Towards an Abstract Lightweight Multi-robot ROS Simulator for Rapid Experimentation*

Laurence Roberts-Elliott[1][0000−0001−9877−0553], Gautham P. Das[1][0000−0001−5351−9533], and Alan G. Millard[2][0000−0002−4424−5953]

[1] Lincoln Agri-Robotics, University of Lincoln, United Kingdom
{lrobertselliott,gdas}@lincoln.ac.uk
[2] Department of Computer Science, University of York, United Kingdom,
alan.millard@york.ac.uk

Abstract. Modern robot simulators are commonly highly complex, offering 3D graphics, and simulation of physics, sensors, and actuators. The computational complexity of simulating large multi-robot systems in these simulators can be prohibitively high. To achieve faster-than-realtime simulation of a multi-robot system for rapid experimentation, we present 'move_base_abstract', a ROS package providing a high-level abstraction of robot navigation as a "drop-in" replacement for the standard 'move_base' navigation, and a bespoke integrated minimal simulator. This bespoke simulator is compatible with ROS and strips the simulation of robots down to the representation of robot poses in 2D space, control of robots via navigation goals, and control of simulation time over ROS topic messages. Replication of an existing MRS simulated study using 'move_base_abstract' executed 2.87 times faster than the real-time that was simulated in the study, and analysis of the results of this replication shows room for further optimisations.

Keywords: Abstract Simulation · Multi-Robot Systems Simulation · Robot Navigation.

1 Background

Most popular robot simulators, such as Gazebo [4] and Webots[7], can boast 3D graphics, and simulation of physics, sensors, and actuators. This detailed simulation is necessary for many applications, but it has performance costs that can make it difficult for simulations of multi-robot systems to run at the pace of real-time. Some applications, however, may not require such complexity of simulation. For many MRSs (Multi-Robot Systems), representation of time, robot poses in a coordinate system, and the ability for robots to navigate in this space, is sufficient for initial broad exploration or comparison of algorithms and methods [2,1]. Specific behaviours that robots may engage in once they reach allocated tasks may be abstracted to simply having a robot wait for the expected task duration. In some cases, they may be implemented ad-hoc by the simulator's user,

* This work was supported by the UKRI's E3 fund via Lincoln Agri-Robotics.

as more niche robot tasks are less likely to have a readily available implementation. While this high level of abstraction may not present as reliable an analogue of a real-world MRS as a more complex simulation might, its simplicity enables experimentation at a pace unattainable within more realistic simulators, especially when the complexity of simulations often scales poorly with increasing numbers of robots. Abstract simulators can enable researchers of MRSs to discover potentially more optimal solutions for their applications, the best of which can be simulated more accurately in a realistic simulator to increase confidence in these methods before time-consuming implementation in a real-world MRS.

We present 'move_base_abstract' (MBA), a Free and Open-Source Software (FOSS) Robot Operating System (ROS) compatible "drop-in" replacement for the standard move_base navigation ROS package [5], with a bespoke built-in minimalist simulator aimed at maximising performance, but also compatible with the 'stage_ros' and 'Gazebo' simulators, for users requiring more detailed simulation. It can be accessed at GitHub.com/laurencejbelliott/move_base_abstract.

2 Methodology

2.1 move_base_abstract (MBA)

MBA replaces the 'move_base' navigation stack with a high-level abstraction of robot navigation and provides a bespoke integrated minimalist simulator. Upon receiving a navigation goal, this abstracted navigation system essentially calculates the time that would be required for the robot to reach the goal moving in a direct line, multiplying Euclidean distance by a given constant speed. It then waits for the calculated time to elapse in ROS time, and finally updates the robot's pose to the pose specified in the navigation goal. The abstracted navigation system is interfaced as a 'ROS Action' compatible with the commonly used 'move_base' ROS Action, allowing for sending, monitoring, and cancelling goals in a standardised manner. The bespoke simulator is designed to leverage standard ROS utilities, such as 'map_server' [5], 'RViz' [3], and control of ROS time. MBA can also publish to topics used to update robot poses in 'stage_ros' and 'Gazebo', if the user needs a more full-featured simulator.

2.2 Replication of Multi-Robot Soil Compaction Mapping Trials

In a previous work, 'Agent-Based Simulation of Multi-Robot Soil Compaction Mapping' [8], we developed a simulation of a MRS for mapping soil compaction in an open field using the Mesa agent-based simulation Python library [6]. Simulated trials assessing the performance of different configurations of a multi-robot soil compaction mapping system were previously conducted in this Mesa-based simulation, and we replicate these in a ROS compatible simulation that uses MBA to vastly reduce the execution time of robot navigation. Currently the 'stage_ros' simulator is used in conjunction with the abstracted navigation of MBA, but future soil mapping MRS work will also use MBA's built-in simulator, which we expect will result in further reduction of computational complexity.

In the ROS version of the simulation, we re-ran the simulated trials at 3 speeds: 1x, 10x, and 15x, to measure an over-estimation of time elapsed during execution of complex algorithms in the MRS's coordinator. Simulation speed was reduced to 1x realtime while processes such as task allocation and task creation ran to somewhat account for this over-estimation. To measure the similarity of results between different runs of these simulated trials, the cosine similarity was calculated between a vector comprised of the mean performance metrics from one run of the simulated, and a vector of the same from another run, as shown in Table 1.

3 Results and Analysis

Table 1. Mean of cosine similarities calculated across metrics recorded in pairs of simulated multi-robot soil mapping studies.

Name of 1st Set of Trials	Name of 2nd Set of Trials	Mean Cosine Similarity
TAROS 22 Mesa trials	TAROS 23 stage_ros trials - 10x sim. speed	0.969960079
TAROS 23 stage_ros trials - 1x sim. speed	TAROS 23 stage_ros trials - 10x sim. speed	0.992865229
TAROS 23 stage_ros trials - 1x sim. speed	TAROS 23 stage_ros trials - 15x sim. speed	0.869587149

Cosine similarity can range from -1 to 1, with values closer to 1 indicating greater similarity. As seen in Table 1, a relatively high cosine similarity of 0.969960079 is measured when comparing the performance metrics from the Mesa trials with the those from the ROS trials run at 10x speed. This shows that the the Mesa simulation trials can be replicated in the ROS based simulation with very similar results. The small amount of dissimilarity may be explained by the assumptions made by the Mesa sim, that are not made in the ROS simulation. These being that Mesa operates in discrete integer steps of time (seconds in our case). Continuous processes run in its 'step' loop, called once every step. This means that complex computation that could take several seconds on modern PC hardware always takes 1 second in Mesa's time. Another assumption made in the Mesa sim is that A* pathfinding cost for movement between cells in the discrete grid representation of the environment is the same for movement to a neighbouring cell diagonally or orthogonally. The mean cosine similarity appears to decrease as the speed at which simulation progress time is increased. The results remain highly similar to 1x speed at 10x speed, but are relatively dissimilar at 15x speed. At 10x speed, the ROS simulation trials executed 2.87x faster than realtime overall. At 15x speed, this increased to 14.18x times faster than realtime, but the results in some metrics were significantly different to those measured at 1x speed.

4 Conclusions and Future Work

At 10x speed, the results of the stage_ros simulation trials are very similar to those seen when running the sim at wall time. It may be necessary to further optimise the simulator and MRS coordinator's more complex Python code, or port some of the more complex nodes to C++ (starting with the coordinator). This could help to attain results identical to those seen when running the simulation with simulation time progressing at a constant speed, when running the simulation with simulation time progressing faster than wall time, and attain reasonably accurate results running the sim at speeds beyond 10x. It is expected that running simulated trials on a faster CPU could enable faster simulated time speeds, and this may be investigated in future work. This simulation, and the bespoke simulator, provide a closer analogue to a real-world multi-robot system than Mesa, both by making fewer assumptions, and through their use of the ROS middleware. This should require minimal changes to the software to enable it to coordinate real robots with ROS.

References

1. Choudhury, S., Gupta, J.K., Kochenderfer, M.J., Sadigh, D., Bohg, J.: Dynamic multi-robot task allocation under uncertainty and temporal constraints. Autonomous Robots **46**(1), 231–247 (Jan 2022). https://doi.org/10.1007/s10514-021-10022-9, `https://doi.org/10.1007/s10514-021-10022-9`
2. Das, G., Cielniak, G., From, P., Hanheide, M.: Discrete Event Simulations for Scalability Analysis of Robotic In-Field Logistics in Agriculture – A Case Study. Brisbane (May 2018), `https://eprints.lincoln.ac.uk/id/eprint/32170/`
3. Kam, H.R., Lee, S.H., Park, T., Kim, C.H.: RViz: a toolkit for real domain data visualization. Telecommunications Systems **60**(2), 337–345 (Oct 2015). https://doi.org/10.1007/s11235-015-0034-5, `https://doi.org/10.1007/s11235-015-0034-5`
4. Koenig, N., Howard, A.: 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, pp. 2149–2154 vol.3 (Sep 2004). https://doi.org/10.1109/IROS.2004.1389727
5. Marder-Eppstein, E., Berger, E., Foote, T., Gerkey, B., Konolige, K.: The Office Marathon: Robust navigation in an indoor office environment. In: 2010 IEEE International Conference on Robotics and Automation. pp. 300–307. IEEE, Piscataway, New Jersey, USA (May 2010). https://doi.org/10.1109/ROBOT.2010.5509725, iSSN: 1050-4729
6. Masad, D., Kazil, J.: Mesa: An Agent-Based Modeling Framework. pp. 51–58 (Jan 2015). https://doi.org/10.25080/Majora-7b98e3ed-009
7. Michel, O.: Cyberbotics Ltd. Webots™: Professional Mobile Robot Simulation. International Journal of Advanced Robotic Systems **1**(1), 5 (Mar 2004). https://doi.org/10.5772/5618, `https://doi.org/10.5772/5618`, publisher: SAGE Publications
8. Roberts-Elliott, L., Millard, A.G., Das, G.P.: Agent-Based Simulation of Multi-Robot Soil Compaction Mapping. In: To appear in Lecture Notes in Computer Science. p. 15. Springer, Oxford, UK (Sep 2022)