














Simulation of Autonomous Robotic System for Intelligence and Reconnaissance Operations

Milos Cihlar¹ , Petr Raichl¹ , Petr Gabrlík¹ , Jiri Janousek¹ ,
Petr Marcon¹  , Ludek Zalud¹ , Tomas Lazna¹ , Karel Michenka² ,
Jan Nohel² , and Alexandr Stefek² 

¹ Faculty of Electrical Engineering and Communication, Brno University of Technology,
Technicka 12, 616 00 Brno, Czech Republic
marcon@vut.cz

² University of Defence, Brno, Czech Republic

Abstract. This paper examines the possibilities of realistic simulations of robotic systems employed for intelligence and reconnaissance operations in the outdoor environment. Including simulation into the development process accelerates and facilitates testing, verification, and evaluation of algorithms, and prevents potential damage of expensive hardware. To achieve fast and flexible development, we utilize a widely used Robotic Operating System (ROS) framework, which, together with Gazebo simulator, enables to deploy robots and test algorithms in both real-world and simulation. Gazebo supports a wide range of customization options, including the creation of own worlds, robots, and sensors. Moreover, the solution allows to deploy a multiple heterogenous robots within one simulation instance to test cooperative missions. To demonstrate the potential of this simulation concept for intelligence operations, we introduce a scenario involving several flying and terrestrial robots during a Chemical, Biological, Radiological, Nuclear, and Explosive (CBRNE) mission comprising a radiation source search. For this purpose, we deployed realistic robotic platforms into the aerial photogrammetry-based 3D world, and, above all, we improved the standard radiation plugin to collect credible data. The results indicate that the concept fulfills requirements for intelligence and reconnaissance robotic operation simulation.

Keywords: Simulation · mobile robots · Unmanned Aircraft System · Unmanned Ground Vehicle · swarm · ROS · Gazebo

1 Introduction

In the last twenty years, robotics, especially mobile robotics, has been on the rise. Modern robotic systems are equipped with a variety of sensors, actuators, computers, and other equipment, that aim to accomplish a desired mission, a task that can be performed by a human, or an activity that a human does not want or cannot perform. A robot competent in performing these tasks must be capable of basic movement, orientation, perception, and decision-making. Therefore, robots are complex hardware devices with complicated software. Sensors are expensive, hardware is difficult to work with, and the risk of

accidents is high. These reasons make the development extremely expensive. There is pressure and a tendency to move the development, testing, and debugging of algorithms to simulators, which greatly speed up, simplify and cheapen robot development and research [1–4].

The 2020 survey showed that approximately 72% of robotics organizations use simulators for their activities [5]. Types of commonly used robotic simulators are described and compared in the literature [6]. With correctly modeled robots and simulation environments, the software used in the simulation can be directly used in the real world [7]. Therefore, this article deals with the possibilities of simulating mobile robots in real-world inspired environments for fast and effective detection of CBRN threats. Obtaining information through CBRN reconnaissance will contribute to the protection of military units and ensure awareness within the fulfillment of Intelligence analyses of a battlefield to identify and specify CBRN threats. Moreover, the cooperation of the swarm of flying robots (Unmanned Aircraft Systems, UASs) with terrestrial robots (Unmanned Ground Vehicles, UGVs) takes advantage of fast aerial mapping and ground reconnaissance with high sensitivity.

2 Materials and Methods

2.1 Gazebo

Gazebo is an open-source 3D robotics simulator developed by Open Source Robotics Foundation used for indoor and outdoor missions [2]. Nowadays, simulation plays an important role in robotics research, and it is used for design, development, verification, validation, testing, and proving entire robotics systems. Today's robots represent complex hardware devices including numerous sensors, and they may be deployed to fulfill different tasks in various environments. Therefore, well-designed robotic simulators accelerate the development time. The Gazebo has become the most popular robotic simulator because it provides many options to simulate indoor and outdoor 3D environments, numerous robots, objects, and sensors that are protected before the damage, unlike real-world experiments. Moreover, Gazebo enables easy integration into the Robotic Operating System (ROS) [8], a widely used robotic framework. The application itself consists of two main parts: first, *gzserver*, which executes the physics update-loop and sensor data generation; second, *gzclient*, which runs a graphical user interface. Both components are independent of each other [1].

2.1.1 World

The Gazebo world represents a simulated environment where the robots operate and perform desired tasks, and typically it embodies a simplified representation of a portion of a real-world environment to achieve authentic conditions. The world, 3D model, may be assembled in two major ways: manually utilizing a software tool for 3D modeling, or by using a technique for environment scanning such as laser scanning or photogrammetry. The latter approach is advantageous especially outdoors since it enables to obtain a 3D representation of a large-scale area with minimal effort and with a reasonable level of detail. Within this paper, we use UAS photogrammetry-based model of a real-world in

COLLADA format, which is compatible with the Gazebo. The world has an approximate area of 2 ha, and since it involves flat areas, numerous buildings, vegetation, and other non-traversable zones, it represents a suitable world for simulating the operation of a heterogeneous robotic system (Fig. 1).



Fig. 1. The Gazebo world generated from the UAS photogrammetry-based 3D model.

2.1.2 Unmanned Ground Vehicle

The simulation scenario presented within this paper involves the deployment of simulated a UGV (Fig. 2), namely a four-wheel skid steering mobile robot based on a real UGV Orpheus. This custom-built platform was designed for reconnaissance purposes, and it stands out for its mechanical robustness and maneuverability in an outdoor environment. The robot integrates ROS-based software to enable motion control and reading sensor data in both the real-world and simulation. Within the proposed scenario, the UGV has equipped with a $2'' \times 2''$ NaI (TI) radiation detector (see *Gazebo Radiation Plugin* section).

Gazebo loads models and worlds from the self-descriptive System Description Format (SDF) file. This approach is unsuitable for editing by hand and dynamic robot configuration because SDF is not scriptable. Therefore, the SDF file contains similar duplicate link descriptions. The solution for the swollen code is a parser for the Unified Robot Description Format (URDF) that provides macro for XML language (XACRO). URDF is XML robot model representation. The format consists of an exclusive robot tag, its links, and nested visual and collision tags. The independent definition of visual and collision properties enables collision simplification with the simultaneous preservation of the robot's looks. This simplification saves computation time and resources. Joints combine links together.

Inertial tags are required in each link for proper parsing to SDF format. This tag defines the mass of links and their moment of inertia. Hence, the main physical properties are determined. The gazebo tag is optional and defines other simulation properties such as damping, friction, contacts, and repulsion force. It also includes plugins for specifying a link's behavior [1].



Fig. 2. Orpheus model in Gazebo simulator.

XACRO provides a math constant, simple math, and parametric macros for a significant reduction of a line of code. XACRO makes code more clear and adjustable. Making a robot with macros allows adjusting (by loading a YAML file) the robot's settings, especially its sensors, without managing multiple similar files with robot descriptions.

2.1.3 Unmanned Aircraft System

The software architecture for the UASs is based on several components, namely, on the PX4 flight stack, the MRS UAV control and estimating system [3], and custom-made high-level algorithms for mission control (Fig. 3). Except the PX4, which is designed for flight controllers (such as Pixhawk, for example), the components are ROS-based, and run on the on-board computer while using real robots. However, all the software may run on a PC together with the Gazebo simulator to evaluate algorithms before real experiments.

In general, the models of UASs for Gazebo are created similarly to the UGV model described above, and the MRS UAV system already includes several simulated platforms inspired by real vehicles. For our simulation, we employed Tarot t650 UAS (Fig. 4), and integrated the radiation plugin (see *Gazebo Radiation Plugin* section) into the system. The solution enables the deployment of multiple robots simultaneously and performs various tasks from waypoint following to complex algorithms such as autonomous landing, for example [9].

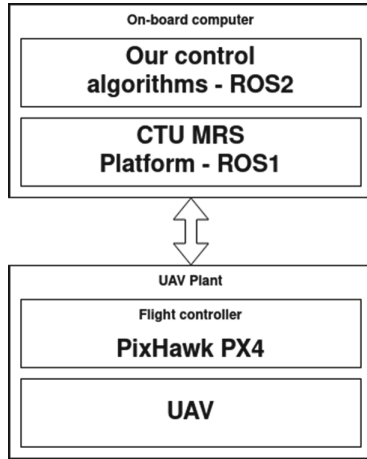


Fig. 3. System architecture of Unmanned Aircraft System.



Fig. 4. Simulation of UASs in Gazebo world.

2.1.4 Gazebo Radiation Plugin

Radioactivity simulation is done with the help of two plugins attached to the detector and radiation source model in the Gazebo simulator. The source is modeled as a point radioactivity source with constant activity. The physics of the radiation measurement consists of two principles: first, it depends on the number of particles that go through the simulated detector, which is strongly correlated with distance; second, the attenuation due to obstacles between sources and sensor is taken into account. The plugins are based on the *gazebo_radiation_plugin* ROS package, which also consider the situation when the radiation source is close to the detector, and intensity goes to infinity according to

the inverse square law. The implementation captures half of the released particles when the distance goes to zero, which reflects reality satisfyingly [4].

We reimplemented the plugin for ROS2, and improved it to respect the effects of the radiation background, and sensor's dead time, a phenomenon causing the detector cannot to count particles due to the previous particle pass. Therefore, the package was adjusted to behave similarly to a real NaI(Tl) detector. The difference between ideal sensors and sensors with dead time illustrates Fig. 5.

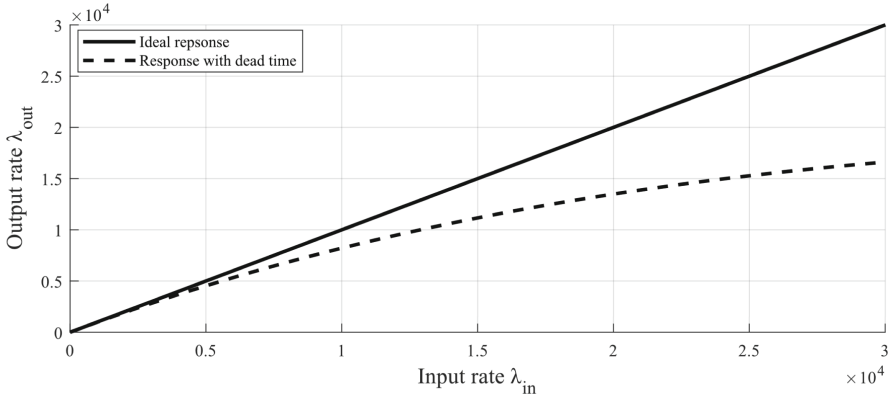


Fig. 5. The response comparison of an ideal sensor and the sensor with dead time.

The property of dead sensor time τ is modeled as a function that transforms input ideal rate λ_{in} to output real rate λ_{out} .

$$\lambda_{out} = \lambda_{in} e^{-\lambda_{in} \tau} \quad (1)$$

2.2 Simulation Scenario

To demonstrate Gazebo's capabilities for simulating multi-robot CBRN missions, we built a real-world scenario involving a squad conducting a CBRN reconnaissance using robotic assets in an urban area with potential radiation contamination.

The contamination is characterized by a single-point radiation source, namely, a Co-60 isotope exhibiting the activity of 170 MBq, positioned in the container in the vicinity of the middle building. However, in terms of the simulated robotic reconnaissance, these parameters were not available.

The robotic mission consists of two main phases. The former rests in the automatic aerial radiation data collection employing three identical UASs equipped with compact $1.5 \times 1.5''$ NaI(Tl) detectors. To obtain rough, equally distributed data in the minimum amount of time, the area of interest is divided into three portions and mapped by the individual UASs in parallel. The goal is to operate at a minimal, obstacle-free altitude, and fly in parallel lines with spacing equal to half of the above-ground level (AGL) altitude.

The latter phase involves terrestrial reconnaissance by using the UGV fitted with a $2 \times 2''$ NaI(Tl) detector. The UGV operation is intended to be manual, i.e. remotely controlled, or semi-automatic constituting an operation mode comprising waypoint-following. In general, the waypoints may be created utilizing the photogrammetry-based map manually or automatically. The latter approach is feasible, for example, by using Maneuver Control System CZ software developed at the University of Defense, which calculates the shortest and safest route of movement based on a combination of the effects of surface character, elevation, weather, enemy deployment and own troops [10, 11]. In any case, the goal is to visit the potential radiation hotspot found in the interpolated aerial radiation map, and collect more accurate radiation data suitable for a spectral analysis and source identification.

3 Results

The UAS operation lasted approximately 5:08 min in the simulation time and 5:50 min in the real time at PC with Intel(R) Core(TM) i5-10210U CPU 1.60 GHz, 8 GB DDR4 RAM. Across the 2.1 km long flight at 20 m AGL (Fig. 6), more than 10 thousand radiation measurements were performed with a minimum and maximum values of 67 and 245 CPS (Counts Per Second), respectively (Fig. 7). After the radiation data processing and interpolation, a single hotspot may be clearly recognized just above the location containing the source (Fig. 8). Based on the approximate hotspot position, waypoints for the ground reconnaissance were manually selected, and UGV followed the coordinates automatically. This part of the mission lasted 3.5 min in the real time, and about 200 radiation measurements were accomplished within 180 m long ride. Thanks to the higher sensitivity of the onboard detection system and smaller distance from the source, the data exhibit minimum and maximum value of 169 and 3688 CPS, respectively (Fig. 9).

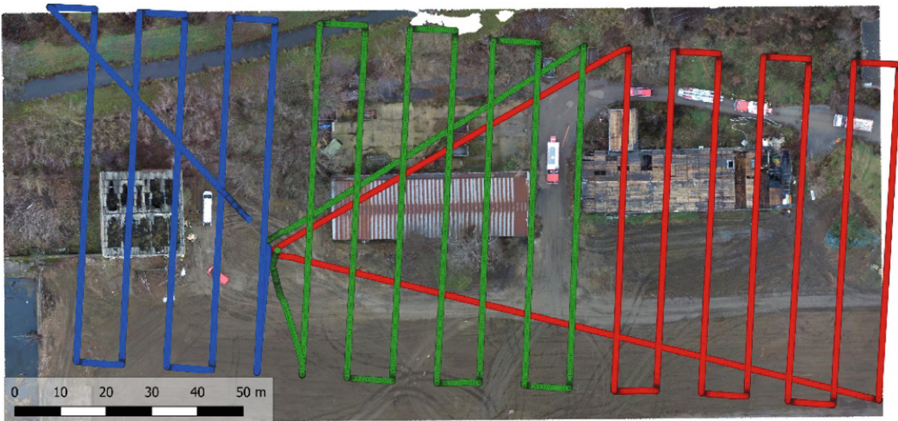


Fig. 6. The flight trajectories of the individual UASs.

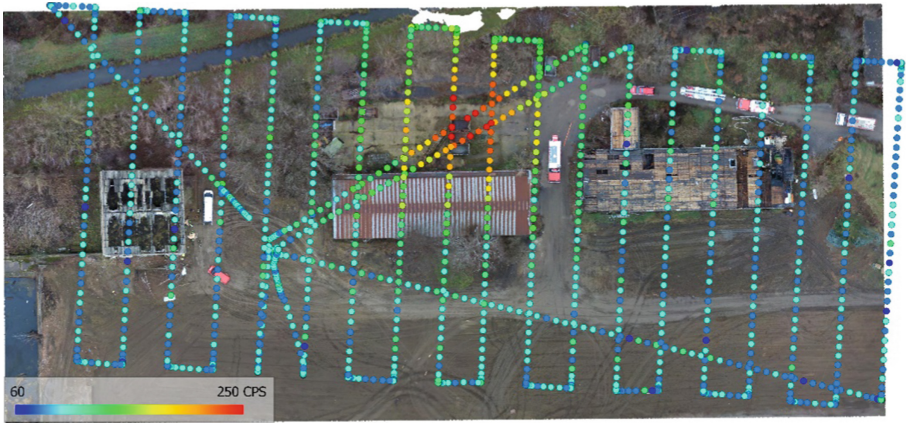


Fig. 7. Aerial radiation data collected by all the UASs.

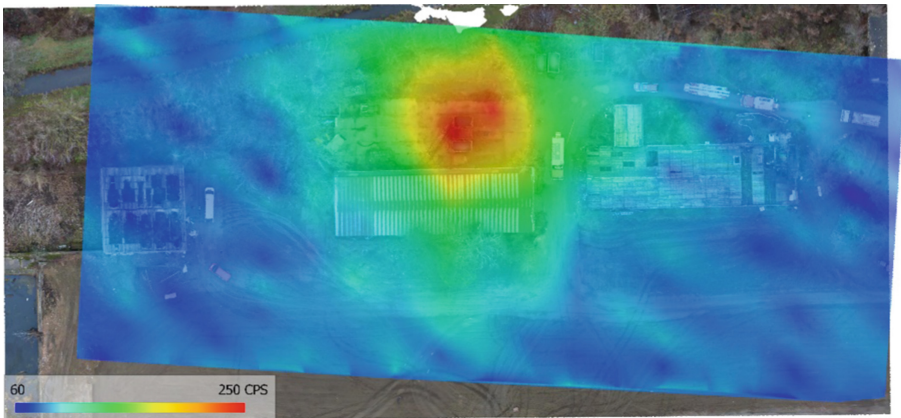


Fig. 8. Interpolated aerial radiation data collected by all the UASs.



Fig. 9. Terrestrial radiation data collected by the UGV.

4 Discussion and Conclusion

Within this paper, we presented basic possibilities of highly realistic simulations for autonomous robotic operations during reconnaissance missions. The employed simulation software, Gazebo, integrates a physics engine with sufficient accuracy for desired tasks and enables setting up the simulation realistically in terms of the environment, robotic platforms, and sensors.

We demonstrated that the simulation environment, or world, may be easily assembled from the aerial photogrammetry or laser scanning-based 3D models of real-world study sites allowing us to test various scenarios in natural conditions. The models of the robotic platforms were created concerning their actual physical parameters, including the drive type, which is essential to achieve reliable behavior while traversing the terrain or moving in the environment, in general. Gazebo enables the deployment of a large number of heterogeneous robotic platforms within one simulation instance to test cooperative missions involving robot interactions; the main limitation rests in the availability of computational resources. To demonstrate the capability of utilizing various sensors, we used customized gamma radiation sensors to detect radiation sources. Similarly, other sensors can be incorporated into the simulation as well. Currently, the following sensors are available: RGB camera, depth camera, laser rangefinder, LiDAR, GPS, IMU, and others.

The goal of the presented simulated mission including the operation of four robots (two different platforms), and radiation hotspot localization was not to demonstrate advanced navigation and localization algorithms, but rather to introduce an overall concept. We plan to utilize this approach to test high-level algorithms for mission control during scenarios comprising a swarm of UASs fulfilling reconnaissance goals, and thus prevent potential failures and damages of real robots.

Acknowledgments. The research was funded from the Ministry of the Interior of the Czech Republic (MVCR) grant no. VJ02010036 (An Artificial Intelligence-Controlled Robotic System for Intelligence and Reconnaissance Operations).

References

1. Cihlar, M.: Simulation of robotic search of lost radiation sources. Master thesis. Brno (2022)
2. 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), vol. 3, pp. 2149–2154 (2004)
3. Baca, T., et al.: The MRS UAV system: pushing the frontiers of reproducible research, real-world deployment, and education with autonomous unmanned aerial vehicles. *J. Intell. Robot. Syst.* **102**, 26 (2021)
4. Wright, T., West, A., Licata, M., Hawes, N., Lennox, B.: Simulating ionising radiation in gazebo for robotic nuclear inspection challenges. *Robotics* **10**, 86 (2021)
5. Afzal, A., Katz, D.S., Goues, C.L., Timperley, C.S.: A study on the challenges of using robotics simulators for testing (2020)
6. Collins, J., Chand, S., Vanderkop, A., Howard, D.: A review of physics simulators for robotic applications. *IEEE Access* **9**, 51416–51431 (2021)

7. Takaya, K., Asai, T., Kroumov, V., Smarandache, F.: Simulation environment for mobile robots testing using ROS and Gazebo. In: 2016 20th International Conference on System Theory, Control and Computing (ICSTCC), pp. 96–101 (2016)
8. Quigley, M., et al.: ROS: an open-source Robot Operating System. Presented at the ICRA Workshop on Open Source Software, Kobe, Japan, p. 5 (2009)
9. Marcon, P., Janousek, J., Kadlec, R.: Vision-based and differential global positioning system to ensure precise autonomous landing of UAVs. In: 2018 Progress in Electromagnetics Research Symposium, Toyama, pp. 542–546 (2018)
10. Nohel, J., Flasar, Z.: Maneuver control system CZ. In: Mazal, J., Fagiolini, A., Vasik, P. (eds.) MESAS 2019. LNCS, vol. 11995, pp. 379–388. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-030-43890-6_31
11. Nohel, J., Zahradnicek, P., Flasar, Z., Stodola, P.: Modelling the manoeuvres of ground reconnaissance elements in urban areas. In: 2021 Communication and Information Technologies (KIT), pp. 1–6 (2021)