# Example System Simulation

Peter Kohler

March 27, 2023

For my final project I am using the default system because I am interested in robotics and also a robotics major here at Mines. To simulate this two wheeled robot, I am using a common and popular simulation tool in the robotics community called Gazebo. Gazebo is an open source 3D simulator with a fully integrated physics engine. I am using an open source simulation package of a robot called the turtlebot3 which is pictured below.

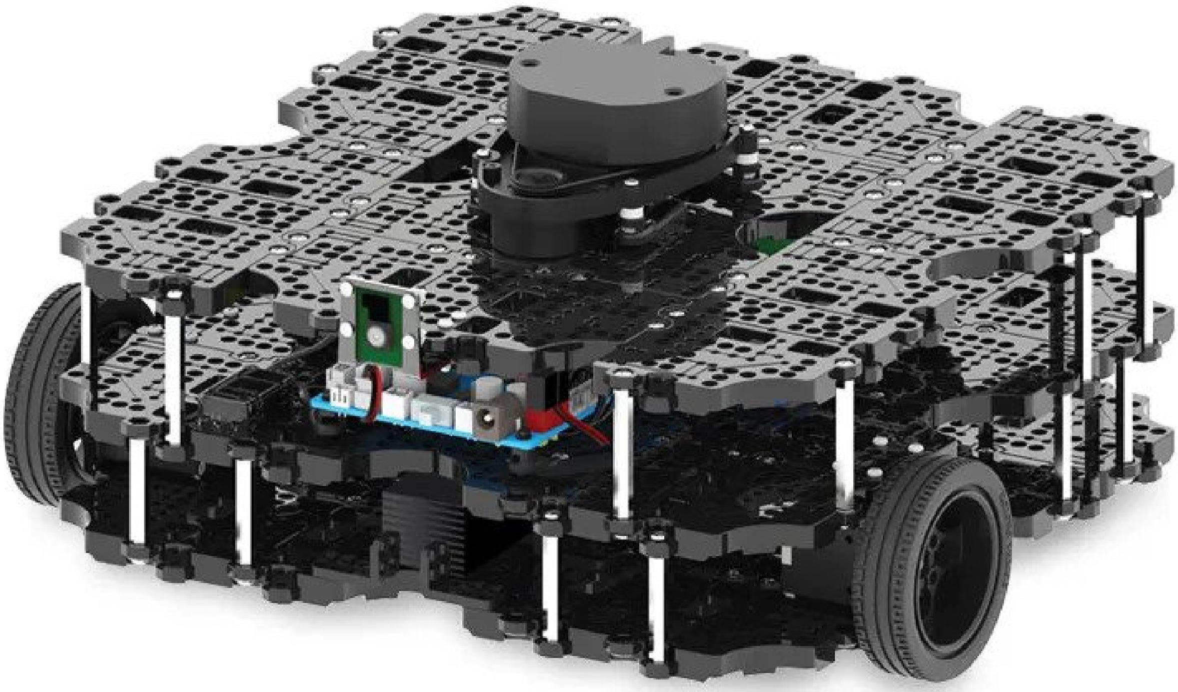

Figure 1: Turtlebot3 WafflePi

I have all of the open source packages installed and up and running. Figure 2 and 3 shows screenshots of the simulator running a simple world with the turtlebot robot in action.

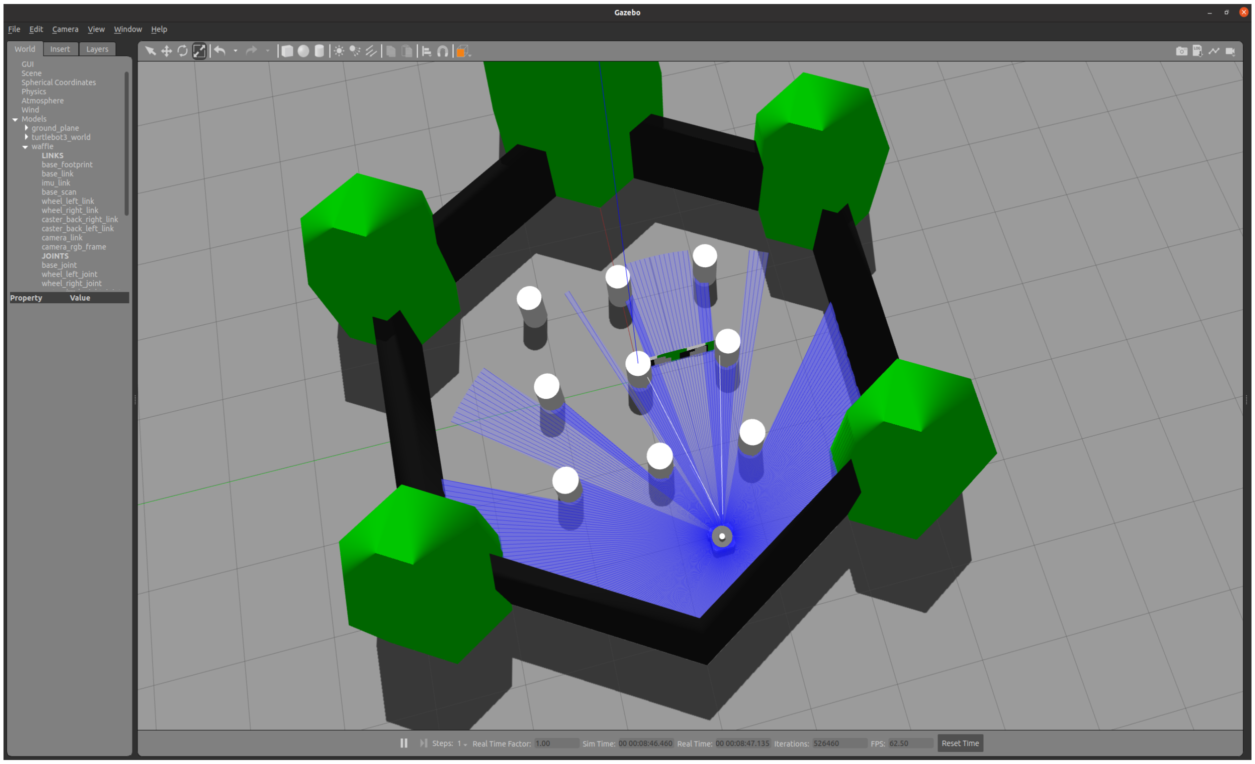


Figure 2: Overhead view of Gazebo and the Turtlebot

The blue ”rays” coming out of the turtlebot are visual indicators of the lidar sensor on the turtlebot.

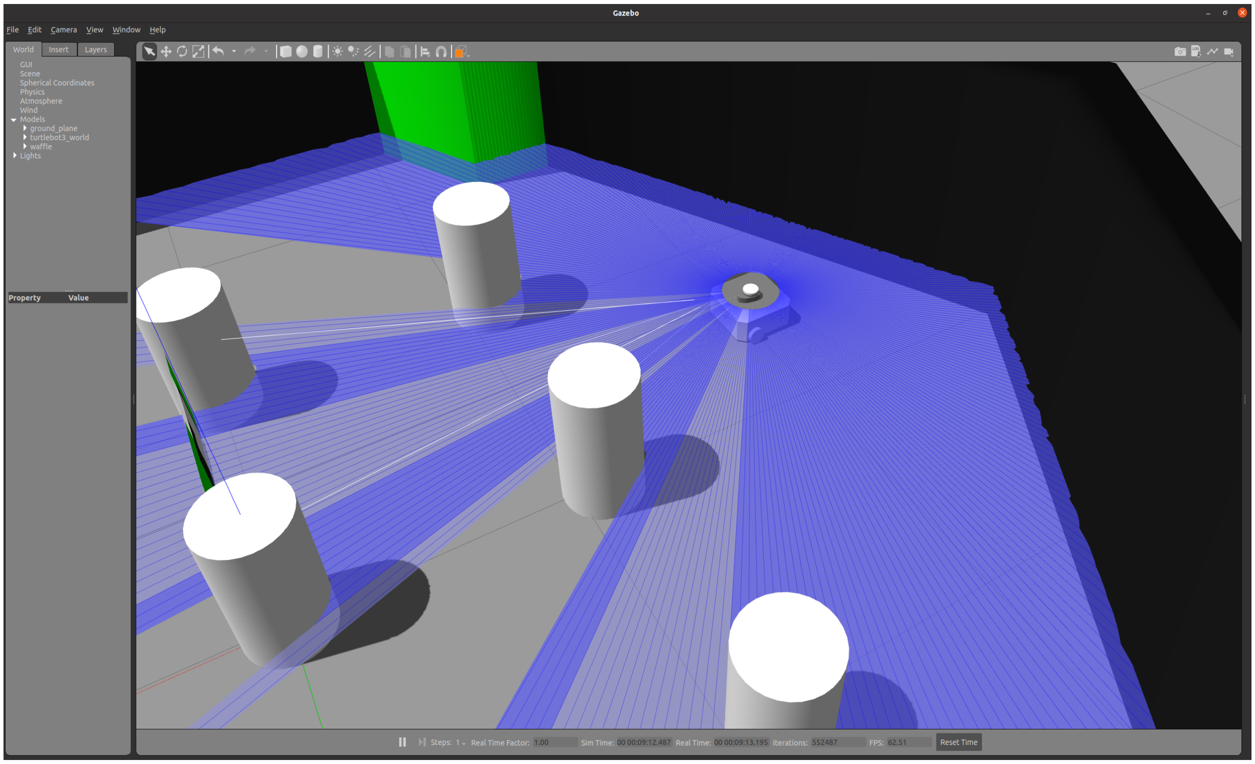


Figure 3: Closeup Screenshot of Turtlebot in Gazebo

For the purposes of the final project I have access to a large amount of data coming from the simulation. Relevant to the final project and state estimation I have access to the odometry of the robot which includes position, pose, linear and angular velocity of the turtlebot. Also available is the state of the joints of the turtlebot robot. This includes the position and velocity of both wheels which is directly relevant to the given control law that will be used. I plan on using this data as a "ground truth" since this data that is being published by the simulator is "perfect" in that there is no noise. For the purposes of state estimation and the work to be done for the project I wrote my own plugin that integrates with the Gazebo simulator that takes the noiseless joint state information (particularly velocity) and added gaussian noise with a mean of 0 and a variance of 0.01. Conveniently, MATLAB has a toolbox that integrates with the meta-operating system called ROS2 that powers the Gazebo simulator which lets me pull data directly being published by the simulator. Figures 4 and 5 are plots of data I pulled into MATLAB that was published from Gazebo while operating the turtlebot and driving in a simple S curve.

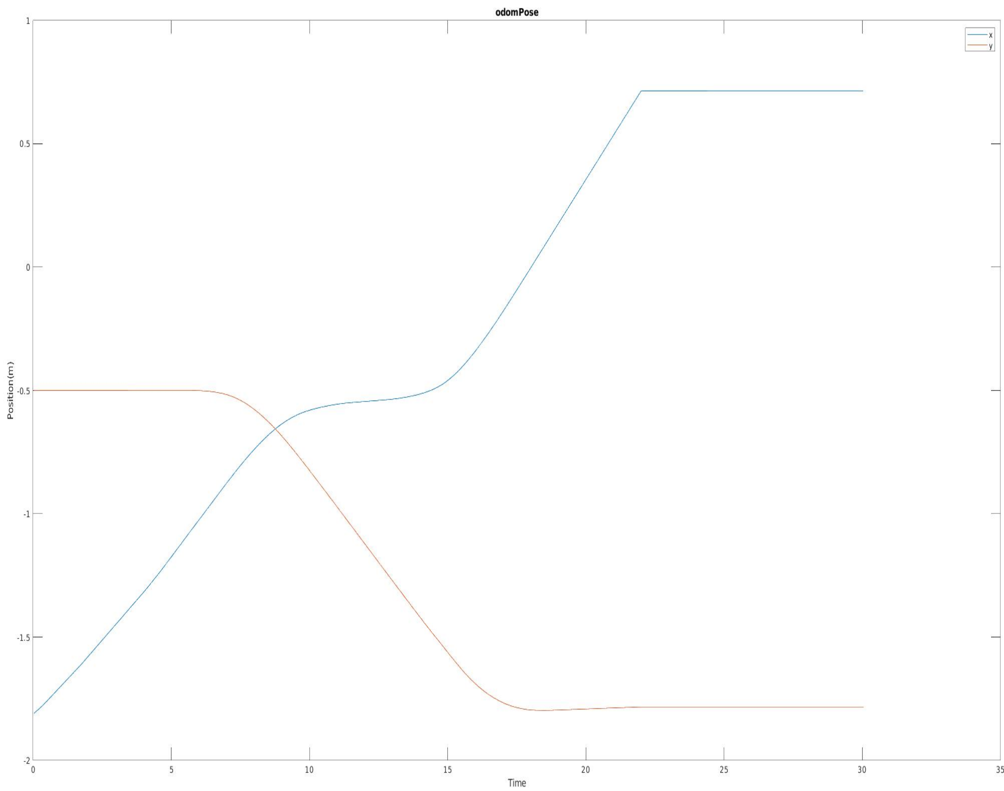


Figure 4: Ground Truth Odometry Plot in MATLAB

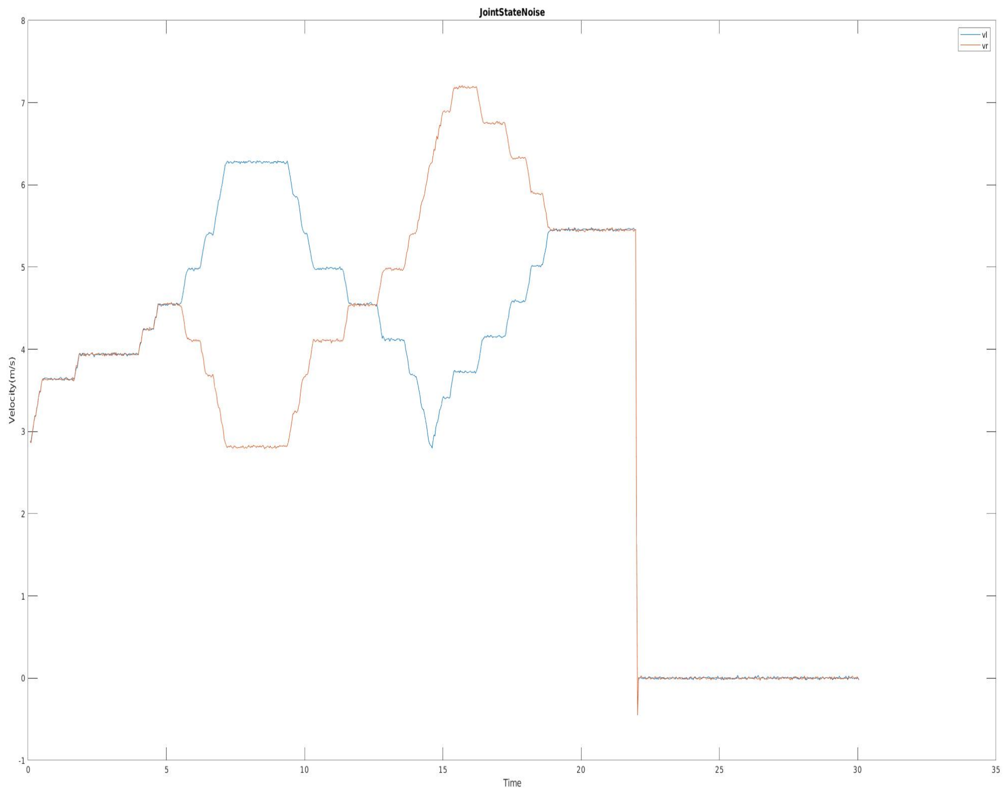


Figure 5: Wheel Velocities with Gaussian Noise Added Plot in MATLAB

The plots in MATLAB are a little hard to see, but if you zoom in on Figure 5 you can see the jagged edges that constitute the added Gaussian noise.

To summarize, I plan on using the Gazebo simulator to generate noisy wheel velocity data of a fully modeled and simulated two-wheeled robot. This will serve as my **input** for the project. I will write code that will take in this noisy wheel velocity data in real time and perform state estimation to determine the estimated wheel velocities and will then attempt to perform odometry calculations to compare the estimated position of the robot with the ground truth value from the Gazebo simulator. These odometry calculations will serve as my **output**. If I have extra time, I'd like to try and get some kind of real time plot of the robot's estimated position vs. the ground truth data but I'm not sure if I'll have time to do that.