IMHuS: Intelligent Multi-Human Simulator

Olivier Hauterville[1] Camino Fernández[1,2] Phani Teja Singamaneni[1] Anthony Favier[1,3] Vicente Matellán[1,2] Rachid Alami[1,3]

***Abstract*— Simulation is a basic tool for testing robots' behavior that must include humans when dealing with social robotics. The goal of Human-Robot-Interaction simulation is not only to test different techniques but also to provide the replicability needed to study which metrics are best suited to measure social interaction between robots and humans. IMHuS offers a system in which humans can be choreographed to create high-level social behaviors, such as taking an elevator in scenarios defined to place a robot and measure its performance. In addition, the system can be easily modified to define new actions and metrics so that developers can create new scenarios to test and measure social HRI.**

***Index Terms*— HRI, simulation, social robotics**

I. Introduction

One of the basic requirements of science is the replicability of the experiments designed to validate any proposed model, theory, or hypothesis. However, when the research topic involves interaction with people, as in social robotics, this replicability is difficult to achieve. Even detailed descriptions such as the one in [1] are really difficult to replicate. Furthermore, small variations in people's behavior have a significant impact on the robot's behavior. Software simulators cope with this problem by providing the ability to replicate the exact same situation repeatedly. These simulators should be able to provide "simulated" human behavior letting researchers test their socially acceptable robotic behaviors, but currently offer only limited capabilities in this area.

The goal of the tool described in this paper, IMHuS (Intelligent Multi Human Simulator), is to provide a means to generate scenarios in which predefined social interactions of individual simulated humans and groups of reactive humans can be used to assess the social navigation performance of the robot behavior under evaluation. In the remainder of the paper, we will refer to this robot as *tested robot*. The aim is to use IMHuS for benchmarking human-robot interaction behaviors thanks to its flexible design that allows adding new actions to create new scenarios in a simple way.

II. State of the art

There are several simulators commonly accepted in the robotics research community such as Gazebo [2], Unity [3], Coppelia (V-REP)[4], Morse [5], Unreal, etc. These simulators have proven to be very useful for testing basic service robotics skills, such as obstacle avoidance, localization, global navigation, etc. However, they do not provide simulated humans to test social navigation. These simulated human should be aware of the presence of the robot and offer some basic social navigation capabilities.

The problem of human social navigation is also an active research question in the crowd simulation research community. Crowd simulation has applications in many fields, from transportation to psychology, video game design, or digital effects for movies. But they are generally more concerned with the management of large crowds than with the intelligent behaviors of individuals. Unfortunately, classical crowd simulators, such as Continuum [6], Pedsim [7], or Menge do not simulate robots. Some adaptations have been proposed to include robots in these crowd simulators. The best known is Pedsim-ROS [8], a set of packages developed at the University of Freiburg for 2D pedestrian simulation integrated with ROS. However, these simulators usually provide simple behaviors for simulated humans. For example, Pedsim-ROS only provides local collision avoidance based on the Social-Force Model (SFM) [9].

MengeROS[10] is a 2-D crowd simulator that integrates multiple robots and a crowd. Menge's crowd scenario includes the static map (walls and obstacles), the simulated people and their initial locations, along with a navigation algorithm (A^* or potential fields) and a collision-avoidance one (SFM, ORCA, or PedVO) for all the simulated pedestrians. Some severe constraints in Menge are that all robots have a circular shape and that laser scanning and robots' actions are noise-free.

Another recent simulator, SEAN [11] (Social Environment for Autonomous Navigation), is a high-fidelity Unity-based visual simulator that provides only two realistic scenes. The Trial Runner tool included in the toolkit performs the simulations by setting up the scene and assigning start and goal positions for human agents who use Unity's built-in path planning algorithm to generate their routes. However, no tools are provided to define the social behavior of the human avatars.

OCIALGYM [12] is a lightweight 2D simulation environment where social navigation is modeled as an action selection problem, and the optimal action is selected from a set of four discrete sub-policies: Halt, GoAlone, Follow and Pass. In SOCIALGYM, each simulated human moves back and forth between a starting point and a goal pose by computing a global path of intermediate nodes between them and using a separate local planner (Pedsim-ROS social force model) to handle dynamic obstacles. It also includes a

[1] RIS group - LAAS (CNRS), Colonel Roche, 31400 Toulouse (France) {Olivier.Hauterville, phani-teja.singamaneni, anthony.favier, Rachid.Alami}@laas.fr,

[2]Robotics Group, Universidad de León, 24007 León (Spain),{cferl,vmato}@unileon.es

[3]ANITI, Univeristé Fédérale Toulouse Midi-Pyrénées, B612, Rue Tarfaya, 31400 Toulouse (France)

"Navigation Module" responsible for planning the obstacle-free path from the starting point to the goal. Therefore, SOCIALGYM only provides a combined behaviour using the above sub-policies, but does not provide group behaviors, neither social interaction.

Regarding benchmarking, currently most proposals to compare social navigation algorithms are based on datasets. Some of them are designed to test perception, such as SRIN (Social Robot Indoor Navigation) [13] which contains images taken by medium size robots. SocNav1 [14] includes 9,280 scenarios containing locations of humans and a robot in an 8x8 area and the subjective level of discomfort generated by the robot's presence in that scenario. Although datasets are essential for machine learning training algorithms, their use as benchmarking tools is not justified, as they are intrinsically static. Therefore, dynamic simulators should be part of the benchmarking process.

SocNavBench[15] is an intermediate testbed that mixes a dataset and some simulation. It comprises a photo-realistic renderer, a set of navigation scenarios based on real-world pedestrian data, and a set of metrics to characterize the performance of robot navigation algorithms. Instead of simulating behavior, SocNavBench derives real-world pedestrian trajectories from real behaviour datasets. It applies real-world textures on pedestrians and the environment to overcome the inability to obtain sensor inputs for these datasets. Although it is an improvement over mere images, simulated humans have fixed, non-reactive behavior. They can only exhibit pre-recorded behavior and only existing behaviors can be reproduced. The creation of new scenes requires recording new real data.

DeepSocNav [16] proposes a similar approach, starting from the ETH BIWI Walking Pedestrians dataset (RGB images from a bird's eye view of two real scenarios). It recreates the scenes of this dataset in Unity extracting depth images to create first-person views of the trajectories and with an added ability to include new agents. In this case, these systems can go beyond pre-recorded behaviors, but no social behaviors can be defined for the new agents.

Liu et al. [17] have also proposed interactive pedestrian simulation in iGibson. They decompose the problem into a "Global Planning" module based on A^* to compute coarse-grained waypoints to guide pedestrians towards their goals, and a "Local Planning" module to control pedestrian positions at each time step. The main difference to IMHuS is that iGibson compute these trajectories for each of the simulated humans, focusing on learning socially-aware individual navigation behaviors. Our proposal is not oriented towards learning individual social behaviors, we want to provide a tool for simulating the behavior of groups of humans that can be used to test the social skills of robots.

The closest tool to the proposed system is the Intelligent Human Simulator (InHuS) [18], which proposes a generic architecture for simulating intelligent human agents. It provides a complex "Human Behavior Model" module capable of defining goals, managing relations with other agents and building perception from the data received from the simulator. So, it can be considered as a complete cognitive architecture that could be used to control simulated humans or robots. It has been tested for social navigation, but it only can control a single simulated human, which uses CoHAN[19] as the robot controller, but enhanced with two "attitudes": *Harass* and *StopAndLook*.

Another proposal with similar goals is the one described in [20]. Their goal is also to provide a tool for testing autonomous mobile social robots, but the tool is not freely available, so it is difficult to assess its scope. Their tool uses pre-stored pedestrian patterns for a single human interacting with the robot, not for groups of humans, and also just for a single interaction.

Finally, there are other approaches for generating the behavior of simulated entities. For instance, in the classical planning community, there are simulators as PDSim[21], where the behavior of objects in planning problems can be defined using PDDL. These "animations" may include moving an object onto another object, letting an agent follow a path between two points on a map, etc.

In summary, existing simulation tools that integrate robots and humans do not allow simulating realistic social behaviors of the simulated people. The only available alternative is the use of teleoperated avatars or basic behaviors. The need for simulation tools that provide high-level primitives to define the behavior of simulated humans has been identified, and some commercial products are starting to appear, for instance, PedSim Pro V1.2 has incorporated the "queuing behaviour".

The goal of IMHuS, the tool described in this paper, is to provide an open-source toolkit for defining high-level reactive simulated humans with the ability to show the behavior of social groups, but using realistic standard robotics simulators that allow researchers to use models of their real robots, both for debugging their algorithms and for benchmarking and repeatability.

III. IMHUS DESIGN

The goal of IMHuS is to standardize the validation of the autonomous robot behavior in the presence of people, allowing researchers to design repeatable human social situations. To do this, both individuals and groups must be choreographed. The *tested robot* would not only have to deal with avoiding collision with humans, but also detect social situations and act accordingly. And during the course of the choreography, humans should also exhibit adaptive behavior.

Taking these requirements into account, the static design of the IMHuS module can be explained attending to the layers that interact with it, that is, the application layer and the ROS/Simulator layer:

Application layer This layer takes as input an XML configuration file needed to use IMHuS. The main components of IMHus are defined in this file: map, agents, tasks and scenarios. The map includes the locations to be used during the choreography and the static objects. The description of the agents includes the name and initial position of all humans and robots choreographed, and also the name and composition of the groups that will appear in any step of the scenario. Generic tasks are described with no specific subject assigned so that they can be reused by several agents. Last, scenarios describe the steps of the choreography. Each step includes a set of tasks assigned to a particular agent or group.

IMHuS layer The tool interprets the information contained in the XML configuration file to represent the tasks as actions and the humans/robots as agents. When one scenario is run, a dynamic command is created for every element of the steps contained in that scenario. A command combines an action with the agent/s that will perform it. The commands concerning each agent are executed in a separate thread inside a step and the step finishes when all the threads are done. In order for the *tested robot* to communicate with the agents, there is an asynchronous step included in each scenario. This way, at any point in time, the *tested robot* can ask for something, for example, to press the elevator button. This action would be done through the communication module and would be answered by the agents in the simulation in this asynchronous step by means of a respond action. Agents can be designed to respond to a request from the *tested robot* only in specific circumstances, for example, when they are in an idle state.

ROS/Simulator layer This layer is used by IMHuS to place the agents in the simulation and request the trajectories to move them, as well as to communicate with the *tested robot*. The ORCA algorithm [22] is used to make the agents move while avoiding the dynamic movement of the rest of the agents, including the *tested robot*. IMHuS takes the trajectories provided by move-base as a starting point, and only asks for re-planing when ORCA causes the deviation from the initial trajectory to exceed certain thresholds.

Interface of the *tested robot* The way to include the software of a robot in the simulator for its social behavior to be tested would be through the simulator, Gazebo for now. Outside the simulator, the only communication would be through the asynchronous action thanks to which the robot can send a request action to be answered by a respond action of one agent of the simulation.

The dynamic explanation of IMHuS starts with the scenarios defined in the XML of the choreography where the following types of *basic actions* are assigned to the *agents* (simulated robots or simulated people):

Navigation actions Those actions modify the global pose of the agents in the environment. Typical actions in this group are $GoTo$ and $Wait$.

Turning actions : Actions related to the facing of the agents, such as $LookAt$ and $TurnAngle$ depending on the way the action is specified.

Grouping actions : These actions manage the creation and dismantling of groups.

Attitude actions : Actions modeling the high level behavior of the agent. For instance, a simulated human can be ordered to *harass* the robot.

Synchronous actions : Actions related to the environment to be accomplished by an agent during one specific step of the simulation. They have been standardized as $publish$ and $subscribe$.

Asynchronous actions : Actions related to the environment and not associated with a specific step in the timeline of the simulation. They have been standardized as $request$ and $respond$.

These actions are included in the sequential steps that make up a scenario. The regular steps are synchronous, but there is an asynchronous step for interacting with the *tested robot*:

- A scenario is composed of a set of regular steps and a singular asynchronous step.
- Steps are made up of different actions like navigation, grouping, etc.
- Every action of a step is executed *at the same time*, meaning that the actions are executed in simulated parallelism.
- One step ends when all its actions have finished.
- The asynchronous step runs in parallel to the execution of the synchronous steps.
- The scenario ends when the last regular step ends.

The current prototype has been implemented as a new version of the InHus tool [23]. IMHus connects to the Gazebo simulator to obtain the information about the world and uses ORCA-COHAN [18] to generate the global plan for each agent, updating the positions of each agent in the next step of the simulation accordingly. The main difference with the previous version is that IMHus is able to control several simulated humans, while the previous one was only able to manage one. Therefore, IMHus is suitable for benchmarking simulated robots when dealing with simulated humans exhibiting social behaviors such as moving as part of a group, negotiating the path with groups, or with humans walking hand in hand, or standing in from of a showcase or elevator, etc.

IV. USE CASE

An elevator scenario, inspired in the SciRoc competition, has been used to illustrate how IMHuS works, and also to explain the communication between the *tested robot* and the IMHuS agents (video available at this link). In this scenario six human agents and the *tested robot*, whose goal is to go to the second floor and return to the first floor, are defined. To do so, it will go to the second floor with three humans and then ask a human agent to press the elevator button to return to the first floor. Figure 1 shows the challenging initial situation in which the *tested robot* approaches the elevator, two humans exit the elevator while three enter, and human 3 walks in the middle of the rest.

As can be seen in the video, thanks to the use of CoHAN, the *tested robot* is able to enter the elevator and exit at the second floor without colliding with the surrounding IMHuS

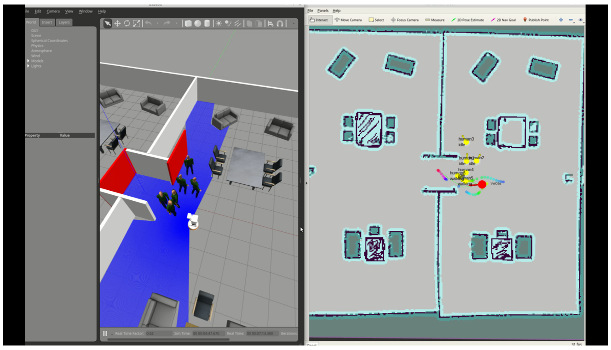

Fig. 1. Initial situation.

agents. Figure 2 shows the moment when the *tested robot* reaches the second floor.

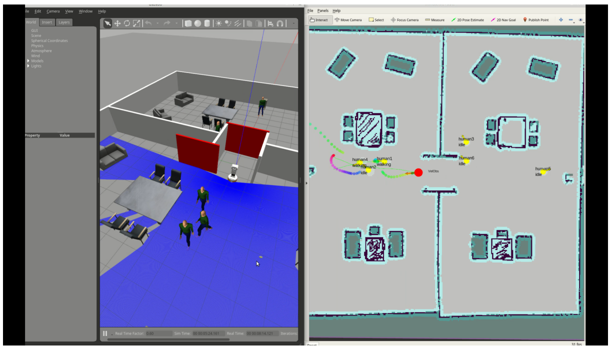

Fig. 2. Reaching the second floor.

Once on the second floor, the *tested robot* wants to return to the first floor. To do so, an agent must respond to the *tested robot*'s request to press the elevator button. At this point, two different things can happen. If there is no human close enough to the *tested robot*, or there is a human but it is not in an idle state, no one will respond to the request and it will be dropped. In the video, human 2 is close enough and in an idle state, so he responds and accepts the request, as Figure 3 shows.

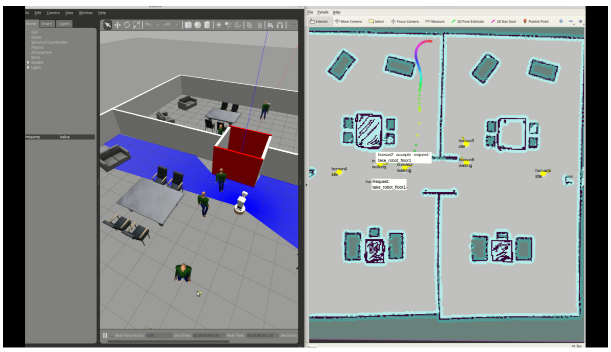

Fig. 3. Request accepted by human2.

Figure 4 shows that the *tested robot* returned to the first floor without collision. The simulation choreographed by IMHuS defines the behavior of the six human agents to provide a challenging environment for the *tested robot*, which is always surrounded by more than one human, both inside and outside the elevator. These human agents are reactive both to the presence of other agents and to that of the *tested robot*.

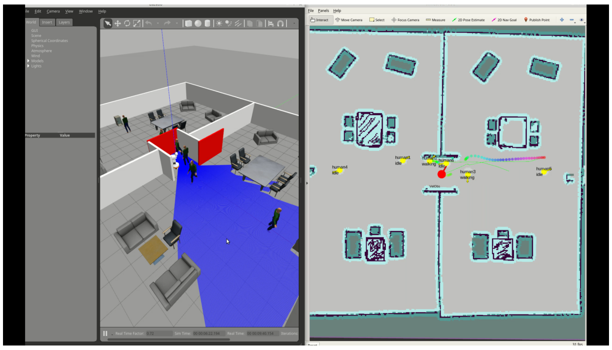

Fig. 4. Final situation. *Tested robot* back in first floor.

V. Conclusions and Future Work

The proposed tool, IMHuS, offers the possibility to create realistic and challenging simulated environments to evaluate the social navigation behaviour of *tested robots*. Different scenarios can be easily created using an XML configuration file in which social situations can be defined to measure the behavior of the *tested robot* in a replicable way. Furthermore, human agents are programmed to respond to interactions related to the particular situation of each scenario and their communication with the *tested robot*.

The IMHuS code is available as Open Source in the IMHuS repository. The current version has been implemented for Gazebo simulator, but the design presented in section III can be easily migrated to other simulators, such as MORSE or Unity. The tool could be used for benchmarking competitions such as SciRoc or RoboCup as a previous step for the teams before getting to the physical robot challenges. To create a new scenario with IMHuS, all that is needed is a map and the configuration file to choreograph the human agents. The software has been designed to easily support the addition of both new tasks to be performed by humans, and new interactions between humans and the *tested robot*.

One of the ongoing developments is the automatic logging of different variables, such as the distance between the robot and the nearest agent, the estimated time-to-collision, etc. These metrics can be obtained manually in the current version, but the goal is to provide these metrics automatically. Another line of work is the extension of basic actions, especially towards the social behavior of groups of agents. IMHuS design also supports the chreographication of robots in the same way as humans.

Finally, CoHAN has been used tested as navigation planner for the simulated humans, which let us to challenge CoHAN in various human-robot interaction scenarios. From this, we plan to identify the areas of improvement for human-aware navigation planners, CoHAN in particular, and provide a benchmark for testing these planners.

ACKNOWLEDGMENT

This work has been partially funded by the Kingdom of Spain under grant PID2021-126592OB-C21.

REFERENCES

[1] Y. Kobayashi, T. Sugimoto, K. Tanaka, F. J. A. G. a. C. H. K. Yuki Shimomura, H. Yabushita, and T. Toda, "Robot navigation based on predicting of human interaction and its reproducible evaluation in a densely crowded environment," *International Journal of Social Robotics*, pp. 373–387, 2022.

[2] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, pp. 2149–2154 vol.3.

[3] A. Juliani, V.-P. Berges, E. Teng, A. Cohen, J. Harper, C. Elion, C. Goy, Y. Gao, H. Henry, M. Mattar, and D. Lange, "Unity: A general platform for intelligent agents," 2020.

[4] E. Rohmer, S. P. Singh, and M. Freese, "CoppeliaSim (formerly V-REP): a versatile and scalable robot simulation framework," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013)*, 2013, p. 1321–1326.

[5] S. Lemaignan, G. Echeverria, M. Karg, J. Mainprice, A. Kirsch, and R. Alami, "Human-robot interaction in the MORSE simulator," in *Human-Robot Interaction Conference (HRI'2012)*, 2012.

[6] A. Treuille, S. Cooper, and Z. Popović, "Continuum crowds," *ACM Trans. Graph.*, vol. 25, no. 3, p. 1160–1168, jul 2006. [Online]. Available: https://doi.org/10.1145/1141911.1142008

[7] C. Gloor, "PedSim: Pedestrian crowd simulation," pedsim.net.

[8] B. Okal and T. Linder, "PedSim: Pedestrian crowd simulation," github.com/srl-freiburg/pedsim_ros.

[9] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[10] A. Aroor, S. L. Epstein, and R. Korpan, "MengeROS: a crowd simulation tool for autonomous robot navigation," *ArXiv*, vol. abs/1801.08823, 2017.

[11] N. Tsoi, M. Hussein, J. Espinoza, X. Ruiz, and M. Vázquez, "SEAN: Social environment for autonomous navigation," in *Proceedings of the 8th International Conference on Human-Agent Interaction*, ser. HAI '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 281–283. [Online]. Available: https://doi.org/10.1145/3406499.3418760

[12] J. Holtz and J. Biswas, "SOCIALGYM: A framework for benchmarking social robot navigation," 2021.

[13] A. B. R. Kamal M O, "SRIN: A new dataset for social robot indoor navigation." *Global Journal of Engineering Sciences*, vol. 4, no. 5, 2020.

[14] L. J. Manso, P. Nuñez, L. V. Calderita, D. R. Faria, and P. Bachiller, "SocNav1: A dataset to benchmark and learn social navigation conventions," *Data*, vol. 5, no. 1, 2020. [Online]. Available: https://www.mdpi.com/2306-5729/5/1/7

[15] A. Biswas, A. Wang, G. Silvera, A. Steinfeld, and H. Admoni, "SocNavBench: A grounded simulation testing framework for evaluating social navigation," *CoRR*, vol. abs/2103.00047, 2021. [Online]. Available: https://arxiv.org/abs/2103.00047

[16] J. P. de Vicente and A. Soto, "DeepSocNav: Social navigation by imitating human behaviors," in *RSS Workshop on Social Robot Navigation*, 2021.

[17] Y. Liu, S. Li, C. Li, C. Perez-D'Arpino, and S. Savarese, "Interactive pedestrian simulation in iGibson," in *RSS 2021 - Workshop on Social Robot Navigation*, Jul 2021.

[18] A. Favier, P.-T. Singamaneni, and R. Alami, "Challenging Human-Aware Robot Navigation with an Intelligent Human Simulation System," June 2022, working paper or preprint. [Online]. Available: https://hal.laas.fr/hal-03684245

[19] R. A. Phani-Teja Singamaneni, Anthony Favier, "Human-aware navigation planner for diverse human-robot interaction contexts," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2021.

[20] Y. Kaneshige, S. Satake, T. Kanda, and M. Imai, "How to overcome the difficulties in programming and debugging mobile social robots?" in *Human-Robot Interaction Conference (HRI'2021)*, 2022, https://doi.org/10.1145/3434073.3444674.

[21] E. D. Pellegrin and R. P. Petrick, "PDSim: Planning domain simulation with the unity game engine," 2021, https://icaps21.icaps-conference.org/demos/demos/387.pdf.

[22] J. V. D. Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," *Robotics research*, p. 3–19, 2011.

[23] R. A. Anthony Favier, Phani-Teja Singamaneni, "Simulating intelligent human agents for intricate social robot navigation," in *RSS 2021-Workshop on Social Robot Navigation*, Jul 2021.