A ROS Gazebo Plugin Design to Simulate RFID Systems

ABDUSSALAM A. ALAJAMI (Student Member, IEEE), GUILLEM MORENO, AND RAFAEL POUS (Senior Member, IEEE)
Department of Communications and Information Technology, Universitat Pompeu Fabra, Campus del Poblenou—UPF, 08018 Barcelona, Spain
Corresponding author: Abdussalam A. Alajami (abdussalam.alajami@upf.edu)

ABSTRACT Simulation, robotics, and Radio Frequency Identification (RFID) technology have significant roles in the new industrial revolution and their applications are key aspects of making Industry 4.0 a reality. Developing efficient use cases in Industry 4.0 almost always requires accurate simulation tools to be used in the digital world. The problem of simulating RFID readers for robotics in environments where high populations of RFID tags exist is addressed in this paper. This paper will discuss the design of an RFID system plugin based on Robot Operating System (ROS) and Gazebo simulator and the probability-based model on which the plugin is based. To assess the performance of the proposed system model, the simulation results of the designed plugin are compared with experiments. We also prove that the proposed simulator is flexible enough to be used on any robot platform, including aerial and ground robots. We show initial results of the simulation of having an Unmanned Aerial Vehicle (UAV) and a Unmanned Ground Vehicle (UGV) equipped with an RFID reader, navigating in an environment in which RFID tags have been placed. The robots will be reading tags in different map layouts using RFID antennas, with different orientations. We compare the simulation and experimental results in terms of the total unique tag readings vs. time, for various map-layouts. Finally, we show how this plugin can be used in robotics research by using it to simulate a novel, RFID-based stigmergic navigation strategy. We illustrate, the accurate navigation of the UAV using the proposed plugin.

INDEX TERMS Gazebo, industry 4.0, inventory, KEONN, retail, RFID plugin, RFID technology, robotics, ROS, UAV.

I. INTRODUCTION
Robotics and automation are quickly becoming one of the main success factors in e-commerce and Industry 4.0. They have a very big impact on the world of logistics. The number of multipurpose industrial robots developed and designed in the Industry 4.0 for Europe alone has almost doubled since 2004 [1]. An essential aspect of Industry 4.0 is autonomous production, warehouse and inventory management methods powered by robots that can complete tasks intelligently, with the focus on safety, flexibility, versatility, cost effectiveness and collaboration. Without the need to isolate its working area, its integration into human workspaces becomes more economical and productive, and opens up many possible applications in the industry [2]. Industry 4.0 technology intends among other goals to revolutionize Inventory Management. New technologies are already transforming how businesses is approaching inventory management. AI algorithms, IoT-powered tracking systems and robots can optimize existing inventory management processes and streamline business planning [3]. Simulations has an important role in Industry 4.0. It leverages real-time data to mirror the physical world in a virtual model, which can include robots, products, humans, and entire warehouses. This allows operators to test and optimize the environment and robot configuration in the virtual world before...
deployment, thereby driving down setup costs and increasing quality. It could also aid retail companies to evaluate the risks, costs, implementation barriers, impact on operational performance, and roadmap toward Industry 4.0 [4]. For much of the robotics community, the Open-source Gazebo [5] robot simulator is a fundamental tool in the development of ground and aerial robot applications for indoor and outdoor environments. The Gazebo simulator allows users to extend its functionality with user-defined plugins. By using an API, plugins have access to the simulation objects and data, can transmit information via topics by using Protocol Buffer [6] messages, and apply torques and forces to objects in the simulation scenario. The plugins must be initialized with a robot, sensor, or world model. For applications using ROS [7], the robot descriptions are specified in the SDF [8] or URDF [9] file formats.

Gazebo supports multiple interfaces, allowing users to interact programmatically with the simulation, including C++, a custom network transport, and ROS messaging.

The use of RFID technology with robotics has recently attracted a lot of attention both in the academic community and in the industry [10], especially in logistics and retail ([11], [12], [13], [14]).

In this paper we propose a solution for the problem of simulating RFID readers for robotics in environments where high populations of RFID tags exist. We propose a design of an RFID system plugin based on ROS and Gazebo simulator and the probability-based model on which the plugin is based. This paper is structured as follows: Section II, presents the state of the art. Section III, present the explanation of a simplified, effective, and fairly accurate Probability of Detection (PD) model on which the plugin is based. Section IV, presents the architecture and model integration of the designed plugin in ROS. Section V, present the environment layouts and robots that were used to conduct the experiments illustrated in this paper. The results of the experiments in the laboratory and simulations are described in Sections VI and VII. Section VIII, summarizes the conclusions of this work, and Section IX, presents the future work.

II. RELATED WORK

Simulating environments before deploying them when large quantities of RFID sensors and robots are used saves a lot of time and cost. The simulation of these environments helps understand the weaknesses of the designed layout before its deployment in the physical world. Such environments can be warehouses, retail shops, etc. Having a simulation platform that allows to test different types of mobile robots not only helps to assess and optimize the environment, but also can be used to validate the performance of these robots, the planning/navigation algorithms/strategies, and the RFID-related algorithms in the presence of RFID technology. Very few RFID simulators have been designed in the past. There has been related work by other researchers in the RFID domain on the simulation of RFID systems. Han et al. in [15], developed a system model of UHF RFID with a strong focus on the RF/analog design of the RFID reader. The model presented by Han et al. models the signal generation in the reader to verify whether the signal transmitted complies with the specified spectrum mask in the radio regulations. There is also a detailed model of the receiver part of the RFID reader that further analyses the effect of transmitter/receiver coupling. The wireless channel in their simulator is modeled as the vector addition of various multipaths. Authors in [16], present an RFID simulation engine, called RFIDSim, which implements the ISO 18000-6C communication protocol [17] and supports path loss, fading, backscatter, capture, and tag mobility models. They show that RFIDSim can be used to simulate large populations featuring thousands of RFID tags. Their model also simulates the deep fades that lead to frequent power losses of the passive RFID tags by modeling the multipath effects statistically. RFIDSim aimed to facilitate the relative comparison of different transmission control strategies. An approach in [18], similar to RFIDSim, proposed a simulation platform that relies on a discrete event simulator, designed also to implement a part of the ISO 18000-6C communication protocol supporting path loss, backscatter, capture, and tag mobility models. These models however are either too old, hard to adapt to robotics simulation platforms, or no longer available, and more importantly focused on the low-level communication issues between tags and reader antennas. Only a few of these models allow environment remodeling, design or manipulation in real time. Most are not open source. Finally, none of the simulators can be easily adapted to work with robots or the tools that comes with ROS. During the development of the proposed simulator in this paper, a study was published by the authors in [19]. The authors propose a simulator that is implemented as a Gazebo plugin integrated with ROS. A tag localization algorithm that uses the phase unwrapping technique and hyperbolae intersection method employing a reader antenna mounted on a mobile robot is used to estimate the position of the tags deployed in the presented scenarios. The user needs to specify the frequency, the range, the phase noise, and the gain of the tag antenna, as well as physical parameters, like damping coefficient and friction. The outcomes of their experiments showed realistic results for environments with a low number of tags (up-to 10). However, it is not known how the model will behave when simulating real environments with large populations featuring thousands of RFID tags. The illustrated experiments comparing the simulator’s performance with real life do not test the behavior of the simulator using different types of robots, assess the accuracy of RFID tag detection in different altitudes, nor was it tested in different environments with different tag densities. Having large populations of RFID tags in an environment introduce more parameters that may reduce the accuracy of the model. Cross interference is another type of interference that can degrade the performance of the simulator. It is most likely to occur between RFID systems and WIFI or personal area networks (WPAN) such as Bluetooth but only when devices share common or adjacent frequency bands within the environment. In [20], authors
propose an interference avoidance scheme which requires the knowledge of the theoretical maximum collision time and collision probability between RFID and WiFi/Bluetooth packets. This scheme generates an optimal channel based on the current usage of the adjacent frequency channels thereby reducing the interference. We conclude from the above, that it is extremely complex to consider all the parameters that effects the interrogated wave in a single algorithm, acquiring an exact estimation of whether a tag is detected or not by a reader, especially in environments where large populations of tags exist. In this paper, we introduce a simulation tool that simulates RFID systems in Gazebo. This plugin uses a different approach than previously mentioned models. The model does not aim to estimate the detection of a single RFID tag. It rather estimates the probability that the placed RFID tag can be detected or not considering various environment and sensor parameters. This model unlike it’s counters is aimed to simulate different types of robots in environments where hundreds or thousands of tags are to be detected. This makes it ideal to simulate retail shops or big warehouses, where autonomous inventory robots would need to be tested in such environments before deployed. The Gazebo environment is a very useful simulation tool for the broad audience in the robotics community, however, it is computationally costly and demands high system specification requirements. Complicated environments where large quantities of RFID tags (hundreds to thousands) are required to be detected, spawned, and managed can be very difficult to simulate. It may require special and expensive machines to be able to run the environment using the simulators discussed from above. Due to the simplicity of the proposed plugin, it enables large population of RFID tags to be simulated within the environment. In order for the plugin to work properly, a calibration phase to adapt the model to the environment is required, considering and encapsulating in a simplified manner most characteristics and interference parameters in the environment that are hard to consider. In this study, we run also multiple experiments comparing and validating the performance of the simulator with different laboratory environments. The proposed plugin is not designed to work for a specific sensor or a specific manufacturer. It is designed to be general for any RFID reader and with any robot in any environment. It is aimed to reach the broad audience of the robotics community; therefore, it is Open Source, and already published [21] in the main ROS Wiki, website for ROS platform users, with about 370 downloads in the month of the submission of this paper.

III. THE PROPOSED RFID SYSTEM MODEL

A. MODEL OVERVIEW

The proposed model is based on estimating a PD for each tag by each RFID reader antenna. This probability depends on the relative position and orientation of the tag with respect to the antenna, so it must be recalculated every time this relative position changes, as the robot moves. The goal of the model is not to estimate whether a specific tag is read or not, but to estimate how many tags will be read from a given constellation of tags. When we compare with experiments, we will not compare tags on an individual basis, but we will compare whether the simulation and the experiment have read approximately the same number of tags, and at a similar rate (tags read per second), which is the way accuracy is calculated in RFID deployments.

The PD is defined for each tag-antenna pair as a function of 6 arguments. The first 3 are the distance and two angle coordinates of the tag with respect to antenna position and its direction of maximum radiation: $R$, $\theta_H$, and $\theta_V$, which in Gazebo are the representation of the transforms between the sensor-frames [22] “RFID-Antenna” and “RFID-Tag”, which are automatically calculated by ROS.

The remaining 3 parameters are constants that depend on the particular RFID system used and its RFID settings. These parameters are $R_0$, the distance at which half of the tags can be read during a specific duration of time, and the antenna beam widths in the horizontal and vertical planes, $\Delta \theta_H$ and $\Delta \theta_V$. $R_0$ depends mostly on the Equivalent Isotropic Radiated Power (EIRP) which is defined as the product of the conducted power ($P_m$) and the antenna gain ($G_t$), $EIRP = P_m \cdot G_t$, as well as the sensitivity of the tags. $\Delta \theta_H$ and $\Delta \theta_V$ depend on the particular reader antenna used in the system. These 3 parameters must be supplied by the user of the plugin. The antenna beam-widths are normally found in the data sheet of the antenna used in the system, and $R_0$ must be adjusted by calibrating the simulation against some experiments.

$R_0$ is defined so that when $R = R_0$ and $\theta_H = \theta_V = 0$ the probability of detection is $PD = 0.5$. At other distances and angles, $PD$ is calculated using the antenna pattern based on the beam-widths, and the $1/R^2$ decay of surface power density.

Increasing $R_0$ (by increasing the EIRP, the gain of the reader antenna, and/or the sensitivity of the tags), will allow the antenna to detect distant tags with higher probability, while decreasing it will tend to allow only the detection of tags at shorter distances. On the other hand, using antennas with wider beam-widths will allow the antenna to detect tags at wider angles from the front direction of the antenna.

B. MODEL DEFINITION

An antenna’s radiation pattern describes how the antenna radiates/receives energy into/from all directions in space, and is three-dimensional. In the model, we approximate the normalized (maximum value of 1) radiation pattern as the function $D_0(\theta_H, \theta_V, \Delta \theta_H, \Delta \theta_V)$ in Eq. 1.

$$D_0(\theta_H, \theta_V, \Delta \theta_H, \Delta \theta_V) = \begin{cases} 
\cos^2\left(\frac{\pi}{2} \cdot \frac{\theta_H}{\Delta \theta_H}\right) & \text{if } \theta_H \text{ and } \theta_V \leq \frac{\pi}{2} \\
\cos^2\left(\frac{\pi}{2} \cdot \frac{\theta_V}{\Delta \theta_V}\right) & \text{if } \theta_H > \frac{\pi}{2} \\
0 & \text{otherwise}
\end{cases} \tag{1}$$
Note that:
- \( D_0(0, 0, \Delta \theta_H, \Delta \theta_V) = 1 \)
- \( D_0(\pm \Delta \theta_H/2, 0, \Delta \theta_H, \Delta \theta_V) = \frac{1}{2} \)
- \( D_0(0, \pm \Delta \theta_V/2, \Delta \theta_H, \Delta \theta_V) = \frac{1}{2} \)
- \( D_0(\pm \Delta \theta_H/2, \pm \Delta \theta_V/2, \Delta \theta_H, \Delta \theta_V) = \frac{1}{4} \)

This approximation is valid for antennas with a well-defined main lobe, and considers any radiation in the back hemisphere as negligible.

Approximating the tag antenna as isotropic, and neglecting any multipath interference, the received power by the antenna is proportional to the reader antenna directivity and to \( \frac{1}{R^2} \), as shown in Eq. 2.

\[
P_{\text{rec}}(R, R_0, \theta_H, \theta_V, \Delta \theta_H, \Delta \theta_V) \propto D_0(\theta_H, \theta_V, \Delta \theta_H, \Delta \theta_V) \cdot \frac{R^2}{R_0^2} \tag{2}
\]

Given that the probability of detection \( p_d(R, R_0, \theta_H, \theta_V, \Delta \theta_H, \Delta \theta_V) \) must be defined so that:
- \( p_d(0, 0, 0, \Delta \theta_H, \Delta \theta_V) = 1 \)
- \( p_d(R_0, 0, 0, \Delta \theta_H, \Delta \theta_V) = \frac{1}{2} \)
- \( p_d(\infty, R_0, 0, 0, \Delta \theta_H, \Delta \theta_V) = 0 \)

We arbitrarily define it as:

\[
p_d(R, R_0, \theta_H, \theta_V, \Delta \theta_H, \Delta \theta_V) = B(x) = \frac{2}{1 + 3\sqrt{x}} \tag{3}
\]

where \( x \) is always positive, and is defined as:

\[
x = \frac{P_{\text{rec}}(R_0, 0, 0, \Delta \theta_H, \Delta \theta_V)}{P_{\text{rec}}(R, R_0, \theta_H, \theta_V, \Delta \theta_H, \Delta \theta_V)} \cdot \frac{R_0^2}{R^2} \tag{4}
\]

Any other smooth function \( B(x) \) so that \( B(0) = 1 \), \( B(1) = \frac{1}{2} \), and \( B(\infty) = 0 \) will give very similar results (e.g., \( B(x) = \frac{2}{1 + 3\sqrt{x}} \), \( B(x) = 2^{-\sqrt{x}} \), \( B(x) = 2^{-x^2} \)), but Eq. 4 has shown the best results when compared with experiment.

Fig. 1, show a 3D visualization of the \( PD \) function using different values for each parameter.

IV. RFID SYSTEM PLUGIN ARCHITECTURE

The RFID system Gazebo plugin is composed of two independent plugins: the RFID-Tag plugin and the RFID-Antenna plugin. Both plugins are called from an SDF model file. Depending on the application scenario, they can be attached to the robot SDF model, or the world model, which represents the environment in Gazebo.

A. RFID TAG PLUGIN

As in real life, RFID tags are placed in the environment attached to products, boxes, shelves or any other objects. Each simulated RFID tag will be spawned and represented as very small cubes. RFID tags will be spawned in groups, associated to box-like transparent object. This object is referred to as a fixture. The user will have the option to either load a costume map of fixtures, or automatically generates a semi costume map by adjusting some parameters. The costume map is composed of a file that contains the fixture’s id and coordinates in the map. The user can choose to load up to 10,000 fixtures and tags in an environment. The automatic map generator parameters are the number of fixtures, their height, their associated number of RFID tags, and the spawn range, which can be accessed through the parameter list in the SDF file that Gazebo API permits. The RFID-Tag plugin allows the graphical change of the fixture and tag model designs. Real time manipulation of the fixtures and tags positions is allowed. The spawned RFID tags will be connected to the \textit{odom_sim} frame which represents the map frame or the world in Gazebo.

B. RFID ANTENNA PLUGIN

Similar to the RFID-Tag plugin, the antenna model in the RFID-Antenna plugin is spawned by loading its SDF model file. A ROS frame will be generated at the same moment the antenna model is spawned in Gazebo, and the RFID antenna child frame will be connected to a parent frame as defined by the user. In the case of the robots used in our experiment, the antenna frames are directly connected to the robot frame \textit{base_link}. The RFID-Antenna plugin inherits the probability model function, as explained in the \( PD \) model section. The \( PD \) requires the distance, azimuth and co-elevation angles of each tag \( (R, \theta_H = \pi, \theta_V) \) to produce a probability of detection value. These parameters are provided by the direct processing of relative transforms between the RFID antenna pose and the RFID tag pose. This process is done in real time during the start of a mission. In Figs. 2, 3a, and 3b, we show the transform tree between the RFID tag, the RFID antennas, the robot frame, the sensor frames, the odom, and the map-world frame.

![Figure 1. PD for various parameter sets.](image-url)
A. A. Alajami et al.: ROS Gazebo Plugin Design to Simulate RFID Systems

FIGURE 2. Frame tree from RQT-ROS.

V. ENVIRONMENT LAYOUTS AND ROBOTS USED IN THE EXPERIMENTS

A. ROBOTS USED IN THE EXPERIMENTS

1) UNMANNED GROUND VEHICLE (UGV)

The pre-designed and patent UGV in [23] is used. This UGV is used to carry the RFID payload, and its hardware block diagram is shown in Fig. 4. The UGV is designed to autonomously navigate within the given path in the intended map layout. The RFID payload consists of a Keonn Advantenna SP11 [25] RFID antenna placed each in a different orientation.

2) UNMANNED AIRED VEHICLE (UAV)

UAVs have evolved a great deal in the last several years in terms of technology (e.g., autopilots, sensors, power efficient motors, battery capacities and sizes) [26], [27], enabling them to be used for different purposes and applications including RFID-based inventory. In addition to the UGV, a custom-designed UAV was built specially for the purpose of aerial based inventory, and to evaluate the performance of the plugin compared to real laboratory environments. It is used to carry the RFID payload for laboratory and tests. The UAV is composed of 3 main hardware blocks:

1) Block 1 (B1): The Main flight system: The UAV was designed to navigate indoors and to carry a heavy payload, therefore, a 6-motor UAV (hexacopter) frame was chosen. This chosen design layout enables a stable flight in indoor spaces during a task-sufficient flight time. An efficient open-source autopilot called (Pixhawk) that is compatible to operate with a companion computer was used. This block is responsible for most physical and mechanical properties required for hovering, and contains 6 motors, 6 propellers, and 6 electronic speed controllers (ESCs). The ESC’s main objective is to translate the signal received from the autopilot to energy from the energy source and supply it to the motors. The energy source is composed of a lithium polymer battery (LIPO) with the capacity of 8000mah, operating voltage 26v, and with high energy discharge rate.

2) Block 2 (B2): Sensors and Processing Units: Most inventory warehouses or retail shops, are indoor spaces or so called GPS Denied Environments. This led us to select a visual Simultaneous Localization and Mapping (SLAM) based camera to supply the UAV with self-localization coordinates. Special adaptation was made to infuse these coordinates to the autopilot, resulting in an indoors guidance system for the UAV. This part is responsible for adding intelligence to the contiguous main flight system part. It also enables the possibility to sense the environment and obstacles nearby through a depth proximity camera. All these sensor data, including the data received from the RFID-Payload, are processed by a companion computer (CC). The CC will incorporate the navigation algorithm. This same algorithm will be used for the simulation and laboratory experiments. The output of this CC will be mainly the control signals in the form of pose-goal, or movement commands to the autopilot.

3) Block 3 (B3): Payload: The UAV will carry the same payload as the UGV but on the bottom of its structure rather than on top. It is composed of a light-weight structure skeleton embedding the RFID-reader, the power converter/distributor circuit board,
the RFID-reader and the 4 antennas. Each antenna will be facing a different direction. This block provides all the RFID-related data to other blocks, if needed. We can see its block diagram in Fig. 5.

\[\text{FIGURE 5. Hardware block diagram of the UAV.}\]

B. LABORATORY AND SIMULATED ENVIRONMENTS LAYOUTS

In this section we present the scenarios used for both robots in simulation and experiment:

1) SCENARIO 1: ONE-SIDED, UNIFORMLY-PLACED TAGS

In Scenario 1, 8 fixtures are placed at a fixed distance away from the robot’s path. 30 tags are placed within each fixture. Figs. 6 and 7 show the scenario map layout in the laboratory and in simulation, each associated with both each robot: the UGV and the UAV.

\[\text{FIGURE 6. Laboratory and simulation setups of Scenario 1 with the UGV.}\]

2) SCENARIO 2: LOW-DENSITY UNIFORMLY-DISTRIBUTED TAGS IN A SQUARE-SHAPED MAP LAYOUT

The RFID system plugin would need to prove its performance and accuracy in more complex environments, with different orientation of the tags with respect to the antennas. Therefore, we designed a square-shaped map layout with 600 RFID tags in Scenario 2. The tags were placed uniformly at each side of the square map, with some of the tags in the middle. The laboratory and Gazebo simulation setups for Scenario 2 are shown in Fig. 8.

\[\text{FIGURE 7. Laboratory and simulation setups of Scenario 1 with the UAV.}\]

\[\text{FIGURE 8. Laboratory and simulation setups of Scenario 2 with the UAV.}\]

3) SCENARIO 3: INCREASED TAG DENSITY, UNIFORMLY-DISTRIBUTED TAGS IN A SQUARE-SHAPED MAP LAYOUT

Scenario 3 was used to prove the performance and robustness of the RFID system plugin in a higher-density situation. This was done by doubling the density of tags in Scenario 2 to 1,200 tags, as shown in Fig. 9. This scenario also tests the performance of the plugin having placing tags in different positions in the z axis.

\[\text{FIGURE 9. Laboratory and simulation setups of Scenario 3 with the UGV.}\]

VI. COMPARISON OF SIMULATED AND EXPERIMENTAL RESULTS

For all the Experiments in these Scenarios, both robots will be tested with a constant velocity of \(1\text{m/s}\). However, lower or higher velocities could be used. It is important, to take into consideration the differences and the effects that the actual physical environment has on both robots as compared in simulation. These effects could be slight odometry or wheel misplacement errors due to the ground surface for a UGV, to small displacements in the UAV position, while
constantly trying to stabilize itself in the air, this is due to the generated air turbulence from the propellers and the quality of the localization messages from the visual SLAM sensor.

4) EXPERIMENT 1. SCENARIO 1 WITH THE UAV
In Experiment 1, we will use the UAV in Scenario 1 to test the performance of the RFID system plugin. We compare the results obtained from the simulation with the ones obtained from running the missions in the laboratory. We first place the UAV at a defined starting position at a distance \( d \) from the fixtures. The UAV will take off from that position and navigate in a straight line. We repeat these missions for different values of the distance \( d \). We run these missions with only the right antenna active (the one facing the fixture). Fig. 10, illustrates the Experiment 1 setup. Figs. 11a and 11c, shows the simulation results at \( d = 1.75m \) and \( d = 3.5m \) respectively. Whereas Figs. 11b and 11d, show the Lab results at those distances. Figs. 12a and 12b, show plots that compare the RFID tag readings vs. time in Experiment 1 in simulation and the in the laboratory, showing a remarkable agreement both in the total number of tags read as well as in the rate at which they are read.

5) EXPERIMENT 2. SCENARIO 1 WITH THE UGV
In Experiment 2, we repeat Experiment 1 but using a UGV. The Simulation results are shown in Figs. 13a and 13c, while Figs. 13b and 13d, shows the results of Experiment 2 in the laboratory. Figs. 14a and 14b show the RFID tag readings vs. time in Experiment 2 in simulation and the in the laboratory, again showing a remarkable agreement.

6) EXPERIMENT 3. SCENARIO 2 WITH THE UAV
This experiment is designed to validate the plugin’s performance in a slightly more complex environment. We compare the results obtained from the simulation and the laboratory when the UAV is navigating throughout Scenario 2. Since the scenario map layout is not a straight line as in Scenario 1, the effect of the orientation of the RFID antenna on the total number of unique RFID tags read is more relevant. For this reason, we run Experiment 2 four times, and for each one we activate only one of the four RFID antennas mounted on the UAV with different orientations, as illustrated in Fig. 5. Figs. 15a, 15b, 15c, and 15d, shows the obtained simulation and laboratory results for each antenna orientation, validating the simulation model once again, in this more complex scenario.

7) EXPERIMENT 4. SCENARIO 2 WITH THE UGV
In Experiment 4, we repeat Experiment 3, but with replacing the UAV with the UGV. Figs. 16a, 16b, 16c, and 16d, illustrate the obtained simulation vs. laboratory results for each antenna orientation. In this experiment the agreement between simulation and experiment is not as good, especially for the side antennas. The reason could be due to the
difficulty of precise navigating the UGV within the path which relies only on the wheel odometry sensor for Dead Reckoning navigation. The UGV was slightly closer to the right tags then the left compared to the simulation. This can be noted from the Figs. 16b and 16d, where more tags are read by the right antenna compared to the left in the laboratory.

8) EXPERIMENT 5. SCENARIO 3 WITH THE UAV
The goal of Experiment 5 is to test the performance of the RFID system plugin when used in an environment with a higher density of tags and having tags and fixtures placed in different positions in the z-axis. We compare the obtained results from the simulation and the laboratory, having the UAV navigating throughout Scenario 3. As in Experiment 4, we repeat Experiment 5 four times, each with one of the four antennas active. We observe in Figs. 18a, 18b, 18c, and 18d, that results are in good agreement for all four antennas even though the tag density was doubled. Figs. 17a, 17c, 17e, and 17g, illustrate the position of the detected RFID tags in simulation, while Figs. 17b, 17d, 17f, and 17h, illustrate the position of the detected RFID tags in the laboratory, which are also in very good agreement.

We can notice in Figs. 17g and 17h, that with only the left antenna active on the UAV, the majority of the detected unique RFID tags were located on the edges of the map environment. On the other hand, in Figs. 17c, and 17d, having only the right antenna active, we see that most of the detected tags
FIGURE 17. Experiment 5: Position of the detected RFID tags in simulation and in the laboratory.

FIGURE 18. Experiment 5: Simulation vs. laboratory unique RFID tag readings.

FIGURE 19. Comparing laboratory vs. simulation UAV paths.

9) EXPERIMENT 6. SCENARIO 3 WITH THE UGV
Finally, we repeat Experiment 5 but substituting the UAV with the UGV. The results from Experiment 6, shown in Figs. 21a, 21b, 21c, and 21d, show realistic and very good agreement with the exception of the right antenna. Figs. 20a, 20c, 20e, and 20g, illustrate the position of the detected RFID tags in Gazebo, while Figs. 20b, 20d, 20f, and 20h, show the detected RFID tags in the laboratory, again with realistic and good agreement.

VII. AN EXAMPLE OF THE USE OF THE PLUGIN IN ROBOTICS RESEARCH: STIGMERGIC NAVIGATION OF A UAV
The ability for the RFID system plugin to simulate accurately the behavior of an RFID system payload (reader and antennas) on board an operating mobile robot in an environment where RFID tags are present, enables it to be used in research that involves RFID and Robotics

were in the fixtures placed in the middle. Some differences can be observed in particular fixtures for other antennas, this is due to that the UAV navigation in the laboratory is not as smooth as in the simulation. As noted before, the UAV in the laboratory continuously tries to resist external drifting forces caused by turbulence generated by the propellers and localization accuracy, these forces could not be precisely simulated in Gazebo. The comparison of both paths of the UAV in the laboratory and simulation environments can be seen in Fig. 19.
technologies working together. For this application scenario, we utilize the RFID system plugin to enable a UAV to navigate using stigmergy [28] to inventory a space using RFID technology. In this scenario, we will have a UAV navigating autonomously through an environment where RFID tags are present, but whose quantity and position are unknown. The used stigmergic navigation technique consists of a robot navigating by choosing the direction in which a higher number of tags are detected for the first time. At every step the UAV measures the number of new tags read by each of the four antennas, and follows the direction of the antenna which reads more unique tags. Research on finding new UAV navigation strategies is expensive in time and money, and the possibility to run simulations to validate the algorithms can speed up the process considerably. But when RFID payloads are used, the simulation is only possible if a simulation tool is available for the RFID system, such as the RFID system plugin that we propose. For this simulation 300 tags were uniformly placed in a horizontal manner, throughout the T-shaped layout. The simulation and laboratory setups are shown in Figs. 22a and 22b. Fig. 22c shows the path of the UAV in simulation, and Fig. 22d shows that the UAV was able to read 283 tags out of 300 overall tags in the environment.
(94.33%). Robotics researchers interested in using the RFID Gazebo plugin in their research can find it in the ROS wiki page and repository [21].

VIII. CONCLUSION
In summary, this paper addresses the problem of simulating RFID systems, including readers, antennas and tags, being used by robots to navigate and perform other tasks, such as inventorying. The designed solution is composed of an easy to use ROS-Gazebo plugin based on a simple but accurate probability model that requires only 3 parameters from the user, only one of which must be calibrated ($R_0$) against measured results. The plugin was extensively tested by comparing simulation and experimental results in 3 different scenarios of increasing complexity and density of tags. In each of the three scenarios an experiment was done with both a UGV and a UAV. In all six experiments the simulation and experimental results were in enough agreement to use this simulation tool in robotics involving RFID sensors.

The plugin allows to simulate environments before deploying them especially when large quantities of RFID sensors and robots are needed to be used, which saves a lot of time and cost. The proposed plugin through simulation of these environments helps users understand the weaknesses of the designed layout before its deployment in the physical world. Such environments can be warehouses, retail shops, etc. The plugin allows robotics researchers to simulate environments in which robots and RFID technology interact for various applications. In summary, it is a powerful tool for researchers that use RFID technology to improve robots in any sort of manner. However, an extra layer of statistically measuring $R_0$ by the user is needed for the plugin to work as intended. The entire neglect of the back lobe and the consideration of the tag orientation is also considered a minor limitation for this version. The following section will briefly explain the future improvements to this plugin.

IX. FUTURE WORK
There are several way in which the accuracy of the Gazebo plugin could be improved:

1) Considering the pattern and orientation of the RFID tag antennas.

2) Using the actual radiation diagram of the reader antennas instead of approximating it based on the beam widths.

3) Automatizing the calibration of $R_0$ measurement, so that the users of the plugin do not have to define calibration procedures on their own.

These, together with improved documentation and examples of use will be the future work related to this Gazebo plugin.

REFERENCES
[1] R. B. S. Consultants, “New industrial revolution: How Europe will succeed. international conference the next industrial revolution manufacturing and society in the XXI century,” Industry 4.0, p. 24, Jan. 2014.
[2] M. A. K. Bahrin, M. F. Othman, N. H. N. Azli, and M. F. Talib, “Industry 4.0: A review on industrial automation and robotic,” Journal Teknologi, vol. 78, nos. 6–13, pp. 137–143, Jun. 2016. [Online]. Available: https://journals.utm.my/journalteknologi/article/view/9285
[3] X.-M. Yuan, “Impact of Industry 4.0 on inventory systems and optimization,” in Industry 4.0. T. Bányai and A. P. F. D. Felice, Eds. Rijeka, Croatia: IntechOpen, 2020, ch. 3, doi: 10.5772/intechopen.90077.
[4] W. de Paula Ferreira, F. Armellini, and L. A. De Santa-Eulalia, “Simulation in Industry 4.0: A state-of-the-art review,” Comput. Ind. Eng., vol. 149, Nov. 2020, Art. no. 106868. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835220305635
[5] L. Pitonakova, M. Giuliani, A. Pipe, and A. Winfield, “Feature and performance comparison of the V-REP, Gazebo and ARGoS robot simulators,” in Towards Autonomous Robotic Systems, M. Giuliani, T. Assaf, and M. E. Giannaccini, Eds. Cham, Switzerland: Springer, 2018, pp. 357–368.
[6] Protocol Buffer for Topics, ROS. Accessed: Oct. 29, 2021. [Online]. Available: https://developers.google.com/protocol-buffers/docs/overview
[7] R. Mittler, “ROS are good,” Trends Plant Sci., vol. 22, no. 1, pp. 11–19, Jan. 2017.
[8] SDF SDF Format Base Element Description. Accessed: 2021. [Online]. Available: http://sdformat.org/spec
[9] URDF Format, Gazebo ROS Wiki. Accessed: 2021. [Online]. Available: http://wiki.ros.org/urdf
[10] P. Mezzanotte, V. Palazzi, F. Alimenti, and L. Roselli, “Innovative RFID sensors for Internet of Things applications,” IEEE J. Microw., vol. 1, no. 1, pp. 55–65, Jan. 2011.
[11] M. Morenza-Cinos, V. Casamayor-Pujol, and R. Pous, “Stock visibility for retail using an RFID robot,” Int. J. Phys. Distrib. Logistics Manage., vol. 49, no. 10, pp. 1020–1042, Dec. 2019.
[12] M. Morenza-Cinos, V. Casamayor-Pujol, J. Soler-Busquets, J. L. Sanz, R. Guzmán, and R. Pous, Development of an RFID Inventory Robot (AdvantoRobot), Cham, Switzerland: Springer, 2017, pp. 387–417.
[13] A. Khazedinov, A. Aleksandrov, A. Zakiev, E. Magid, and K. H. Hsia, “RFID-based warehouse management system prototyping with a heterogeneous team of robots,” Robots Hum. Life, p. 263, Aug. 2020.
[14] R. Polvara, M. Fernandez-Carmona, G. Neumann, and M. Hanheide, “Next-best-sense: A multi-criteria robotic exploration strategy for RFID tags discovery,” IEEE Robot. Autom. Lett., vol. 5, no. 3, pp. 4477–4484, Jul. 2020.
[15] C. Floerkeemeier and S. Sarma, “RFDSim—A physical and logical layer simulation engine for passive RFID,” IEEE Trans. Autom. Sci. Eng., vol. 6, no. 1, pp. 33–43, 2009, doi: 10.1109/TASE.2008.2007929.
[16] C. Floerkeemeier and S. Sarma, “RFDSim—A physical and logical layer simulation engine for passive RFID,” IEEE Trans. Autom. Sci. Eng., vol. 6, no. 1, pp. 33–43, Jan. 2009.
[17] D. Dobkin, The RF in RFID: UHF RFID in Practice. London, U.K.: Newnes, 2012.
[18] T. Zhang, Y. Yin, D. Yue, Q. Ma, and G. Yu, “A simulation platform for RFID application deployment supporting multiple scenarios,” in Proc. 8th Int. Conf. Comput. Intell. Secur., Nov. 2012, pp. 563–567.
[19] S. D’Avella, M. Unetti, and P. Tripicchio, “RFID Gazebo-based simulator with RSSI and phase signals for UHF tags localization and tracking,” IEEE Access, vol. 10, pp. 22150–22160, 2022.
[20] T. Chi and M. Chen, “A frequency hopping method for spatial RFID/WiFi/Bluetooth scheduling in agricultural IoT,” Wireless Netw., vol. 25, no. 2, pp. 805–817, Feb. 2019.
[21] RFID Plugin Abdussalam-Alajami, Rafael-Pous, and Guillen-Moreno. RFID-Sensor Gazebo Plugin Description and Repository in ROS Wiki. Accessed: 2022. [Online]. Available: http://wiki.ros.org/RFIDsensor_Gazebo_plugin
[22] ROS Transforms and Frames Frames and Transform Description MiscSDF. SDF SDF Format Base Element Description. Accessed: 2020. [Online]. Available: http://sdformat.org/spec
[23] V. Casamayor-Pujol, M. Morenza-Cinos, B. Gastón, and R. Pous, “Autonomous stock counting based on a stigmergic algorithm for multi-robot systems,” Comput. Ind., vol. 122, Nov. 2020, Art. no. 103259. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361520304930
[24] Keonn’s AdvantReader 160. Accessed: 2022. [Online]. Available: https://keonn.com/components-product/advantreader-160/
[25] Keonn’s AdvantennaSP11. Accessed: 2022. [Online]. Available: https://keonn.com/components-product/advantenna-sp11/
[26] E. Sholes, “Evolution of a UAV autonomy classification taxonomy,” in Proc. IEEE Aerosp. Conf., Mar. 2007, pp. 1–16.
[27] A. A. Alajami, A. Vulpe, and O. Fratu, “UAVs for Wi-Fi receiver mapping and packet sniffing with antenna radiation pattern diversity,” Wireless Pers. Commun., vol. 92, no. 1, pp. 297–313, Jan. 2017.
[28] A. A. Alajami, G. Moreno, and R. Pous, “Design of a UAV for autonomous RFID-based dynamic inventories using stigmergy for mapless indoor environments,” Drones, vol. 6, no. 8, p. 208, Aug. 2022. [Online]. Available: https://www.mdpi.com/2504-446X/6/8/208

ABDUSSALAM A. ALAJAMI (Student Member, IEEE) received the bachelor’s degree in electrical and electronic/telecommunication engineering from the University of Tripoli, Tripoli, Libya, in 2012, and the M.Sc. degree in advanced wireless telecommunication from the Department of Electronics, Polytechnic University of Bucharest, Bucharest, Romania, in 2016. He is currently pursuing the Ph.D. degree in advanced robotics specializing in UAVs, collaborative, symbiotic heterogeneous robots with the University of Pompeu Fabra, Barcelona, Spain. He is a part of the Research Team with the Ubiquitous Computing Applications Laboratory, University Pompeu Fabra.

GUILLERMO MORENO received the bachelor’s degree in mathematical engineering in data science from University Pompeu Fabra, Barcelona, Spain, in 2022, where he is currently pursuing the master’s degree in intelligent and interactive systems. He is a part of the Research Team with the Ubiquitous Computing Applications Laboratory, University Pompeu Fabra.

RAFAEL POUS (Senior Member, IEEE) received the B.S. degree in telecommunications engineering and in computer science from the Technical University of Catalonia, Barcelona, Spain, in 1988, the M.Sc. degree in electrical engineering from the University of Massachusetts at Amherst, Amherst, MA, USA, in 1989, and the Ph.D. degree in electrical engineering from the University of California at Berkeley, Berkeley, CA, USA, in 1992. From 1993 to 2009, he was an Associate Professor with the Technical University of Catalonia. Since 2009, he has been an Associate Professor with Pompeu Fabra University, Barcelona. In parallel to his academic career, he has co-founded three technology start-up companies. His research interests include numerical methods in electromagnetism, applied superconductivity, and fractal antennas. As the Founder of the Ubiquitous Computing Applications Laboratory, his current research interests include ubiquitous computing and the Internet of Things applied to different industries, using radio frequency identification and robotics, among other technologies.

* * *