10	0.13	0.26	m/s ³
avg_pedestrian_velocity	0.46	0.36	0.36	m/s
avg_closest_pedestrian_velocity	0.27	0.31	0.23	m/s
social_force_on_agents	0.67	1.10	0.44	m/s ²
social_force_on_robot	0.67	1.10	0.44	m/s ²
obstacle_force_on_agents	2.08	6.07	2.08	m/s ²
obstacle_force_on_robot	0.00	0.00	0.00	m/s ²
social_work	1.33	2.20	0.88	m/s ²

colliding. Therefore it obtains the best performance in metrics like time to reach the goal, path length or the average robot velocities.

Secondly, the *SCL* adds an extra cost to the navigation costmap around the people. That provokes the robot to keep more distance from people. In the *Passing* scenario, that is reflected in metrics like the average distance to the closest person, minimum distance to people and the intrusions in the intimate and personal

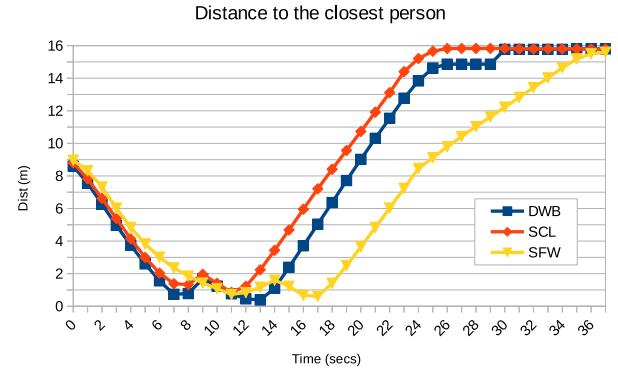


Fig. 5. Distance to the closest person for one trajectories of the *Passing* scenario.

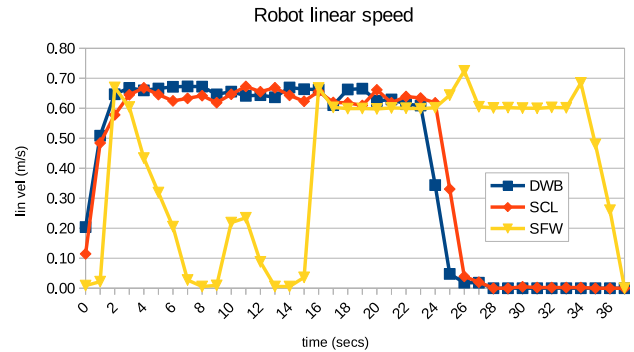


Fig. 6. Robot linear velocity for one trajectories of the *Passing* scenario.

spaces. However, we observe in the *Crossing* scenario that several people may come close to the robot at the same time, which provokes high costs all around the robot. In this situation, with high costs for most of the potential trajectories, the robot decides to move towards the goal, leading to higher intimate space intrusions, lower average distance to the closest person, and higher social force on agents than in the *Passing* scenario.

Finally, the *SFW* planner includes human-aware constraints. Specifically, it uses the social work as a term in its trajectory scoring function. That leads the robot to better preserve the comfort of surrounding people. It reaches the best performance in metrics like social work and social force on agents as could be expected. It also reaches the best results in the intrusions in the intimate space of people, and it obtains good results in the minimum distance to people or the average distance to the closest person. It also spends more time stopped (probably yielding).

The *HuNavSim* also provides the values of the metrics along the robot trajectory. For instance, Fig. 5 shows the distance to the closest person for each time step of one of the trajectories performed of the *Passing* scenario. We can observe the moments in which the robot passes the two pedestrians, and how the *DWB* is the planner which chooses to pass closer to them. The same situation is shown in Fig. 6 for the robot linear speed. In this case, we clearly see how the *SFW* planner decides to almost stop when very close to people in order to not disturb them.

In summary, planners including only distance constrains (like *SCL*) should expect to obtain good results in distance-based metrics. More advanced planners which also predict future people

TABLE IV
SOME METRICS FOR THE PLANNERS IN THE Combined SCENARIO WITH PEOPLE SHOWING DIFFERENT BEHAVIORS. THE ORDER OF THE SHOWN VALUES IS DWB PLANNER, SCL PLANNER AND SFW PLANNER

DWB / SCL / SFW	Regular	Curious	Threatening
min_dist_to_people	0.96 / 1.07 / 1.05	0.77 / 0.80 / 0.75	0.21 / 0.20 / 0.15
avg_dist_to_closest_person	11.42 / 11.31 / 12.85	3.56 / 3.99 / 8.05	3.55 / 3.54 / 3.29
intimate_space_intrusions	0.00 / 0.00 / 0.00	0.00 / 0.00 / 0	52.84 / 45.24 / 58.46
personal_space_intrusions	4.27 / 4.76 / 1.54	10.03 / 11.90 / 13.85	14.56 / 21.43 / 4.62
avg_pedestrian_velocity	0.22 / 0.21 / 0.13	0.59 / 0.56 / 0.46	0.66 / 0.67 / 0.60
social_force_on_robot	0.00 / 0.00 / 0.00	0.07 / 0.08 / 0.30	3.23 / 1.91 / 2.91
social_work	0.00 / 0.00 / 0.00	0.14 / 0.15 / 0.60	6.46 / 3.83 / 5.83

positions regarding velocities and directions (like *SFW*) should obtain better results in metrics based on social forces and social work, since they take into account people distances, velocities and directions. The planners can be better characterized thanks to having a broad set of metrics.

2) *Results for Different Navigation Behaviors:* In a realistic scenario, not only do people walk at different velocities and maintain different distances, but they also react very differently to the presence of an autonomous mobile robot. Some people will treat the robot like another pedestrian while others might feel curiosity, mistrust or even show harassing behavior. In most cases, these possible reactions are neither taken into account while developing robotic navigation algorithms nor are they considered in the current pedestrian simulators.

We evaluate here the results obtained according to the different behaviors of the three human agents in the *Combined* scenario. Table IV shows the results of the three planners for some interesting metrics related to the human behavior. For the three planners some general conclusions can be extracted: in the case of the “regular” human, we obtain the expected metrics as above. However, as we observe, the “curious” and “threatening” agents provoke totally different metrics since they abandon their regular navigation to approach the robot for a while. We also see the differences between a “curious” person and a “threatening” person. The latter is more intrusive, since the human agent tries to block the robot path and moves very close to the robot. In that situation, the *SFW* planner prefers the robot to spend more time almost fully stopped than the other planners to prevent disturbances and possible collisions. The robot navigation algorithms must be very careful since the risk of collision is high.

These results emphasize the complexity of performing robotic navigation experiments in real and uncontrolled scenarios. Additionally, computing significant metrics without considering the different reactions of the pedestrians can lead to wrong or unclear results. *HuNavSim* can help to develop and to benchmark navigation algorithms that consider these possible reactions.

VI. CONCLUSION AND FUTURE WORK

We have introduced a new open-source software library used to simulate human navigation behaviors, *HuNavSim*. The tool, programmed under the new ROS 2 framework, can be employed to control the human agents of different general robotics simulators. A wrapper for the use of the tool with the well-known Gazebo simulator is also introduced.

Moreover, it presents other novelties, such as a set of individualized human navigation behaviors directed by behavior trees. These behaviors are easily configurable and expandable due to the use of behavior trees. The tool also includes an extensive, flexible and extensible compilation of evaluation metrics for robot human-aware navigation taken from the literature.

The provided comparison of three different planners demonstrates that the tool is useful for the development of navigation algorithm and for the evaluation of them. The addition of different reactions to the presence of the robot also presents a novelty that helps to simulate more realistic navigation situations in spaces shared with humans.

Future work includes the extension of the set of metrics. The metrics from the *CrowdBot* [6] and others indicated in the work of Gao et al. [26], will be studied and included. The improvement of the body animations accompanying the agents movements in the Gazebo wrapper is being considered just as the development of wrappers for other simulators like Webots or Isaac Sim. Moreover, the augmentation of the set of individual navigation behaviors will be studied, as well as adding other human-navigation models besides the Social Force Model. Finally, on a technical level, we will work to update the software to use the ROS 2 Humble distribution (under development currently) and the latest version of the Behavior Tree library (v4).

REFERENCES

- [1] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, “Robot operating system 2: Design, architecture, and uses in the wild,” *Sci. Robot.*, vol. 7, no. 66, pp. 60–74, 2022, doi: [10.1126/scirobotics.abm6074](https://doi.org/10.1126/scirobotics.abm6074).
- [2] D. Helbing and P. Molnár, “Social force model for pedestrian dynamics,” *Phys. Rev. E., Stat. Phys., Plasmas, Fluids, Related Interdiscipl. Topics*, vol. 51, no. 5, pp. 4282–4286, 1995.
- [3] A. Aroor, S. L. Epstein, and R. Korpan, “Mengeros: A crowd simulation tool for autonomous robot navigation,” in *Proc. Fall Symp. Artif. Intell. Human-Robot Interact.*, 2017, pp. 123–125.
- [4] J. Van den Berg, S. J. Guy, M. Lin, and D. Manocha, “Reciprocal N-body collision avoidance,” in *Robotics Research*, C. Pradalier, R. Siegwart, and G. Hirzinger, Eds. Berlin, Germany: Springer, 2011, pp. 3–19.
- [5] S. Curtis and D. Manocha, “Pedestrian simulation using geometric reasoning in velocity space,” in *Pedestrian and Evacuation Dynamics*, U. Weidmann, U. Kirsch, and M. Schreckenberg, Eds. Cham, Switzerland: Springer, 2012, pp. 875–890.
- [6] F. Grzeskowiak et al., “Crowd against the machine: A simulation-based benchmark tool to evaluate and compare robot capabilities to navigate a human crowd,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 3879–3885.
- [7] N. Tsoi et al., “Sean 2.0: Formalizing and generating social situations for robot navigation,” *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 11047–11054, Oct. 2022.
- [8] N. Tsoi, M. Hussein, J. Espinoza, X. Ruiz, and M. Vázquez, “Sean: Social environment for autonomous navigation,” in *Proc. 8th Int. Conf. Hum.-Agent Interact.*, 2020, pp. 281–283.

- [9] A. Favier, P.-T. Singamaneni, and R. Alami, "Simulating intelligent human agents for intricate social robot navigation," in *Proc. RSS Workshop Social Robot Navigation*, Washington, United States, Jul. 2021.
- [10] O. Hauterville, C. Fernández, P.-T. Singamaneni, A. Favier, V. Matellán, and R. Alami, "Imhus: Intelligent multi-human simulator," *IEEE/RSJ IROS2022 Workshop: Artif. Intell. Social Robot. Interact. Humans Real World*, Kyoto, Japan, Oct. 2022. [Online]. Available: <https://hal.laas.fr/hal-03865622>
- [11] M. Colledanchise and P. Ögren, *Behavior Trees in Robotics and AI: An Introduction, Ser. Chapman & Hall/CRC Artificial Intelligence and Robotics Series*. Boca Raton, FL, USA: CRC Press, 2018.
- [12] P. Teja S and R. Alami, "Hateb-2: Reactive planning and decision making in human-robot co-navigation," in *Proc. IEEE 29th Int. Conf. Robot Hum. Interactive Commun.*, 2020, pp. 179–186.
- [13] A. Biswas, A. Wang, G. Silvera, A. Steinfeld, and H. Admoni, "Socnavbench: A grounded simulation testing framework for evaluating social navigation," *ACM Trans. Hum.-Robot Interact.*, vol. 11, no. 3, Jul. 2022, Art. no. 26, doi: [10.1145/3476413](https://doi.org/10.1145/3476413).
- [14] L. Kästner et al., "Arena-rotnav: Towards deployment of deep-reinforcement-learning-based obstacle avoidance into conventional autonomous navigation systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 6456–6463.
- [15] L. Kästner et al., "Arena-bench: A benchmarking suite for obstacle avoidance approaches in highly dynamic environments," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 9477–9484, Oct. 2022.
- [16] L. Kästner et al., "Arena-rotnav 2.0: A development and benchmarking platform for robot navigation in highly dynamic environments," 2023, *arXiv:2302.10023*.
- [17] J. Holtz and J. Biswas, "Socialgym: A framework for benchmarking social robot navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 11246–11252.
- [18] Z. Sprague, R. Chandra, J. Holtz, and J. Biswas, "Socialgym 2.0: Simulator for multi-agent social robot navigation in shared human spaces," 2023, *arXiv:2303.05584*.
- [19] D. Perille, A. Truong, X. Xiao, and P. Stone, "Benchmarking metric ground navigation," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot.*, 2020, pp. 116–121.
- [20] E. Heiden, L. Palmieri, L. Bruns, K. O. Arras, G. S. Sukhatme, and S. Koenig, "Bench-MR: A motion planning benchmark for wheeled mobile robots," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 4536–4543, Jul. 2021. [Online]. Available: <https://ieeexplore.ieee.org/document/9387068>
- [21] M. Moussaïd, D. Helbing, S. Garnier, A. Johansson, M. Combe, and G. Theraulaz, "Experimental study of the behavioural mechanisms underlying self-organization in human crowds," *Proc. Biol. Sci.*, vol. 276, no. 1668, pp. 2755–2762, Aug. 2009.
- [22] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz, "The walking behaviour of pedestrian social groups and its impact on crowd dynamics," *PLoS One*, vol. 5, no. 4, 2010, Art. no. e10047.
- [23] N. Perez-Higueras, F. Caballero, and L. Merino, "Teaching robot navigation behaviors to optimal RRT planners," *Int. J. Social Robot.*, vol. 10, no. 2, pp. 235–249, 2018.
- [24] E. T. Hall, *The Hidden Dimension*, ser. A Doubleday Anchor book. Garden City, N.Y.: Anchor Books ed., 1969.
- [25] K. Katyal et al., "Learning a group-aware policy for robot navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 11328–11335.
- [26] Y. Gao and C.-M. Huang, "Evaluation of socially-aware robot navigation," *Front. Robot. AI*, vol. 8, 2022, Art. no. 721317. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/frobt.2021.721317>
- [27] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Automat. Mag.*, vol. 4, no. 1, pp. 23–33, Mar. 1997.