Robotics simulation – A comparison of two state-of-the-art solutions

Maximilian Zwingel[1*], Christopher May[1], Matthias Kalenberg[1], Jörg Franke[1]

[1]Institute for Factory Automation and Production Systems (FAPS), Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany; **maximilian.zwingel@faps.fau.de*

Abstract. Simulation plays a significant role in development and deployment of robotics systems. Depending on the planned tasks, different functionalities shift into focus. For mobile robotics systems using the Robot Operating System (ROS), Gazebo has been the goto solution for simulating behavior, sensors, and interaction with environments for many years. With the availability of numerous novel solutions, a comparison of the different strengths from a use-case perspective becomes necessary.

In our paper, we compare Gazebo and NVIDIAs Isaac Sim in the context of ROS-based autonomous mobile robots. Special attention is given to the integration of the simulators in the ROS-environment and their ability in regards to sensor simulation. This paper aims to provide a comparison between different features in Gazebo and NVIDIA Isaac Sim for simulating mobile robots and sensor systems.

Introduction

Robotic systems play a significant role in addressing the challenges of future production. Tasks like handling, separation, identifikation and transportation need to be automated to enable sustainable productivity in globalized markets. Simulation plays a significant role in development and deployment of robotics systems to reduce the risks of misfitting and delays in ramp up. In light of these developments, different companies from tech business sectors have recently begun to focus on robotics as well as interlocked technologies to challenge existing solutions in the field and to open up new markets. Especially in the area of simulation of mobile robotics new frameworks with a focus on photo-realistic presentation emerge.

NVIDIA is well known for its graphics processors, but is also pushing towards AI as well as autonomous vehicles and robots in recent years. With Isaac Software Development Kit (SDK) and Isaac Sim, they develop a new solution for simulation addressing their Jetson platform. Since 2018, NVIDIA has further developed NVIDIA Isaac Sim (NIS) to support robots based on the Robot Operating System (ROS) and to use NVIDIA Omniverse as its platform.

The current go-to solution for simulating ROS-based robots is Gazebo, an open-source solution being under development since 2001. [1] Gazebo – developed by the Open Source Robotics Foundation (OSRF) - has reached its last major release in 2020 and is currently transferred to a new software foundation called “ignition”. Gazebos main focus is set on realistic physics simulation as well as digitally replicating sensor systems. Gazebo's role as a go-to solution in science and software development is reflected both in the number of publications found using the search term "Gazebo ROS" in the IEEE database and in Google search engine trends. Alternatives, such as RoboDK, SimSpark or Copellia Sim do not reach the popularity of Gazebo. Nvidia Isaac Sim has yet to be focussed in a scientific publication, however Google search trends show a significant increase in interest since January, overcoming most of the other simulation frameworks.

With novel solutions for robotics simulation slowly reaching maturity, a comparison between established software and newcomers is necessary to aid future developers. In our paper we compare Gazebo and NIS in the context of mobile, ROS-based robotics to achieve a use-case comparability. In chapter 1, the two simulation frameworks are detailed and a general comparison of features is done. In chapter 2, the applicability and features of both systems to use cases in mobile robotics is examined. In chapter 3, the real-world accuracy of both simulation environments is tested with different tasks representative for mobile robotics systems. A discussion of strengths and weaknesses is presented in chapter 4 before chapter 5 finishes with an outlook upon further improvements.

Disclaimer: All statements refer to Gazebo 9.16 and NIS version 2021.2, which are the latest version at the time of writing, and might not apply in later versions.

ARGESIM Report 20 (ISBN 978-3-901608-97-1), p 171-178, DOI: 10.11128/arep.20.a2033

1 Comparison of features

Isaac Sim is a simulation suite with focus on photorealistic presentation of simulated environments in conjunction with NVIDIAs hardware systems. Comparable with other visually focused frameworks like Unity [2], its strength lays in graphic representation rather than physics and behavior simulation. [3] Gazebo is a three-dimensional multi body simulation with a strong focus on the physical properties of materials and the modeling of physical effects.

1.1 Hardware requirements

The hardware requirements for Gazebo are low for today's standards. The OSRF specifies an Intel i5 (no specified generation) or equivalent CPU, a dedicated GPU and at least 500 MB free disk space. [4]

NIS has significantly higher requirements: As a minimum, an Intel i7 7th gen. or equivalent CPU, 32 GB of RAM, SSD Storage and a GeForce RTX 2070 GPU are required. [5] Without an RTX™ GPU by NVIDIA, NIS respectively the RTX™ renderer will not run. NIS therefore poses a much higher entrance barrier on the hardware side than Gazebo.

1.2 Degree of physics simulation

Gazebo offers four different physics engines: Open Dynamics Engine (ODE), Bullet, Dynamic Animation and Robotics Toolkit (DART) and Simbody. [6] NIS uses Nvidias PhysX engine, which originally was developed for computer games.

A 2015 paper found PhysX to be fast, but sometimes lacking in real world accuracy compared to Bullet and ODE. [7] Since then, NVIDIA has continuously worked on their physics engine, including an announcement to incorporate the needs of robotics simulation in 2018. [8] To date there is no specific research in performance and truth to reality of the different engines.

One major task of simulation is description of multibody systems. In the following we will compare the joint types available in Gazebo and the joint types available in NIS. Both programms provide essential standard types of joints by their file formats. Those are rotational joints in one or two degrees of freedom (dof), geared rotational joints, linear and fixed joints. Both simulators allow adding limits to those, although to a different extent. While Gazebo requires adding a different joint type to implement limits, NIS enables the users to add limits in the properties of the joint, which is more intuitive to the user.

In addition to those, each software has more special types. Gazebo additionally offers the "revolute2" joint – two serially coupled rotational joints – and a "screw" joint, which couples linear and rotational movement in one axis. NIS additionally offers three joints: Firstly, a 6-dof joint, which allows movement in all three linear and rotational axes, which can be limited. Secondly, a "distance" joint, which is only limited by the distance between the two connected bodies. Thirdly, a "rack-and-pinion" joint, which translates between rotational and linear movement, including a gear-ratio. Table 1 provides an overview over the mentioned joint types in each software. [10,9]

Joint type	Gazebo	NIS
Rotational (1-dof)	Yes	Yes
Rotational (2-dof)	Yes	Yes
Rotational (geared)	Yes	Yes
Rotational (serial)	Yes	No
Linear (1-dof)	Yes	Yes
Fixed	Yes	Yes
Screw	Yes	No
6-dof	No	Yes
Distance	No	Yes
Rack-and-Pinion	No	Yes

Table 1: Comparison of available joint-types in Gazebo and NIS

In NIS, non-geared joints can also be transferred into "articulations", which NVIDIA claims to behave more accurately than traditional simulated joints. [10]

Gazebos physics library supports aerodynamics and hydrodynamics which allows third party plugins to extensively simulate unmanned aerial vehicles (UAVs) and unmanned underwater vehicles (UUVs). [12,11] NIS does not support this and is therefore currently no alternative for those domains of robotics.

1.3 Graphics simulation

A field where NIS outdoes Gazebo literally on first look is the graphics simulation. Graphics is where Gazebo/OGRE start showing their age, while NIS shows its roots at NVIDIA and their profound experience in computer graphics.

Gazebo utilizes "OGRE" – the "Obejct-Oriented Graphics Rendering Engine". [6,13] NIS uses the

NVIDIA Omniverse RTX™ Renderer which allows for high-fidelity rendering using raytracing. NVIDIAs renderer uses their “Material Definition Language” (MDL) which is layer based and supposed to interact realistically with virtual light sources. [14] NIS comes with varied materials provided. Figure 1 illustrates the differences in graphics simulation between both programs.

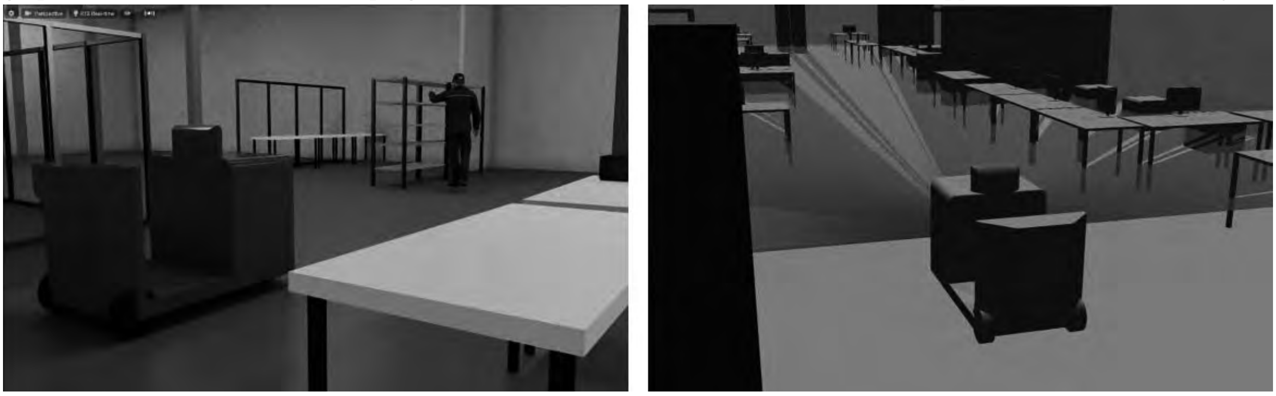

Figure 1: Screenshot from NIS (left) and Gazebo (right) showing a representative application scenario

Albeit in different ways, both Gazebo and NIS allow their users to add and modify virtual cameras. The parameters are comparable and allow the editing of intrinsic camera parameters like focal length or the projection type. NIS additionally allows editing the aperture and f-stop to simulate the depth of field of real cameras. [16,15]

1.4 Connection to other Frameworks

While Gazebo and NIS can be used as a standalone simulator, for mobile robotics they are often used in conjunction with ROS. Both simulators can initialize a ROS-node which acts as the translator between ROS and the simulator.

In Gazebo, this can be achieved by installing the “gazebo_ros_pkgs” metapackage in ROS, which interfaces ROS with Gazebo. Other open-source robotics frameworks cannot be integrated natively with Gazebo. [4]

NIS is comparable in this regard, but additionally can be integrated with the NVIDIA Isaac Toolkit, aimed towards the development of AI-powered robots on NVIDIAs platforms. [17] For NIS there is no need to install a ROS-package, as the ROS-bridge is included as an extension.

2 Use-case specific comparison

The usefulness of different simulation environments is highly dependent on the corresponding use case. For comparability, we define a use case relevant for research in mobile robotics to test both simulation toolkits. Subsequently, we are comparing the capabilities of Gazebo and NIS in regards to use-case specific features, mainly sensor simulation adressing mobile robots. We also address the potential change and interoperability from Gazebo to NIS, based on the broad application of Gazebo today.

2.1 Mobile robotics as use-case

Mobile robot systems are often used in intralogistics applications for warehousing and commissioning. Perception is especially important for autonomous mobile robots, since they need to be able to react safely in dynamic environments. To guarantee a safe and predictive behaviour, simulation of systems and intralogistics environments is necessary, especially in regards to capabilities of used sensor systems, like radio detection and ranging (RADAR), cameras, light detection and ranging (LiDAR) and other sensor principles. Additionally, testing mobile robots in real environments is challenging, because they might pose a threat to bystanders and hinder operations in work environments. Thus, we chose a mobile robot to compare Gazebo with NIS. As real-world-reference, we use an autonomous tow-truck, which is described in [18]. The robot and its testing environment have been recreated in both simulators.

2.2 Interfaces between ROS, Gazebo and Isaac Sim

The first step in simulating mobile robotics is the integration of robot models and worlds into the corresponding environment. Gazebo uses .sdf-files to describe robots and worlds, NIS uses the .usd-format. Right now, both formats are not directly interchangeable and to different degrees supported by ROS. To facilitate the use of models in other formats, NIS offers a variety of “connectors”, which allow easy switching back and forth with software by other companies like Autodesk.

For .urdf-files, which are used in ROS to describe robots and are also compatible with Gazebos .sdf-Format, NIS provides a simple import function. However, the importer does not deliver reliable and deterministic results. Especially the type of joints and their orientation are not converted reliably for different models. For more complex .sdf-files, currently no way to import them into NIS exists. This results in the need to completely rebuild the simulated environments and also the robots in most cases. For some common commercially available robots, NIS provides pre-built models. If worlds or objects are available in a format usable by the free 3D-modeling software blender, they can be exported as .usd-files from there. To sum up, migrating extensive worlds and models between Gazebo and NIS is a laborious task. It is advantageous to decide early in a project, which software to use.

Otherwise, using ROS with NIS is pretty straightforward. To use the ROS-bridge in NIS, users have to add so called “ROS-Components” to their model, which is done intuitively via the GUI. Available are: ROS-clock, Camera, Joint State, Lidar, Pose Tree, Teleport, Surface Gripper and Differential Base. The components can be configured in various ways and their corresponding topics can be modified to work with the topics of an existing ROS project. Therefore, if all necessary components exist in NIS, the integration of ROS on the software side is just a minor inconvenience.

However, not all necessary components for the mentioned use-case are available in NIS. The considered tow-truck doesn’t use a differential drive model which is common for robots, but a model which is comparable to Ackermann steering for vehicles. For this reason, some modifications are necessary, which translate the standard message type of ROS to a rotation angle of the steering and velocity of the drive wheel. Those topics can then be handled by joint state components in NIS. Additionally, not all necessary sensors are available, which is described in the following subsection.

2.3 Sensor simulation

Sensor systems are a core element of the simulation of autonomous robots. For this reason, special attention is given to the capabilities of both simulators in this area of operation. The real use-case robot utilizes a LiDAR-scanner, two RGB-D cameras with integrated Inertial Measurement Units (IMUs), a RADAR sensor, ultrasonic sensors and different local and global multilateration systems like GPS.

Natively, Gazebo offers mainly the following virtual sensors: Different cameras (including depth-cameras), beam and wave-based range sensors, which natively allow the simulation of 1D to 3D laser sensors, inertial measurement units (IMUs) and a bumper or contact sensor. Further sensors can easily be added via third-person plugins, exemplary a GPS receiver. [4]

NIS offers the same sensors that Gazebo offers natively, additionally the range sensor can be implemented as an ultrasonic sensor. However, the ROS-Bridge of NIS only allows the use of cameras and LiDARS – support for other sensors currently has to be developed by the users. Ultrasonic and contact sensors as well as IMUs can therefore not be used with ROS out of the box. Additionally, the documentation for implementation and modification those sensors is extremely scarce. NIS does allow user-made extensions and provides a tutorial on how to integrate the contact sensor with ROS. Unfortunately, at the time of writing only very few of those extensions are publicly accessible. Thus, not all sensors necessary for mobile robotics systems are available. The following Table 2 provides an overview on the availability of different sensor types in both simulators out of the box.

Sensor type	Gazebo	NIS
RGB camera	Yes, natively	Yes
Depth camera	Yes, natively	Yes
LiDAR	Yes, natively	Yes
IMU	Yes, natively	Limited, see text
Contact	Yes, natively	Limited, see text
Ultrasonic	Yes, plugin	Limited, see text
GPS	Yes, plugin	No

Table 2: Availability of relevant sensors in Gazebo and NVIDIA Isaac Sim

When using simulations, one always has to keep in mind the deviations between the simulation and reality, known as “Sim2Real Gap”. NVIDIA claims to close this gap with NIS. [19] Upon inspection, the statement not yet holds.

Comparable to Gazebo [1] NIS calculates the measurements of range sensors using the collision geometry of objects. That approach neglects the considerable effects of material properties like transparency and reflections on optical time-of-flight sensors. NVIDIA is planning on supporting RTX™ and with it raytracing for sensors in NIS in the future, starting with the version released in spring 2022. [20]

An even more relevant influence on real sensor measurements is noise, which is existent in all sensor systems. Some sensors and environmental influences have been studied extensively, for example LiDAR systems, especially in the context of safety systems. [22,21] To simulate real working conditions, it is necessary to also model noise in simulated sensors. Gazebo supports adding Gaussian noise to sensor readings, which is usually sufficient to approximate sensor noise. [4,22,21] NIS does not support adding noise to any sensors, which is a major drawback for realistic simulation tasks. Modifications by adding sensor noise in ROS are possible but cannot substitute an integration in the simulation environment. Support for sensor noise is also announced for the next release of NIS. [23]

2.4 Simulation of moving objects and persons

In recent years, increasingly many robots are used in mixed areas with human workers. To simulate operation in those spaces, especially for mobile robotics, a realistic simulation of moving persons is necessary. Gazebo handles this task using “actors”. Actors are animated models, on which a skeleton and a trajectory animation can be applied. Their trajectory can also be influenced by the environment. [4] NIS does not have those capabilities. While NIS does provide animated and realistic looking models, eg. of a worker, the animation in NIS only influences the visual representation of a model. If a collision model is added, it stays in a stationary “T-Pose”. Also, there is no way to let those movements be influenced by the environment. At the time of writing, there are no noticeable intentions by NVIDIA to change this.

3 Comparison of simulation and a real system / Sim to Real

Based on the defined use case, a comparison of both simulations and a real system is conducted to evaluate the degree of concordance. In mobile robotics – like the described autonomous tow-truck - occupancy grid maps (OGMs) are commonly used for mapping and localization purposes and heavily dependend on the sensors and their perception of the working environment. With many algorithms allowing for extensive parametrization, simulation plays a major role in optimization and reducing field testing.

The evaluation is conducted by comparing different mapping algorithms in three representations of the same environment. We apply the particle-filter-based “hector_mapping”, which only uses LiDAR-data, and the two graph-based approaches “slam_toolbox" and “RTAB-map”. [24,26,25] The former utilizes LiDAR- and odometry-data, the latter has been configured to use RGB-D- and odometry-data.

Each simulation environment is parameterized and hand-built to replicate the real environment as realisticly as possible. The driven trajectory of the mobile robot is comparable for all mapping processes and consists of a closed loop trough the environment. Mapping in the real environment has been conducted with an Omron OS32C-SP1-4M laser scanner and an Intel D435 RGB-D camera. Both sensors are recreated in the simulators as precisely as possible. Noise is added to the laser scanner readings natively in Gazebo and using a self-written script with comparable functionality in NIS. The odometry-data of the real system is generated using the sensor data fusion approach described in [18]. For the simulated systems, no modification of odometry data is possible. This is suboptimal, since the quality of odometry data significantly influences the SLAM-problem. [27]

The following maps in Figure 2 are generated using the LiDAR-based hector_mapping in the three environments. It can be seen as a reference for Extended Kalman Filter – SLAM algorithms (EKF-SLAM).

We can see, that hector_mapping delivers good results, both in reality and simulation. The influence of noise for detectable surfaces can be can be sufficiently approximated using gaussian noise. In addition, however, it also becomes apparent that critical interfering factors for laser systems, such as reflections or materials with different optical properties, cannot be simulated correctly.

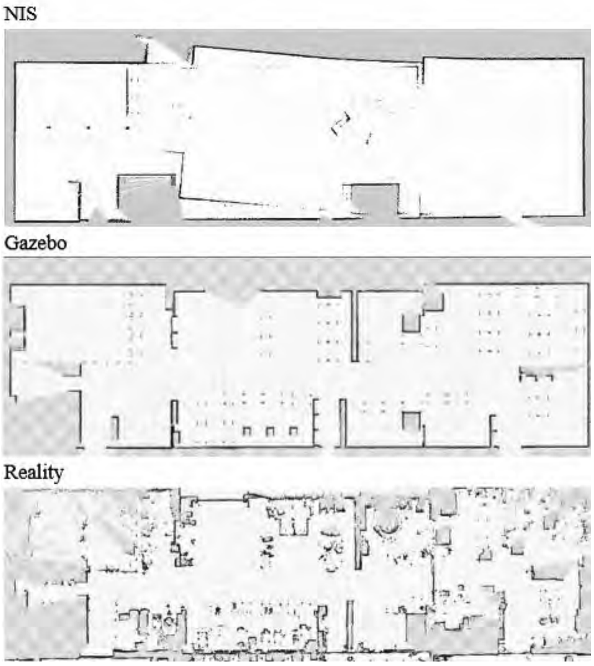

Figure 2: Occupancy Maps generated using hector_mapping in NIS, Gazebo and using a real system; The two simulated environments were built to mirror the real environments in dimension and interior.

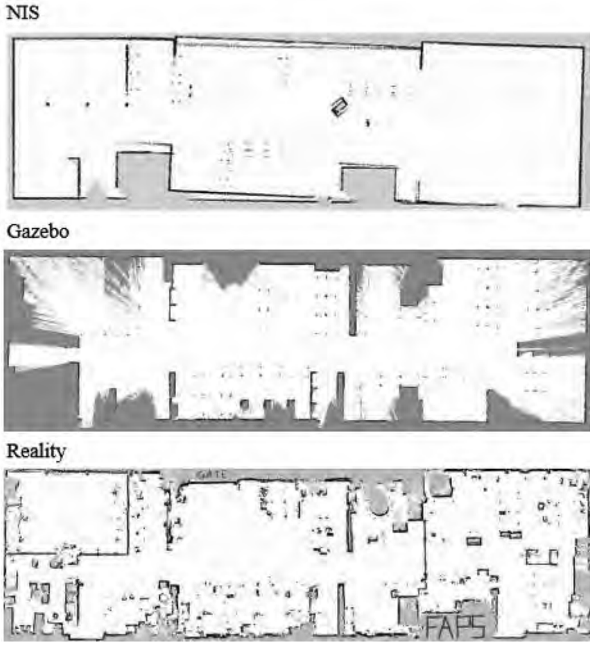

Figure 3: Occupancy Maps generated using SLAM Toolbox with added noise in NIS (σ = 0.5%)

The maps shown in Figure 3 are generated using the “slam_toolbox” ROS-package to show the capabilities of the different environments for development and testing of graph-based SLAM. Again, both simulators deliver usable results. However, the Sim2Real gap is wider here, likely because of the use of perfect odometry data in simulation.

The previously described approaches did not deliver significantly distinguishable results between both simulators. This meets our expections, since like described in section 2.3, NIS does not yet provide improvements in simulation of LiDAR-scanners. Next, we compare the results from the “RTAB-Map” package, which uses data from a RGB-D camera and odometry data.

From Figure 4 it can be seen that Gazebo is not suitable for photorealistic rendering of depth images. Due to the low level of detail, feature-dependent algorithms such as “RTAB-Map” can only be used and tested to a limited extent. NIS enables the robust use of the same algorithms, even though a difference in results between the real system with changing lighting situations and perturbations and the simulation can still be seen.

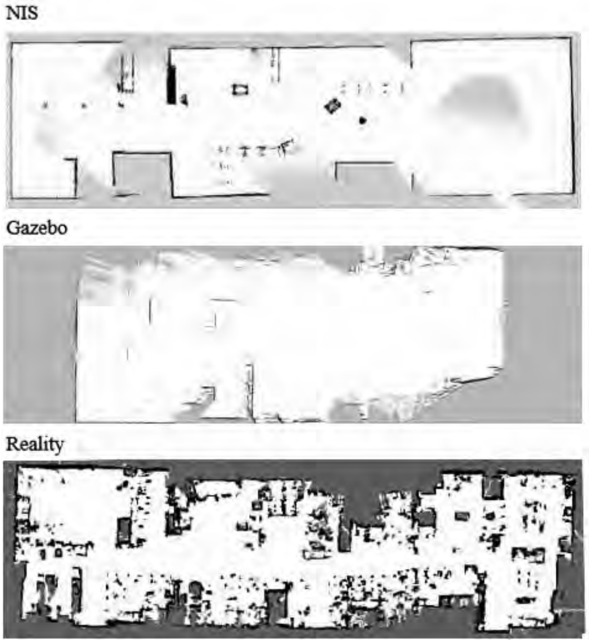

Figure 4: Occupancy Map generated using RTAB-Map with RGB-D and Odometry data for the different environments

4 Discussion of performance

In this chapter we summarize our findings into recommendations for a ROS-simulator for different demands. The results are presented as tables. Table 3 compares NIS and Gazebo in different Criteria which are based on performance described in the previous sections. Both simulators are rated either with ++ (very positive), + (positive), 0 (neutral), - (negative), -- (very negative) as shown in [28–30]).

Criteria	Gazebo	NVIDIA Isaac Sim
Hardware requirements	+	--
Software requirements	+	**0**
Physics simulation	+	+
Graphics simulation	-	++
Sensor simulation	+	-
Universality	++	**0**

Table 3: Rating of NVIDIA Isaac Sim and Gazebo in different Categories

Table 4 lists different possible demands in a project and suggests the better suited simulator. Both tables are meant to provide guidance when choosing a software to simulate mobile ROS-based robots during development. The criteria and ratings are explained in detail in the previous sections.

Demand	Better suitable simulator
Physics simulation	Gazebo
Evaluation of vision-based algorithms	NVIDIA Isaac Sim
Evaluation of optical-/laser-based algorithms	Gazebo (until NIS utilizes raytracing for laser simulation)
Project duration	Short- & medium-term: Gazebo Long-term: NVIDIA Isaac Sim
Simulation of UAVs or UUVs	Gazebo
Availability of moving objects and persons	Gazebo

Table 4: Simulator recommendations for different demands

5 Outlook on further developments

As described above, development cycles are becoming shorter and shorter and new players in the field of robotics simulation are entering the market. Based on our expertise, we venture a look into the crystal ball and attempt a forecast for developments in the near future.

5.1 Expected evolution of Isaac Sim

As shown before, NIS is built on a more modern platform than Gazebo. However, NIS is not yet as mature as Gazebo and lacks features in some areas crucial for mobile robotics. In this subsection we summarize what to expect from NIS for the future based on press releases and forum discussions. In general, NVIDIA’s staff is very helpful in the forum and happy to add requested features or reported bugs to their list. It is therefore expected that frequently demanded features will make their way into NIS sooner or later. Unfortunately, there is no official roadmap available, which complicates planning in projects. The OSRF and NVIDIA have announced interoperability between NIS and the new Ignition Gazebo. Prospectively, both simulators are supposed to feature converters for .usd respective .sdf files and also allow direct interaction. Ignitions roadmap aims to release these features in September 2022. [31]

5.2 Conclusion

From our research, we conclude that NIS is not yet ready to supersede Gazebo as the go-to simulation tool for mobile robots using ROS. At the time of writing, NIS imposes many hurdles to overcome when using the software. Many restraints of NIS can be overcome by developing own workarounds. However, those are seldom optimal. Gazebo is also still on par with NIS regarding physics simulation tasks. While NVIDIA has some lighthouse projects together with major corporations like BMW or recently Amazon, it’s scientific user base seems to be small. This is a chicken-and-egg-problem which requires some pioneering work to solve itself. Nevertheless, in the time of our usage, we noticed significant advances in NIS. We therefore see a high potential for NIS. Coming features could lift NIS on a new level, eg. factoring in reflection properties of surfaces for ToF-sensors. If one has access to a system capable of running NIS, trying it out might be worthwhile depending on the type of project – see Table 4. If one can’t access such a system, Gazebo still is a very powerful tool and might have a comparable successor for ROS 2 with Ignition Gazebo. Further research should be conducted to evaluate the current state of the PhysX-engine and comparing performance and truth to reality of the mentioned engines. New releases of NIS might generate additional needs for research, especially on the simulation of reflective properties for ToF-sensors and the interoperation between NIS and Ignition Gazebo.

References

[1] Koenig, N., Howard, A., 2004. 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). 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), Sendai, Japan. 28 Sept.-2 Oct., 2004. IEEE, pp. 2149–2154.

[2] Hussein, A., Garcia, F., Olaverri-Monreal, C., 2018. ROS and Unity Based Framework for Intelligent Vehicles Control and Simulation, in: 2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES). 2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES), Madrid. 12.09.2018 - 14.09.2018. IEEE, pp. 1–6.

[3] Santos Pessoa de Melo, M., Da Gomes Silva Neto, J., Da Jorge Lima Silva, P., Natario Teixeira, J.M.X., Teichrieb, V., 2019. Analysis and Comparison of Robotics 3D Simulators, in: 2019 21st Symposium on Virtual and Augmented Reality (SVR). 2019 21st Symposium on Virtual and Augmented Reality (SVR), Rio de Janeiro, Brazil. 28.10.2019 - 31.10.2019. IEEE, pp. 242–251.

[4] Osrf, 2022. Gazebo : Tutorials. http://gazebosim.org/tutorials. Accessed 22 March 2022.

[5] Isaac Sim Requirements — Omniverse Robotics documentation, 2022. https://docs.omniverse.nvidia.com/app_isaacsim/app_isaacsim/requirements.html. Accessed 15 February 2022.

[6] Osrf, 2022. Gazebo. http://gazebosim.org/. Accessed 18 February 2022.

[7] Erez, T., Tassa, Y., Todorov, E. Simulation tools for model-based robotics: Comparison of Bullet, Havok, MuJoCo, ODE and PhysX, in: 2015 IEEE International Conference, pp. 4397–4404.

[8] Lebaredian, R., 2018. NVIDIA Extends PhysX for High-Fidelity Simulations, Goes Open Source. NVIDIA. https://blogs.nvidia.com/blog/2018/12/03/physx-high-fidelity-open-source/. Accessed 15 February 2022.

[9] Osrf, 2022. SDFormat Specification. http://sdformat.org/spec?ver=1.9&elem=joint. Accessed 28 February 2022.

[10] Physics Core — Omniverse Create documentation, 2022. https://docs.omniverse.nvidia.com/app_create/prod_extensions/ext_physics.html. Accessed 28 February 2022.

[11] Meyer, J., Sendobry, A., Kohlbrecher, S., Klingauf, U., Stryk, O. von, 2012. Comprehensive Simulation of Quadrotor UAVs Using ROS and Gazebo, in: Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M.Y., Weikum, G., Noda, I., Ando, N., Brugali, D., Kuffner, J.J. (Eds.), Simulation, Modeling, and Programming for Autonomous Robots, vol. 7628. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 400–411.

[12] Manhaes, M.M.M., Scherer, S.A., Voss, M., Douat, L.R., Rauschenbach, T., 2016. UUV Simulator: A Gazebo-based package for underwater intervention and multi-robot simulation, in: OCEANS 2016 MTS/IEEE Monterey. OCEANS 2016 MTS/IEEE Monterey, Monterey, CA, USA. 19.09.2016 - 23.09.2016. IEEE, pp. 1–8.

[13] OGRE - Open Source 3D Graphics Engine | Home of a marvelous rendering engine, 2022. https://www.ogre3d.org/. Accessed 18 February 2022.

[14] NVIDIA, 2022. Material Definition Language from NVIDIA. https://www.nvidia.com/en-us/design-visualization/technologies/material-definition-language/. Accessed 18 February 2022.

[15] Osrf, 2022. SDFormat Specification. http://sdformat.org/spec?ver=1.9&elem=sensor. Accessed 21 February 2022.

[16] Cameras — Omniverse Create documentation, 2022. https://docs.omniverse.nvidia.com/app_create/prod_materials-and-rendering/cameras.html#. Accessed 21 February 2022.

[17] NVIDIA Developer, 2019. NVIDIA Isaac SDK. https://developer.nvidia.com/isaac-sdk. Accessed 7 March 2022.

[18] Herbert, M., Zwingel, M., Czapka, C., Franke, J., 2022. A Multi-source Localization System for Driverless Material Transport in Mixed Indoor and Outdoor Areas, in: Production at the Leading Edge of Technology. Springer International Publishing, Cham, pp. 421–429.

[19] NVIDIA Technical Blog, 2021. Training in NVIDIA Isaac Sim Closes the Sim2Real Gap | NVIDIA Technical Blog. https://developer.nvidia.com/blog/training-in-nvidia-isaac-sim-closes-the-sim2real-gap/. Accessed 21 February 2022.

[20] NVIDIA Developer Forums, 2021. Influence of materials on LIDAR and (depth-)cameras - Isaac / Omniverse Isaac Sim - NVIDIA Developer Forums. https://forums.developer.nvidia.com/t/influence-of-materials-on-lidar-and-depth-cameras/197372. Accessed 21 February 2022.

[21] Ye, C., Borenstein, J., 2002. Characterization of a 2D laser scanner for mobile robot obstacle negotiation, in: Proceedings of the 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292). 2002 IEEE International Conference on Robotics and Automation, Washington, DC, USA. 11-15 May 2002. IEEE, pp. 2512–2518.

[22] Reina, A., Gonzales, J., 1997. Characterization of a radial laser scanner for mobile robot navigation, in: Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS '97. 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS '97, Grenoble, France. 11.09.1997 - 11.09.1997. IEEE, pp. 579–585.

[23] NVIDIA Developer Forums, 2021. Adding noise to sensor readings - Isaac / Omniverse Isaac Sim - NVIDIA Developer Forums. https://forums.developer.nvidia.com/t/adding-noise-to-sensor-readings/186141. Accessed 21 February 2022.

[24] Kohlbrecher, S., Stryk, O. von, Meyer, J., Klingauf, U., 112011. A flexible and scalable SLAM system with full 3D motion estimation, in: 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics. 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Kyoto, Japan. 01.11.2011 - 05.11.2011. IEEE, pp. 155–160.

[25] Macenski, S., Jambrecic, I., 2021. SLAM Toolbox: SLAM for the dynamic world. Journal of Open Source Software 6 (61), 2783.

[26] Labbé, M., Michaud, F., 2019. RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. J Field Robotics 36 (2), 416–446.

[27] Thrun, S., Burgard, W., Fox, D., 2005. Probabilistic robotics. MIT Press, Cambridge, Mass., London, 647 pp.

[28] Aaron Staranowicz, Gian Luca Mariottini, 2011. A Survey and Comparison of Commercial and Open-Source Robotic Simulator Software. ACM, New York, NY, 401 pp.

[29] Michal, D.S., Etzkorn, L. A Comparison of Player / Stage / Gazebo and Microsoft Robotics Developer Studio. ACM Conferences. Association for Computing Machinery, New York, NY, 399 pp. http://dl.acm.org/citation.cfm?id=2016039.

[30] Pitonakova, L., Giuliani, M., Pipe, A., Winfield, A., 2018. Feature and Performance Comparison of the V-REP, Gazebo and ARGoS Robot Simulators, in: Towards Autonomous Robotic Systems. Lecture Notes in Artificial Intelligence. Towards Autonomous Robotic Systems (TAROS) Conference, Bristol. July 25-27, 2018. Springer International Publishing, Cham, pp. 357–368.

[31] Ignition - Docs: Roadmap, 2021. https://ignitionrobotics.org/docs/all/roadmap. Accessed 21 February 2022.