Proceedings of the 10th International Conference of Control Systems, and Robotics (CDSR'23)
Ottawa, Canada – June 01 - 03, 2023
Paper No. 202
DOI: 10.11159/cdsr23.202

Low-Cost Distance Sensor Characterization of Ultrasonic and Infrared to Evaluate the "Reality Gap" in Robot System Simulation

Bryan Van Scoy, Tra Yen Nhu Phan, Lam Ha, and Peter Jamieson
Miami University
Oxford, OH, USA
{bvanscoy, phant2, haln, jamiespa}@miamioh.edu

Abstract - Simulation of robotic systems is necessary to test for correctness and safety of these systems, but how accurate is simulation compared to the real system? In this work, we provide a methodology to characterize low-cost distance sensors (ultrasonic and infrared) and then program these sensors into our robot simulator. Our focus robotics system is a simple two-wheel robot mounted with either one of our two distance sensors, and we characterize our robot and define realistic parameters that can be simulated in our open-source software system — Centurion. The goal of this work is to characterize the similarity between our real and simulated systems, and progress to providing a confidence value of the "reality gap" of simulation over time. To do so, we provide results on how similar our real and simulated sensors are as quantified by the confidence of a Bayesian filter, and we run a simple control algorithm for object avoidance over a number of real and virtual trials to establish a measurement of simulation confidence as a function of time.

***Keywords*:** Simulation, Reality, Sensor Characterization, Actuator Characterization

1. Introduction

In 1963, Lorenz discovered and helped formulate Chaos Theory [1] that showed there are fundamental limitations of what a computer can simulate and what, actually, happens with a system in simulation versus the real world. Even with these shortcomings, the simulation of systems allows us to formulate and experiment with concepts that are not feasible with real world experiments given limitations such as cost, time, and safety. For these reasons, simulation is of fundamental importance in engineering design and helps us improve our understanding of a variety of systems including, our focus, mobile robotic systems.

Robotics simulation has developed and improved over the years to allow engineers to create, observe, and test robots in 2D and 3D environments with sophisticated physics simulation. We notice, however, that the work on trying to match up real-world behavior and simulation, called the "reality gap", for low-cost robotic systems lacks characterization methods for sensors and robot dynamics. We believe that starting with simple systems and understanding how to create a "simulation confidence" measure is a research path forward to providing these measures for more complex systems.

In this work, we provide a methodology to characterize low-cost distance sensors and include these characterizations in simulation software. Similarly, we create a methodology to characterize the simple dynamics of a two-wheeled mobile robot that moves with dead reckoning. Using our characterization methods, we show how our simple robots — shown in Figure 1 — have similar probabilistic behavior in both simulation and real experiments, suggesting that our fidelity between the simulated and the real-world system is close and can provide a confidence value of the "reality gap". The key characteristic of our robotic system is that it is a very low-cost system in terms of the unit cost per robot. While we focus on characterizing simple sensors and robot dynamics, we believe this is a fundamental starting point for characterizing and simulating more complex mobile robotic systems.

The contributions of this work are:

- A methodology to characterize distance sensors and simple dynamics of a two-wheeled mobile robot;
- Simulation software implementations of our characterizations of distance sensors and simple robot moving dynamics.
- An experimental setup that provides means to compare the traces of robotic movement to quantify and provide early measures that describe the similarity between real and simulated robot behavior.

The remainder of this paper is organized as follows: Section 2 provides background on robot simulation and sensor and movement characterization. Section 3 describes the overall assumptions for our robotic system including which aspects of the real world are included in our characterization methods. Section 4 describes how we characterize the robotic sensors and dynamics, including how we incorporate the characterizations into the simulator. Section 5 provides the experimental results of comparing the simulation to real-world behavior, and finally, Section 6 concludes the paper.

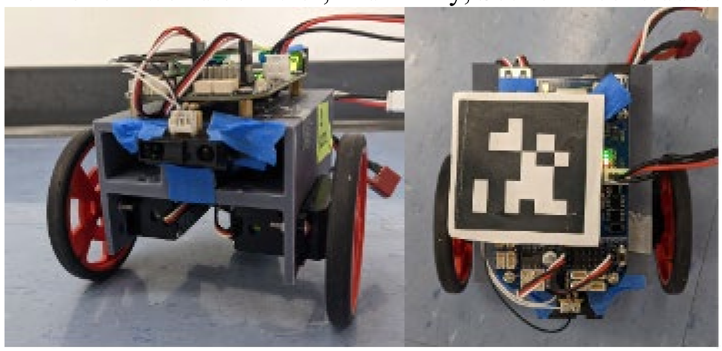


Fig. 1: A front (left) and overhead (right) view of our low-cost robot equipped with a distance sensor (ultrasonic or infrared) for sensing, servo motors for actuation, and an AprilTag [2] marker for detection by an overhead camera.

2. Background

In this section, we review existing robotic simulation software and related research, sensor characterization, and work examining the fidelity between real and simulated systems - "reality gap".

2.1. Robotic Simulation

A number of research efforts have been made to understand the state of robotic simulation across the years including an early survey in 2007 [3] which was followed by two surveys in 2011 [4] [5] that all describe a broad set of commercial and open source systems. In 2015, Ramli et. al. [6] looked at the complexity of simulation systems from a beginner perspective, and in 2018, Pitonakova et. al. [7] described a comparison of three specific simulators in terms of features and performance. Survey papers and comparison papers, as recent as 2021 [8] are strewn across the literature space and continue to be published on robotic simulation.

The breadth of robotic simulators leads us to pick a subset of popular simulation systems to discuss as references in this work. We cite three systems that have qualities similar to what we need: Gazebo [9], Webots [10], and Player/Stage [11]. Each of these simulation systems has been used and improved on since their inception in 2003 and 2004, and includes a strongly supported set of features and computational performance considerations. The capabilities of these systems are far beyond what our simulator can achieve in terms of features, but the main goal of our simulator is to help quantify the similarity between simulation and reality.

One of the key features in robotic simulation is how does virtual world operate, and typically, this is achieved via physics simulation of multiple rigid bodies or particle interactions over time. Collins et. al. [12] made a recent attempt at quantifying the actuation differences between physic simulators of robot movements compared to the real world for a number of physics simulators. Our work is less concerned with the accuracy of the physical world to start with but will attempt similar ideas to Collins's work, focusing on a simple low-cost 2D system that can ignore some of these more complex questions.

2.2. Sensor Characterization

Sensors of all sorts have been characterized where the process of sensor characterization is to evaluate and record the readings of a sensor under a set of experimentally fixed conditions. In the case of a distance sensor, such as an ultrasonic sensor in this work, the main characteristics of evaluation for an object include an object's distance, angle of incidence, and angle of offset relative to the sensor, as well as the material, color, and shape of the object. Figure 2 shows three measurement characteristics described diagrammatically. Additionally, characteristics such as sensor sampling rate and operational distance range are important characteristics of a sensor that are needed for simulation.

There is a significant number of publications on sensor characterization, and we will mainly look at research that focuses on characterizing distance sensors (with a focus on cheap simple sensors such as ultrasonic and infrared). The characterization of distance sensors (also referred to as range finders) starts with the design and understanding of these sensors in 1992 and includes ultrasonic characterization [13] [14], infrared characterization [15], and laser scanner characterization [16]. Most of these works focus on understanding a sensor's capabilities under different characteristics.

Beyond sensor characterization, sensor research, starting in the late 90s, changed focus to research such as map creating [14] [15], robot response [17], and simulation [18] [19] [20]. In all of these cases, the goals of the sensor provide constraints that simplify the characterization process.

Within the simulation space, there has been some work on providing sensor toolboxes [21], and our work differs slightly in that we are focused on characterization and then implementation of a small set of distance sensors.

2.3. Simulation and Reality gap

Lorenz [1] developed an entirely new line of mathematics based on the observed end results in weather simulation for small differences in initial conditions - Chaotic Systems. Two recent Ph.D. dissertations by Collins [22] and Murator [23] provide modern treatments of the challenges of matching robot simulation and the real world.

Brooks [24] provides an early discussion of the disconnect between simulating and training robots with the following statement that captures some of the goals of this work: "They (simulation) both rely on and at the same time generate, the dynamics of the interaction of the robot with the world. Simple state space models of the world do not suffice in the internal control program of a physically embodied mobile robot."

More recent work by Koos et. al. [25] continues to look at what they call the "reality gap" between evolutionary algorithms for control and the desired realistic robotic control algorithms. They invent an idea called transferable controllers where the goal is to evolve control algorithms in virtual space that can be used in both reality and simulation. The "reality gap" in many algorithmic design approaches is dealt with in simulation via two approaches: first, adding randomization in terms of sensor and dynamics noise [26] [27], and second, training AI to understand the weaknesses of the simulation [28] or using the results from multiple simulation software packages [29] (sometimes with real-in-the-loop).

Our work acknowledges Mouret and Chatzilygeroudis [28] suggestion that simulators are not perfect and instead should include a "confidence in simulation" value. This work attempts to progress toward this idea. There are continuing efforts on bridging the "reality gap" that focus on creating benchmarks [30] and providing users insight on how to practically deal with the disconnect [31]. Simulation in robotics is necessary because of the cost of real systems and performing experiments with them versus virtual systems. This work attempts to further narrow the gap by using a first-principle approach to simulation.

3. Our Robotic System and Assumptions

In this section, we describe our simple low-cost robot system including our control algorithm, describe the accompanying simulator, discuss our Bayesian filter for belief calculation, and finish with a description of assumptions we make in our method description and experiments.

3.1. Low-Cost Robot System

Our robot, pictured in Figure 1, is a two-wheeled robot with one distance sensor mounted on the front (either an ultrasonic (HC-SR04) or infrared (Sharp GP2Y0A41SK0F) the latter of which is mounted in the figure). The robot is controlled with a BeagleBone Blue. Our system is a simple 2D agent system similar in goals to Kilobot [32] and Robotarium [33]. Our system has an "overlord" computational system with an overhead camera that tracks AprilTags [2] on the robots.

3.2. Simulator — Centurion

Centurion is our open-source robotic system simulator (available and in development at https://github.com/drpaj12/CENTURION). Centurion allows each of the sensors and actuators, once characterized, to be described and programmed in C such that a robot can be simulated. From a simulation lens, the beam sensors in this work require simple collision detection techniques such as line segment-to-line segment intersections, and line segment-to-circle intersections that allow distance to be calculated and then passed into either an ideal sensor or characterized sensor function.

Centurion uses an XML file to describe the robots in the system and the parameters of the 2D simulation space. In terms of our robots, we assume robots are circles in the simulation where the bounding circle captures our robots' square dimensions of 11.5cm as a radius of approximately 8cm. Our field of experimentation is 91cm by 122cm wide, and later we will show an image of our experimental space in the experiment section. Finally, our simulation epoch is defined as

0.01 seconds which allows motion and sensor sampling rates to be scheduled in simulation. This assumes that the processing capabilities of the robot are faster than this epoch rate and that all decisions can be made in an epoch.

3.3. Other System Assumptions

As just described, our simulator attempts to capture the time and distance of the real robotic system, but performs a discrete time simulation of the system, and therefore, assumes that the computation power of the robot's processor is capable of completing everything per simulation epoch. Also, within the simulator, the sensor beams are considered to be line segments (θ is equal to 0 degrees) that do not take into account the angle of the object (ϕ is also assumed to be 0 degrees) as shown in Figure 2, below.

The boundaries of the experimental space are assumed to have nothing nearby and will not cause the sensors to return values. In a simulation, an ideal actuator and ideal distance sensor are implemented to move at a fixed velocity, turn radius, and sense the exact distance of objects. Our simulations are executed in C on a Linux Ubuntu 18.04 distribution on a virtual machine.

4. Sensor Characterization

We now describe the methodology we use to characterize the sensors in our simple robot. Motivated by [34], we construct probabilistic models of the sensors (ultrasonic and infrared).

Our robot uses distance sensors to detect nearby objects as shown in Figure 2. While the measurements may be influenced by a number of factors, we restrict our characterization to the distance d, angle of incidence θ, and angle of offset from the beam line ϕ to an object; additional factors that we do not model include the material, color, and shape of the object.

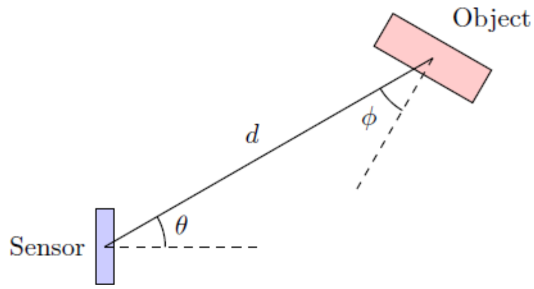


Fig. 2: A sensor measures an object at distance d, angle of incidence θ, and angle of offset from the beam line ϕ.

We denote the state of the robot as x, which consists of the position and orientation of the robot in the plane. Additionally, we let m denote the map of the environment, which consists of the location and orientation of all objects. Given the state x and map m, the robot can calculate the distance d and angles θ and ϕ to each object. Using these parameters, we model the probability of receiving a distance measurement z as a Gaussian random variable

$$p(z \mid x,m) \sim N(\mu, \sigma 2) \quad (1)$$

where the mean μ and standard deviation σ may depend on the parameters d, θ, and ϕ.

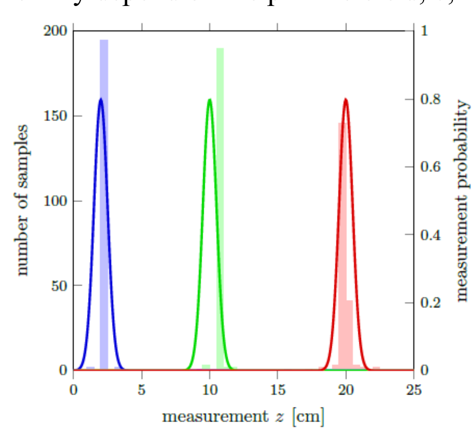
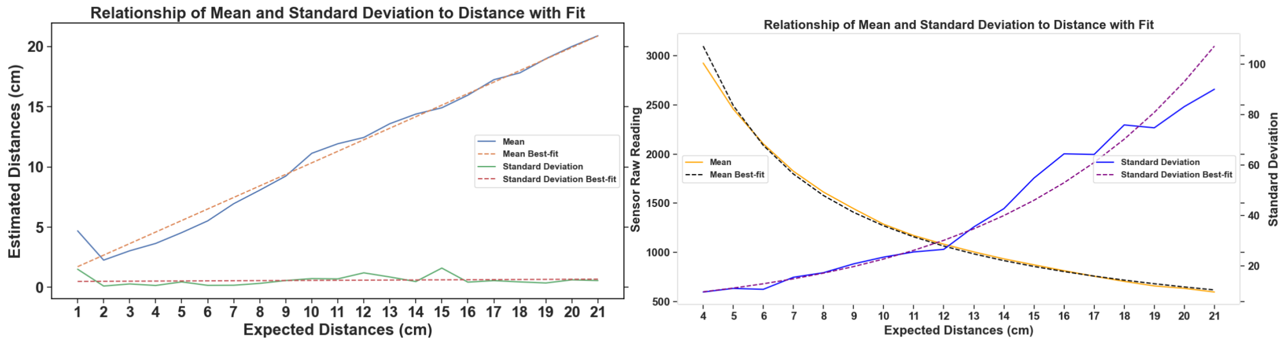


Fig. 3: Empirical measurement data and derived model for the ultrasonic sensor.



(a) (b)

Fig. 4: (a) Relationship of the mean (blue) and standard deviation (green) to distance with the fit (orange, red). (b) Relationship of the mean (blue) and standard deviation (orange) to distance with the fit (green, red)

4.1. Ultrasonic Sensor Characterization

Our goal is to characterize the ultrasonic sensor in the distance range of 1cm to 21cm. Due to the ultrasonic sensor's protocol, each measurement sample takes approximately 0.5 seconds to complete, and our first step is determining the number of samples at a set distance d to get consistent data sets.

To determine an appropriate number of samples to be used throughout further characterization, a specific distance of 10cm is chosen and we collect measurements for samples ranging in the set {10, 50, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1500, 2000} and examine histograms (bin size equal to 0.1cm) of the actual sensor data. We then note that additional samples do not significantly change the data set and use this number of samples for our characterization. The result of this experiment finds that 200 samples generate a data set that we use in the following collection of data for characterization. With the trial size set, we next characterize the data for an object at distances from 1cm to 21cm with 1cm increments and generate histograms of each set of measurements. From the generated histograms shown in Figure 3 (The histograms show 200 measurements for an object at a distance d of 2 cm (blue), 10 cm (green), and 20 cm (red). Also shown for each distance is the measurement probability function), we observe that the sensor data can be approximated by a Gaussian distribution.

For the ultrasonic sensor, the probability of a measurement z given the state x of the robot and a map m of the environment is given by (1), where the mean $\mu = 0.7626 + 0.9582d$ and standard deviation $\sigma = 0.4878 + 0.0093d$ are linear functions of the distance d to the nearest object with zero angle of incidence (found by ray casting) as depicted in Figure 2. The data and model are shown in Figure 4 (a).

4.2. Infrared Sensor Characterization

Following the same approach for the IR distance sensor, we found that due to the speed of a sensor reading, we could easily collect 10,000 samples per distance measure. We model the probability of a measurement z given the state x of the robot and a map m of the environment by (1), where the mean $\mu = 14001d-1.005+0.001\theta$ and standard deviation $\sigma = 3.78e0.157d$ are functions of the distance d to the nearest object with the angle of incidence θ. The data and model are shown in Figure 4 (b).

5. Experimental Results

Our first experiment attempts to compare the quality of our characterization of the distance sensors by comparing the confidence of our Bayesian filter in both real and simulation and comparing our two confidence values. In the second experiment, we have the robot progress in a simple experimental space and turn right to avoid hitting two objects, and in this experiment, we attempt to compare the ideal, real, and virtual paths as a first step to creating a confidence in simulation calculation. In both cases, the robot and simulator software is as described earlier with the respective system assumptions.

5.1. Comparing Distance Characterization with Bayesian Filter

In our first experiment, we compare the robot's belief about the distance to an object obtained using a discrete Bayes filter — described in Algorithm 1 (below) — in both simulation and real-world scenarios. In this experiment, we run our prediction model using the Bayes filter for 200 unique trials, where each trial contains 10 samples (or time steps t). We measure starting at 1cm to 21cm for both the ultrasonic and infrared sensors.

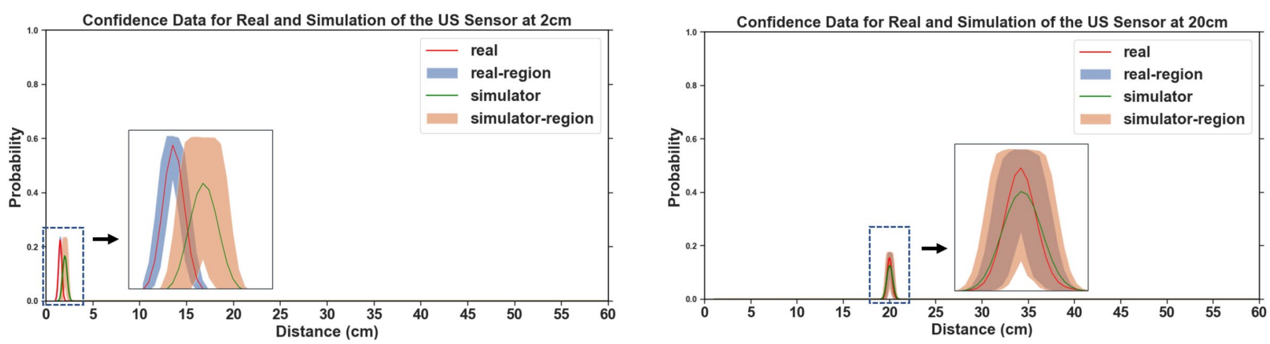
Algorithm 1 Discrete Bayes Filter

Input: Measurement model $p(z \mid x)$
Output: Belief $p_{k,t} = p(x_k \mid z_1, \ldots, z_t)$ of state x_k at time t

```
Initialize p_{k,0} = constant for each state index k
for each time t = 1,2,3, ... do
    Observe measurement z_t
    for each state index k do
        p_{k,t+1} = constant · p(z_t | x = x_k) p_{k,t}
    end for
end for
```

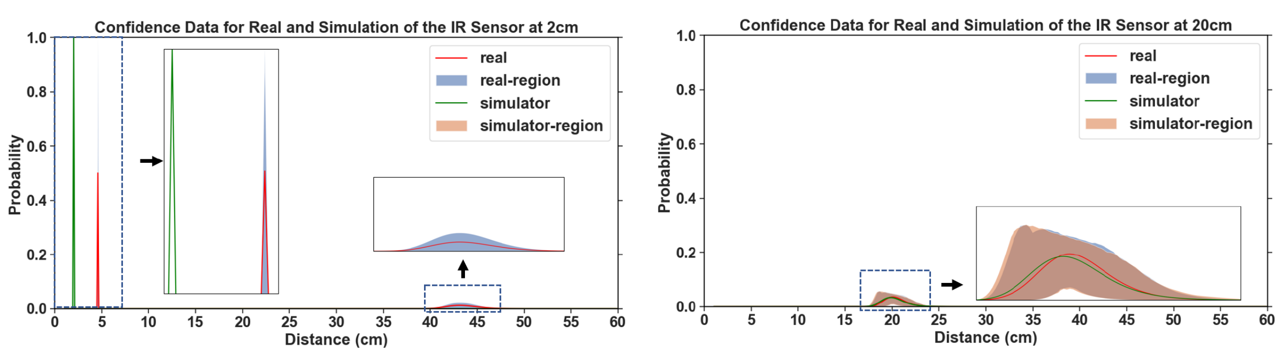
Ultrasonic: Figures 5 (a) and 5 (b) show the average plot of 200 trials in simulation (green) and real-world (orange). The real (blue-shaded) and simulated (orange-shaded) regions represent the 90% confidence interval of our data. The real and simulated distributions are similar in both cases, which verifies that our model accurately describes the sensor. We note, however, that the model is more accurate at 20cm than at 2cm.

Infrared: Figures 6 (a) and 6 (b) show similar results from the ultrasonic sensor data for the 200 trials, shown in the real (orange) and simulation (green). The real (blue-shaded) and simulated (orange-shaded) colored regions represent the 90% confidence interval. Once again, the distributions are similar, which verifies the accuracy of the sensor model.



(a) (b)

Fig. 5: (a) Confidence data for real (orange) and simulation (blue) of the ultrasonic sensor at 2cm. (b) Confidence data for real (orange) and simulation (blue) of the ultrasonic sensor at 20cm.



(a) (b)

Fig. 6: (a) Confidence data for real (orange) and simulation (blue) of the infrared sensor at 2cm. (b) Confidence data for real (orange) and simulation (blue) of the infrared sensor at 20cm.

5.2. Comparing Robotic Behavior in Simulation and Reality

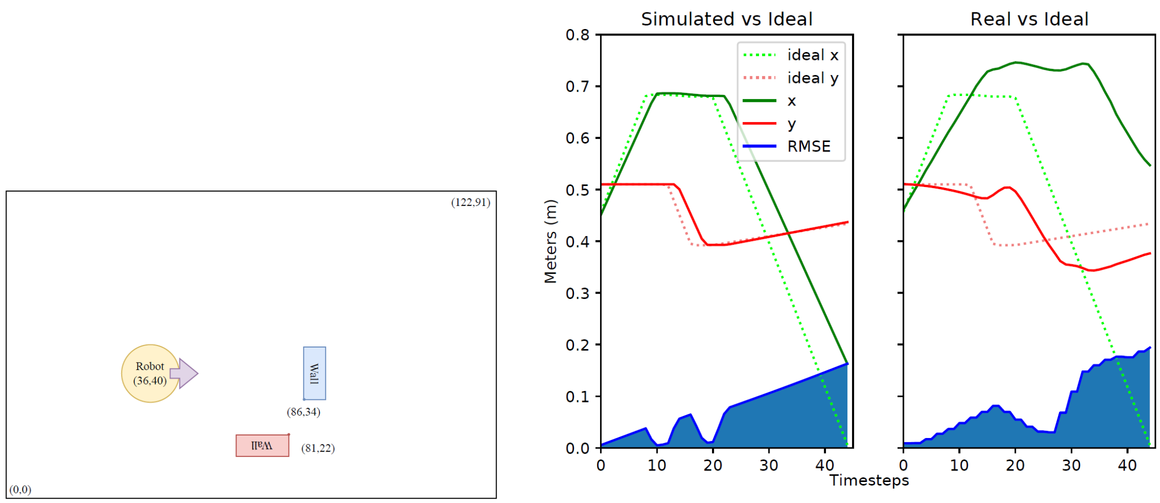
Our next experiment compares the real and simulated robot systems in an object avoidance task. This diagram assumes one measurement; in some cases, the sensor reading is dependent on the Bayes filter, which we require to make four readings to develop a belief of the sensor reading. In both the real and simulation, we record the x, and y coordinates of the robot as it progresses through the given layout shown in Figure 7 (a). Based on the intended control, the ideal robot will move forward until it is at coordinates (76, 51), it will turn 90 degrees right, it will continue until (76, 42), it will turn 90 degrees right, and will continue to the left.

The experiment here is to compare what really happens compared to the ideal robot for the robot with ultrasonic and infrared sensors under real and simulation.

Figure 7 (b) shows two plots of the x (green) and y (red) values of the robot positions averaged over 25 runs in simulation on the left and real on the right. In both cases, the ideal x and y points are plotted in dotted lime green and light rose lines, respectively. Finally, a blue line is shown underneath which represents the root mean squared error (RMSE) calculated with respect to adding the x and y errors as in [35].

The key idea we note from these results is the RMSE continues to rise as the simulation continues, which is to be expected. One question to ask is how can this error be used to quantify simulation confidence, and we made attempts to use Kullback–Leibler divergence [36] to compare the two multivariate distributions, but found no interesting results, so far.

Our experimental setup does have some shortcomings. First, the U-turn nature of the experiment results in the dips in the RMSE plots just due to the nature of where the robot is in comparison to where it should be. Second, the experiment relies on very few sensor readings such that the probability of an incorrect reading that causes the control to leave the intended path is so low, and does not occur in the small set of experiments we run.



(a) (b)

Fig. 7: (a) The experimental setup for one robot. (b) The x and y coordinates for the ideal, simulated, and real system with root mean squared error plotted in blue.

6. Conclusion

In this work, we show our first steps towards creating a meaningful confidence measure of the simulation "reality gap". Our approach differs from many other attempts in this space in that we shift to a simple robotic system and make an attempt at a first-principle characterization of robot distance sensing and movement. Our results show that our sensor characterization is reasonably close for both a cheap ultrasonic and IR sensor between real and simulation based on Bayesian filters confidence readings. Second, we show that our characterization approach results in similar behavior for simulation and real execution of the robot for a simple control algorithm and experiment. These are promising steps towards creating a confidence measure of the "reality gap".

Our future work in this space will focus on further detailing a method to make a confidence measure of the simulation of 2D mobile robots. In particular, we believe that by creating a more complex experiment where we have the robot run continuously noting state changes in control and then using a count of these state changes as a metric to give them a confidence value meaning. The interesting thing about state changes on the robot is they correspond to two varying aspects of simulation and reality fidelity. Simple states that result in movement and sensing leads to diverging simulation points with respect to the reality gap, but other states such as the Bayesian filter (or other more complex control such as a PID line control) lead to converging simulation points. Because the robots are agents that may have intelligence the divergent chaos of simulation has a reverse entropy aspect, which will be interesting to explore.

References

[1] E. N. Lorenz, "Deterministic nonperiodic flow," Journal of atmospheric sciences, vol. 20, no. 2, pp. 130–141, 1963.

[2] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in 2011 IEEE international conference on robotics and automation. IEEE, 2011, pp. 3400–3407.

[3] J. Craighead, R. Murphy, J. Burke, and B. Goldiez, "A survey of commercial & open source unmanned vehicle simulators," in Proceedings 2007 IEEE International Conference on Robotics and Automation. IEEE, 2007, pp. 852–857.

[4] A. Staranowicz and G. L. Mariottini, "A survey and comparison of commercial and open-source robotic simulator software," in Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments, 2011, pp. 1–8.

[5] A. Harris and J. M. Conrad, "Survey of popular robotics simulators, frameworks, and toolkits," in 2011 Proceedings of IEEE Southeastcon. IEEE, 2011, pp. 243–249.

[6] N. R. Ramli, S. Razali, and M. Osman, "An overview of simulation software for non-experts to perform multi-robot experiments," in 2015 International Symposium on Agents, Multi-Agent Systems and Robotics (ISAMSR). IEEE, 2015, pp. 77–82.

[7] L. Pitonakova, M. Giuliani, A. Pipe, and A. Winfield, "Feature and performance comparison of the v-rep, gazebo and argos robot simulators," in Annual Conference Towards Autonomous Robotic Systems. Springer, 2018, pp. 357–368.

[8] J. Collins, S. Chand, A. Vanderkop, and D. Howard, "A review of physics simulators for robotic applications," IEEE Access, vol. 9, pp. 51 416–51 431, 2021.

[9] O. Michel, "Cyberbotics ltd. webots™: professional mobile robot simulation," International Journal of Advanced Robotic Systems, vol. 1, no. 1, p. 5, 2004.

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

[11] B. Gerkey, R. T. Vaughan, A. Howard et al., "The player/stage project: Tools for multi-robot and distributed sensor systems," in Proceedings of the 11th international conference on advanced robotics, vol. 1, 2003, pp. 317–323.

[12] J. Collins, D. Howard, and J. Leitner, "Quantifying the reality gap in robotic manipulation tasks," in 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 6706–6712.

[13] A. S. Fiorillo, "Design and characterization of a pvdf ultrasonic range sensor," IEEE transactions on ultrasonics, ferroelectrics, and frequency control, vol. 39, no. 6, pp. 688–692, 1992.

[14] J. Ure˜na, M. Mazo, J. J. Garc´ıa, E. Bueno, A. Hernandez, and D. Hernanz, "Low-cost improvement of an ultrasonic sensor and its characterization for map-building," IFAC Proceedings Volumes, vol. 31, no. 2, pp. 293–298, 1998.

[15] M. Alwan, M. B. Wagner, G. Wasson, and P. Sheth, "Characterization of infrared range-finder pbs-03jn for 2-d mapping," in Proceedings of the 2005 IEEE international conference on robotics and automation. IEEE, 2005, pp. 3936–3941.

[16] L. Kneip, F. Tˆache, G. Caprari, and R. Siegwart, "Characterization of the compact hokuyo urg-04lx 2d laser range scanner," in 2009 IEEE International Conference on Robotics and Automation. IEEE, 2009, pp. 1447–1454.

[17] J. Poppinga, A. Birk, and K. Pathak, "A characterization of 3d sensors for response robots," in Robot Soccer World Cup. Springer, 2009, pp. 264–275.

[18] Z. Pezzementi, E. Jantho, L. Estrade, and G. D. Hager, "Characterization and simulation of tactile sensors," in 2010 IEEE Haptics Symposium. IEEE, 2010, pp. 199–205.

[19] A. Visser, N. Dijkshoorn, M. Van der Veen, and R. Jurriaans, "Closing the gap between simulation and reality in the sensor and motion models of an autonomous ar. drone," in International Micro Air Vehicle conference and competitions 2011 (IMAV 2011), The Netherlands, September 12-15,
2011. Delft University of Technology and Thales, 2011.

[20] J. R. V. Rivero, I. Tahiraj, O. Schubert, C. Glassl, B. Buschardt, M. Berk, and J. Chen, “Characterization and simulation of the effect of road dirt on the performance of a laser scanner,” in 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2017, pp. 1–6.
[21] M. Gschwandtner, R. Kwitt, A. Uhl, and W. Pree, “Blensor: Blender sensor simulation toolbox,” in International Symposium on Visual Computing. Springer, 2011, pp. 199–208.
[22] J. T. Collins, “Simulation to reality and back: A robot’s guide to crossing the reality gap,” Ph.D. dissertation, Queensland University of Technology, 2022.
[23] F. Murator, “Randomizing physics simulations for robot learning,” Ph.D. dissertation, Technische Universitat Darmstadt, 2021.
[24] R. A. Brooks, “Artificial life and real robots,” in Proceedings of the First European Conference on artificial life, 1992, pp. 3–10.
[25] S. Koos, J.-B. Mouret, and S. Doncieux, “Crossing the reality gap in evolutionary robotics by promoting transferable controllers,” in Proceedings of the 12th annual conference on Genetic and evolutionary computation, 2010, pp. 119–126.
[26] N. Jakobi, P. Husbands, and I. Harvey, “Noise and the reality gap: The use of simulation in evolutionary robotics,” in European Conference on Artificial Life. Springer, 1995, pp. 704–720.
[27] F. Zhang, J. Leitner, M. Milford, B. Upcroft, and P. Corke, “Towards vision-based deep reinforcement learning for robotic motion control,” in Proceedings of the Australasian Conference on Robotics and Automation 2015. Australian Robotics and Automation Association, 2015, pp. 1–8.
[28] J.-B. Mouret and K. Chatzilygeroudis, “20 years of reality gap: a few thoughts about simulators in evolutionary robotics,” in Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2017, pp. 1121–1124.
[29] M. Eaton, “Bridging the reality gap a dual simulator approach to the evolution of whole-body motion for the nao humanoid robot.” in IJCCI (ECTA), 2016, pp. 186–192.
[30] B. Mehta, A. Handa, D. Fox, and F. Ramos, “A user’s guide to calibrating robotics simulators,” arXiv preprint arXiv:2011.08985, 2020.
[31] A. Afzal, D. S. Katz, C. L. Goues, and C. S. Timperley, “A study on the challenges of using robotics simulators for testing,” arXiv preprint arXiv:2004.07368, 2020.
[32] M. Rubenstein, C. Ahler, N. Hoff, A. Cabrera, and R. Nagpal, “Kilobot: A low cost robot with scalable operations designed for collective behaviors,” Robotics and Autonomous Systems, vol. 62, no. 7, pp. 966–975, 2014.
[33] S. Wilson, P. Glotfelter, L. Wang, S. Mayya, G. Notomista, M. Mote, and M. Egerstedt, “The robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems,” IEEE Control Systems Magazine, vol. 40, no. 1, pp. 26–44, 2020.
[34] S. Thrun, W. Burgard, and D. Fox, Probabilistic robotics. Cambridge, Mass.: MIT Press, 2005.
[35] L. Lee, M. Jones, G. S. Ridenour, S. J. Bennett, A. C. Majors, B. L. Melito, and M. J. Wilson, “Comparison of accuracy and precision of gps-enabled mobile devices,” in 2016 IEEE International Conference on Computer and Information Technology (CIT). IEEE, 2016, pp. 73–82.
[36] S. Kullback and R. A. Leibler, “On information and sufficiency,” The annals of mathematical statistics, vol. 22, no. 1, pp. 79–86, 1951.