CHAPTER I

Navigation Results

To test the efficacy of the GODZILA algorithm, the Nao humanoid platform was instrumented with a Hokuyo LIDAR and set in three different arenas to navigate to a goal. The Nao was used as the mobile base for the experiment while the Hokuyo LIDAR provided range and bearing information about arena obstacles. The NaoQI API provides functions for tracking a red object using its onboard cameras to estimate the range and bearing to the object. A red cube was designated as the goal location that the robot could track. A secondary camera was mounted to the ceiling of the room in which the arena was set up such that the entire arena and progress of the robot could be recorded. This video was only used to record the results of the experiment and at no time did the robot have access to this data during the experiment. The relative pose estimates from the Nao about the goal location were also recorded for analysis.

Details about the Nao and LIDAR can be found in Chapter **??** and details about the arena setup, data collection, and algorithm parameters can be found in Section 1.1. Data collected by the robot about the goal pose is presented in Section 1.2 and robot pose data collected by the global camera is presented in Section 1.3.

1.1 Experimental Setup

Using the NaoQI API discussed in Chapter **??**, the GODZILA algorithm was implemented on the Nao in C++. Different arenas were constructed and the robot was set at some start location and navigated to some radius from the goal location before stopping. Concurrently, the parameters of the algorithm were tuned such that the robot performed well in any of the tested arenas. This section discusses the platform hardware (Section 1.1.1), goal pose estimation (Section 1.1.2), and the tuned GODZILA parameters (Section 1.1.3). Finally, it overviews the experimental arenas (Section 1.1.4) and the global camera setup (Section 1.1.5) used to record each test.

1.1.1 Mobile Platform and LIDAR

The Nao humanoid platform was used as the mobile base for the experiment. The NaoQI API provides a bipedal gaiting function that allows the robot to walk over flat terrain. As the floor of the arena was approximately flat, this gait was applicable to this experiment and was not modified before use. To the robot's chest was attached a Hokuyo LIDAR via a 3D printed mount. The LIDAR had a measurement range between 0.02 and 5.6 meters with an angular resolution of 0.352° and a viewing angle of ±120°. Data from this sensor was used by the robot to detect obstacles in the environment. GODZILA used this data to generate the repulsive force vectors discussed in Chapter **??**. These repulsive forces allow the robot to avoid colliding with obstacles in the environment. An in depth discussion of the Nao, LIDAR, and LIDAR mount can be found in Chapter **??**. A shortcoming of this approach was that while the LIDAR and mount were physically compact and well within the load carrying capacity of the Nao platform, the dynamics generated by moving this additional mass during gaiting were not accounted for by the in-built walking algorithm as there was no method available to modify the dynamic model. As a result, the walking gait became marginally stable and at times, unstable. The marginal stability of the platform can be seen in the results presented in Section 1.3. In future work, these additional dynamics will need to be accounted for in order to increase the robustness of the platform, should it be used again for this or a similar experiment.

1.1.2 Goal Pose Estimation

The head of the Nao is equipped with two color cameras. The optical axis of the first camera is nearly collinear with the x-axis of the head, allowing the robot to see in front of it. The optical axis of the second camera is pitched down by approximately 40°, allowing the robot to see objects near it's feet. The NaoQI API provides functions to track a "red ball" using the images from the first camera. These functions actuate the head in an effort to keep the "red ball" in frame at all times. The functions also return the position of the "red ball" relative to the torso of the robot. This can be done because the API assumes the diameter of the ball to be $0.06m$, allowing the apparent diameter of the ball to estimate the actual diameter. With this as a tool, a red object was constructed to indicate to the robot the location of the navigation goal. The object was tracked during navigation and its estimated position was used by GODZILA to generate the attractive force vector discussed in Chapter **??**. The red object was built as a cube with sides $0.127m$ long. The cube was mounted on a wooden dowel that was nearly as tall as the Nao which was in turn affixed to a heavy wooden base. The cube was mounted on the dowel to allow the robot to see it more easily as this allows the head to minimize the amount of pitching that must be done to keep the cube in view. The red object was a cube because during testing the "red ball" tracking algorithm did not seem to be affected by the change in geometry and the cube shape was easier to construct. The cube width was twice as wide as the expected diameter to allow the target to be seen by the robot from longer distances. This longer range allowed for the construction of a larger arena as well as more robust tracking at shorter ranges. However, the larger object results in an incorrect range estimate of the tracked object. While this estimation error is rectifiable through appropriate calibration, during these experiments the corrections were not performed. This would mean that the algorithm parameters would be different in a calibrated system and the presented data shows the robot stops roughly twice as far from the goal than as instructed.

1.1.3 GODZILA Parameters

The GODZILA algorithm uses a number of tunable parameters to generate vehicle navigation commands. Chapter ?? discusses the theoretical basis for these parameters. Each needs to be experimentally tuned in order to achieve optimal performance. They can be roughly divided into three categories which are discussed in this section. The first set is concerned with limits and thresholds governing navigation performance. The second influences the behavior of the robot while turning, while the third influences the forward motion of the robot. While testing the GODZILA implementation, these parameters were tuned according to the observed behavior during the testing runs. Table 1 shows the parameters used by the algorithm which produced the results presented in this chapter.

Navigation Thresholds

Seven parameters limit the navigation performance of the robot to be within certain bounds. Generally, these parameters exist to ensure the safety of the robot and environment but their inclusion affects the tuning of the other parameters as they introduce system non-linearities.

Category	**Parameter Name**	**Value**
Thresholds	Goal Stopping Radius	0.3 m
	Minimum Linear Velocity	-0.4 $\frac{m}{s}$
	Maximum Linear Velocity	0.4 $\frac{m}{s}$
	Minimum Angular Velocity	-0.2 $\frac{rad}{s}$
	Minimum Angular Velocity	0.2 $\frac{rad}{s}$
	Clearance Threshold	0.3 m
	Obstacle Threshold	3.0 m
Turning	Goal Attraction	100
	Obstacle Repulsion	20
	Obstacle Attraction	0
	Obstacle-Goal Bearing Ratio	1
	Vehicle Inertia Gain	15
Forward Motion	Velocity Gain	5
	Obstacle Repulsion	5
	Angular Rate Braking	3

Table 1: Table of GODZILA parameters and the values used during experimentation. Each parameter is divided into categories controlling thresholds, turning, and forward motion. While the thresholds have units, the turning and forward motion parameters consist of gains and ratios, which have no unit.

Goal Stopping Radius This is the maximum distance the robot can be from the goal point before GODZILA considers the robot to have reached the goal position. Practically, the robot will never estimate its range from the goal to be exactly zero, so to prevent the robot from oscillating around the goal pose, this parameter gives a stopping criteria for navigation.

Minimum Linear Velocity This is the minimum linear velocity of the vehicle in $\frac{m}{s}$, which is allowed to be negative. Minimums were decided we defined this way to allow simple configuration of non-symmetric velocity limits.

Maximum Linear Velocity This is the maximum linear velocity of the vehicle in $\frac{m}{s}$.

Minimum Angular Velocity This is the minimum angular velocity of the vehicle in $\frac{rad}{s}$, which is allowed to be negative.

Minimum Angular Velocity This is the maximum angular velocity of the vehicle in $\frac{rad}{s}$.

Clearance Threshold This is the minimum acceptable distance between the vehicle and an obstacle. It is not guaranteed that this threshold will never be violated, rather a very high cost is associated with traveling this close to obstacles. This distance is defined as the center of the robot to the center of an obstacle in meters. This center to center distance is because all objects are modeled as points.

Obstacle Threshold Obstacles that are farther from the robot than this distance are treated as attractive rather than repulsive. This allows the robot to approach obstacles in order to allow for some degree of environmental exploration and prevents the robot from simply settling in the middle of the largest expanse. This quantity is measured in meters.

Turning Parameters

Five parameters are used to shape the function used to generate turning commands for the robot. They affect the attraction and repulsion to the goal and obstacles as well as dampening oscillations.

Goal Attraction This gain influences the strength of the goal attraction force. Larger values increase attraction strength.

Obstacle Repulsion This gain influences the strength of the obstacle repulsion force for obstacles closer than the Obstacle Threshold distance. Larger values increase repulsion strength.

Obstacle Attraction This gain influences the strength of the obstacle attraction force for obstacles farther than the Obstacle Threshold distance. Larger values increase attraction strength. This predisposes the robot to point in the direction of far obstacles.

Obstacle-Goal Bearing Ratio This gain tunes the trade off between avoiding obstacles which are in the robot's current direction of travel versus avoiding objects which are in the direction of the goal. This parameters ranges from 1 to 0. Values closer to 1 amplify the avoidance of obstacles in the direction of travel. Values closer to 0 amplify the avoidance of obstacles in the direction of the goal.

Vehicle Inertia Gain This gain influences the strength of the vehicle's resistance to turning. Larger values mean more resistance to turning. This can be seen as increasing the inertia of the robot and will help dampen high frequency oscillations associated with intermittent obstacle detection and local minima conditions. The algorithm is constructed such that forward motion is reduced in the presence of large angular velocities so high frequency changes in desired heading angle slow the robot down.

Forward Motion Parameters

Three parameters are used to shape the function used to generate forward motion commands. The term forward motion is used as this function only commands the robot to move in the direction (or in the opposite direction) of its current heading. A different function would have to be formulated to take advantage of the planar holonomy of the Nao humanoid, which would likely have different parameters.

Velocity Gain This gain influences the "aggressiveness" of the forward velocity function. Larger values effectively increase the slope of this function, meaning the change in velocity will be greater for a given change in the other input parameters.

Obstacle Repulsion This gain influences the strength of repulsion force generated by the closest obstacle. Larger values increase repulsion strength. The strategy here is that the robot should slow down if there are obstacles close to it. Only the closest obstacle is used to modify forward velocity.

Angular Rate Braking This gain influences the amount by which high turning rates reduce forward velocity. Larger values reduce forward velocity to a greater extent. In general, when the robot is trying to turn quickly, it is likely that heading forward in that direction is no longer desired.

1.1.4 Experimental Arena

Three arenas were constructed to test the efficacy of the GODZILA algorithm. In all three, the goal location is observable by the robot at all times as the robot had no other method to know goal location, nor is there any provision in the GODZILA algorithm for doing searches or explicit explorations. The first arena was simply an open area in which the robot could walk towards the goal. This trivial environment acts as a control in which the robot is always expected to reach the goal. The second arena contains an obstacle which divides the arena into two parts such that the only means to reach the goal is through a narrow opening. The objective is to demonstrate the narrowest opening that can be traversed using this method. Finally, an arena was constructed which had a large obstacle that would always have a large local influence on the navigation. The robot must overcome the large repulsive forces and walk through a hallway to the goal. In common to the three arenas was a perimeter wall which acted to contain the environment and prevent the presence of unplanned obstacles from influencing test results. The arena perimeter was approximately $11.75m$ long by $5.5m$ wide. As an aside, none of the perimeter walls are depicted in any of the figures seen in this chapter. This is simply an aesthetic choice and are present in each experiment.

Arena Construction

The arenas were constructed using large cuboidal objects and flat walls. Aside from the ease of construction, this is motivated by the choice of sensor used on the robot for obstacle avoidance. The cuboids and walls have the same cross section when viewed from above. As the LIDAR used only detects objects in a plane, obstacles which have overhangs or voids at different levels are effectively invisible to the robot and pose a

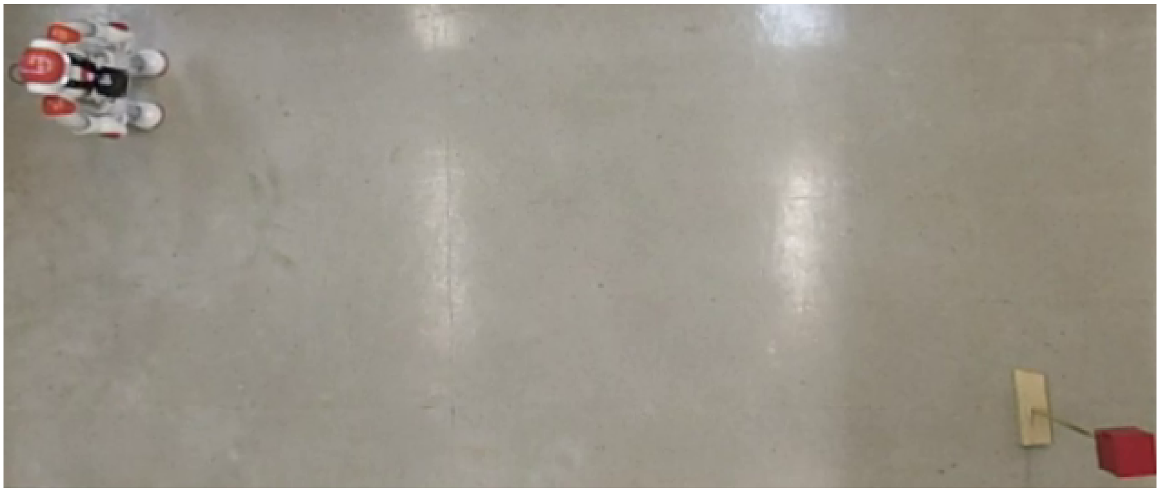

Figure 1: This figure shows the open arena. The Nao can bee seen on the left at the starting location while the goal cube can be seen to the right.

collision risk. This is not a shortcoming of the algorithm but rather the sensor, therefore, using obstacles that avoid this risk is appropriate.

Open Arena

The open arena contained no obstacles and only the arena perimeter walls and the goal object were present to affect the navigable path. Figure 1 shows a picture of the arena as seen by the overhead camera.

Narrow Opening Arena

This arena contains an obstacle that divides the area into two equal parts, with the robot and goal being in opposing partitions. The only path between the two rooms is a narrow opening that is $73cm$ wide. This width is approximately 2.6 times the shoulder-to-shoulder width of the robot. Figure 2 shows the arena with the narrow opening. The objective was to demonstrate the narrowest opening that the robot could travel through with this method. It was found than openings narrower than this caused the robot to approach other obstacles too closely when the parameters were tuned to allow the robot to traverse this aperture. At times, with the parameters tuned to allow for tighter openings, the robot would collide with corners it might otherwise have avoided. To allow for the navigation of narrower openings that are still physically traversable, one strategy is to have a global path planner that could place intermediate goals to “pull” the robot through. As GODZILA is designed to be a lightweight local algorithm, such

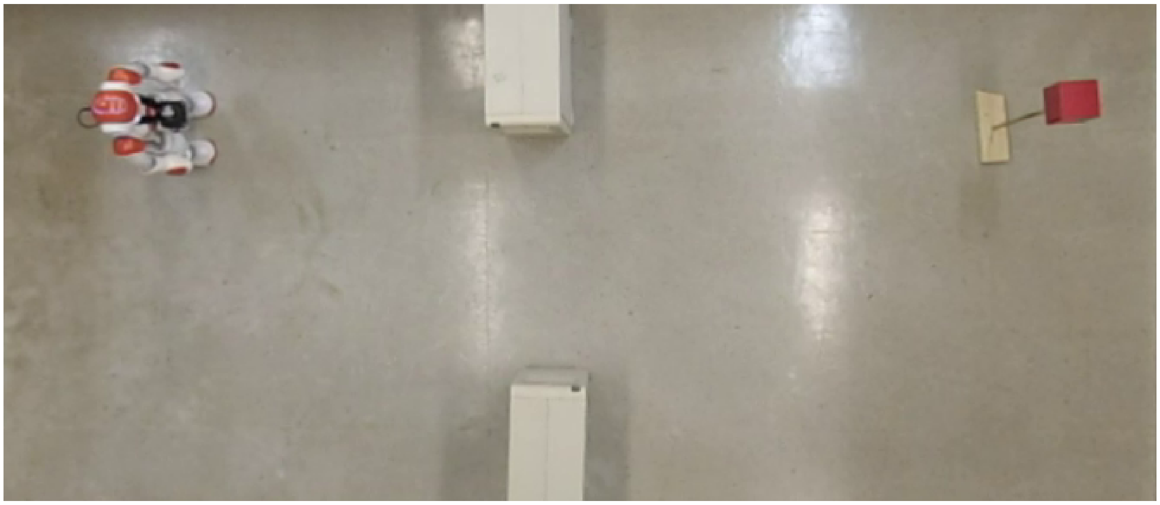

Figure 2: This figure shows the arena with the narrow opening between partitions. The opening is approximately 2.6 times the width of the robot and represents the practical limit to the traversable apertures using this approach. The Nao can bee seen on the left at the starting location while the goal cube can be seen to the right.

provisions are inappropriate for addition to the approach as they tend to require maps. The different algorithm classes and approaches are reviewed in Chapter **??**.

Large Obstacle Arena

This arena contained the largest single obstacle possible while still allowing the arena to be traversable. It consists of a corridor constructed to be the minimum width a hallway can be with this approach. The corridor also has two 90° turns. The objective was to have the robot navigate in an area where the repulsion of obstacles was a strong influence during the entire path. It shows that while the robot is constantly being repelled, it can use the goal and far obstacle attraction to find a path. Figure 3 shows the arena.

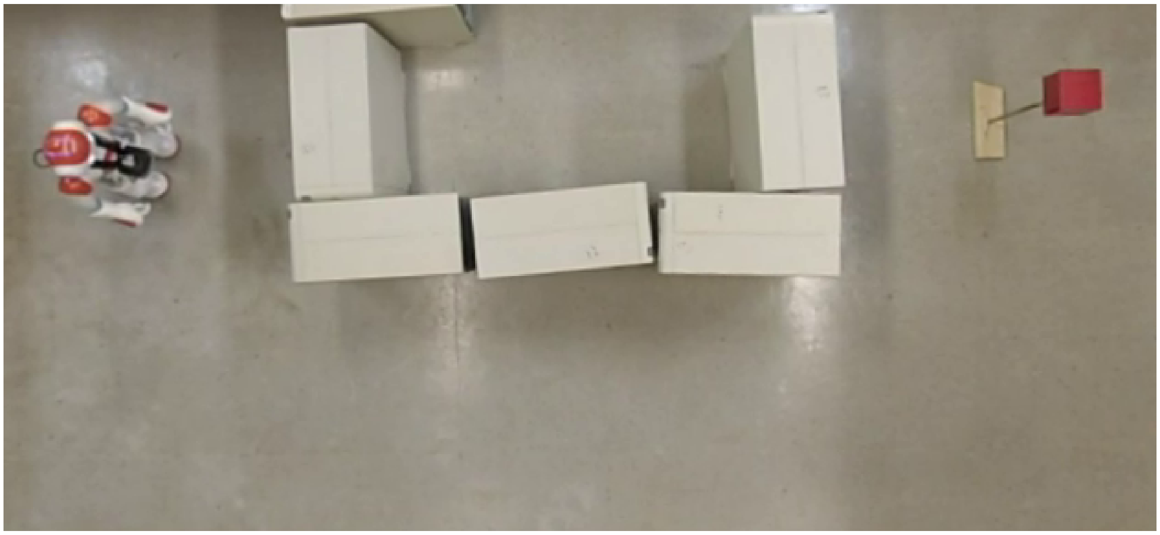

Figure 3: This figure shows the arena containing a large obstacle. The result is a hallway that is the narrowest constructible and still have this approach be viable. This results in the robot being under strong repulsion for the length of the path. The Nao can bee seen on the left at the starting location while the goal cube can be seen to the right.

1.1.5 Robot Pose Estimation

In order to track the robot's path for later analysis, a GoPro camera was mounted on the ceiling above the arena that could record the starting and goal location in the same frame. This meant that the camera could remain fixed throughout the experiment, simplifying later analysis. As presented in Section 1.3, this data was processed using the OpenCV library in Python to track the Nao through the video and display the path of the robot for each experiment.

1.2 Goal Pose Tracking

During navigation, the robot used its camera to track the goal location relative to the robot as discussed in Section 1.1.2. Figure 4 shows the perceived relative goal range and bearing during the open arena experiment while Figures 5 and 6 show the results from the narrow opening and large obstacle experiments. All three datasets show the goal range data following an approximate asymptotic decrease over time. This is expected as in each of the arenas the robot gets closer to the goal in each step. While these datasets demonstrate this occurrence, in general there are some arenas where the range could be constant for a time or even increase slightly. In the presence of a long obstacle that was radially constant from the goal, the range would constant. Alternatively, if there was a short wall that was in the path of the goal, the range could increase as the robot avoids it.

The goal bearing angle in the open arena converges to a small value during the majority of the experiment while in the narrow opening and large obstacle experiments, the bearing can be seen as oscillating around some slowly decreasing function. In all of the experiments, the robot is initially oriented facing the goal to ensure strong goal tracking. In the open arena, this means the robot is already nearly optimally oriented to approach the goal, and only small corrections to heading are necessary to reach the goal. In the narrow opening and large obstacle arenas, the robot must first turn away from the goal to avoid obstacles before the robot can start to face the goal again. Figures 8 and 9 give additional insight into this process. The oscillations are likely due to a combination of corrections to the navigation process and to large head movement due to gaiting oscillations. Section 1.3 gives more insights to these gaiting oscillations, due to the added LIDAR mass discussed in Section 1.1.1.

In Figures 4, 5, and 6, it is also shown that the final perceived goal range is close to the $0.3m$ prescribed by the Goal Stopping Radius from Table 1. Section 1.3 shows that the robot stopped at a radius closer to $0.8m$, a mismatch generated by the lack of calibration mentioned in Section 1.1.2.

Additionally, a future application of this data is to localize the robot during the experiment. This localization data could be used to inform other path planning algorithms or trap detection schemes to improve navigation in more complex environments.

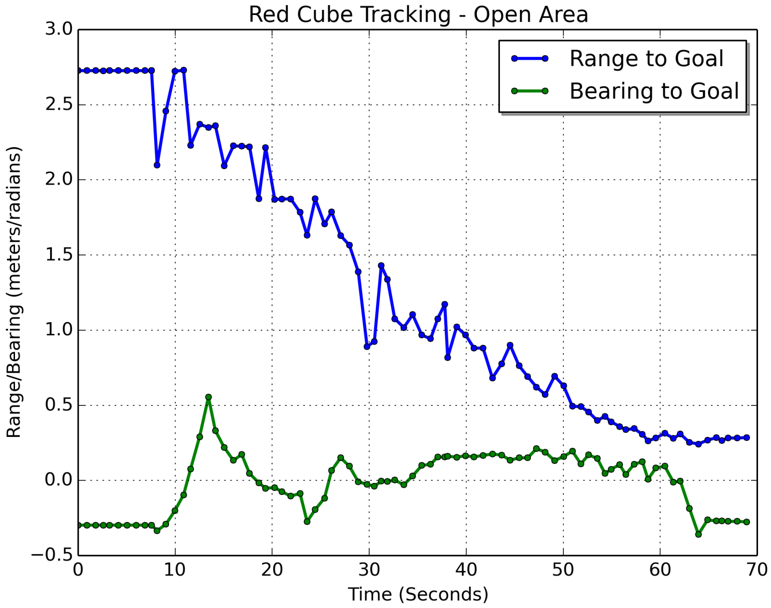

Figure 4: This figure plots the perceived range and bearing to the goal relative to the robot during the open arena experiment. The range to the goal decreases until the Goal Stopping Radius is reached.

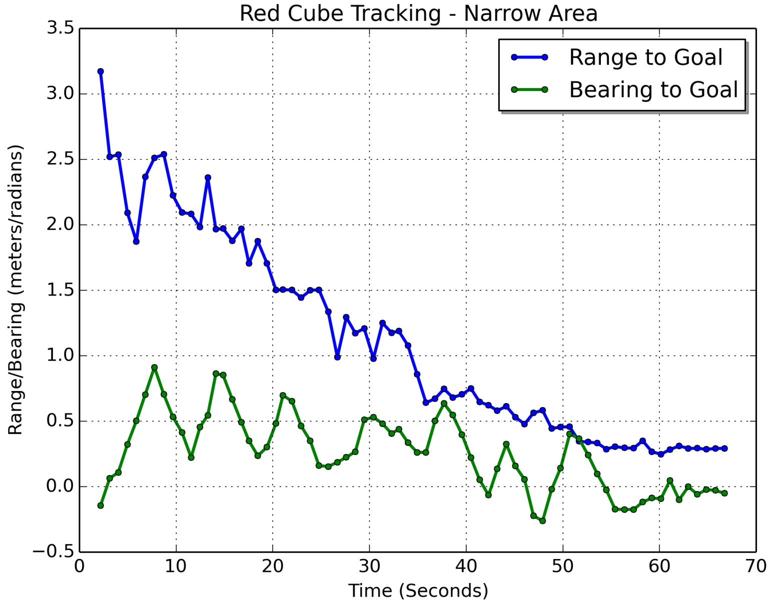

Figure 5: This figure plots the perceived range and bearing to the goal relative to the robot during the narrow opening arena experiment. The range to the goal decreases until the Goal Stopping Radius is reached.

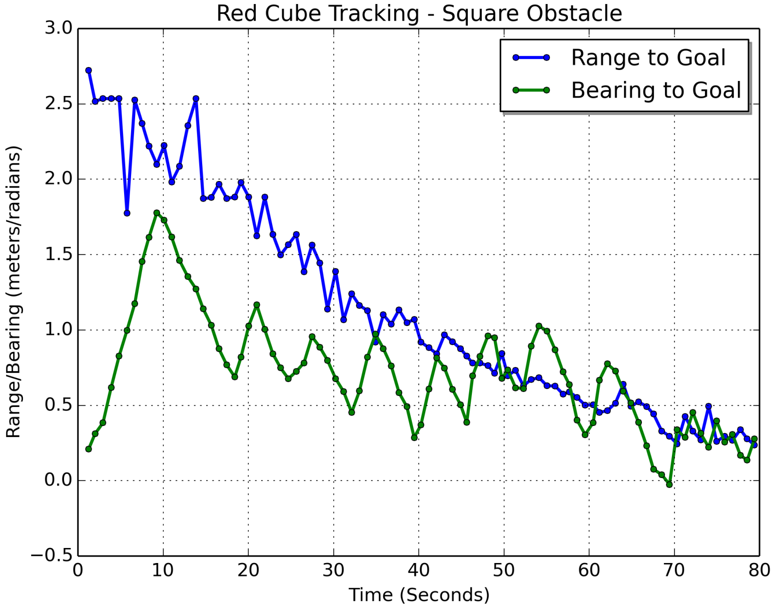

Figure 6: This figure plots the perceived range and bearing to the goal relative to the robot during the large obstacle arena experiment. The range to the goal decreases until the Goal Stopping Radius is reached.

1.3 Robot Pose Tracking

In order to record the pose of the robot during the experiments, a GoPro camera was mounted above the arena. This global camera could see the entire arena during the experiment, simplifying path analysis. The resultant videos show the robot traversing from the start location to the goal location while avoiding obstacles. The video was then analyzed to extract the approximate location of the robot in each frame and a path was produced from these samples.

1.3.1 Observed Path

As recorded by the global camera during the experiments, Figures 7, 8, and 9 show the robot as in traverses the arenas. The best fit path, explained in Section 1.3.2, is overlaid over each frame to better demonstrate the progression of the robot through the arena. In each of the experiments, the robot can be seen navigating from its starting position, to some position near the goal cube, while avoiding collision with obstacles. The robot does not approach the obstacles too closely, nor does it wander in the open areas. An issue that can be seen in each of the experiments, is the camera parallax between the head and feet of the robot as well as the goal cube and stand. This effect

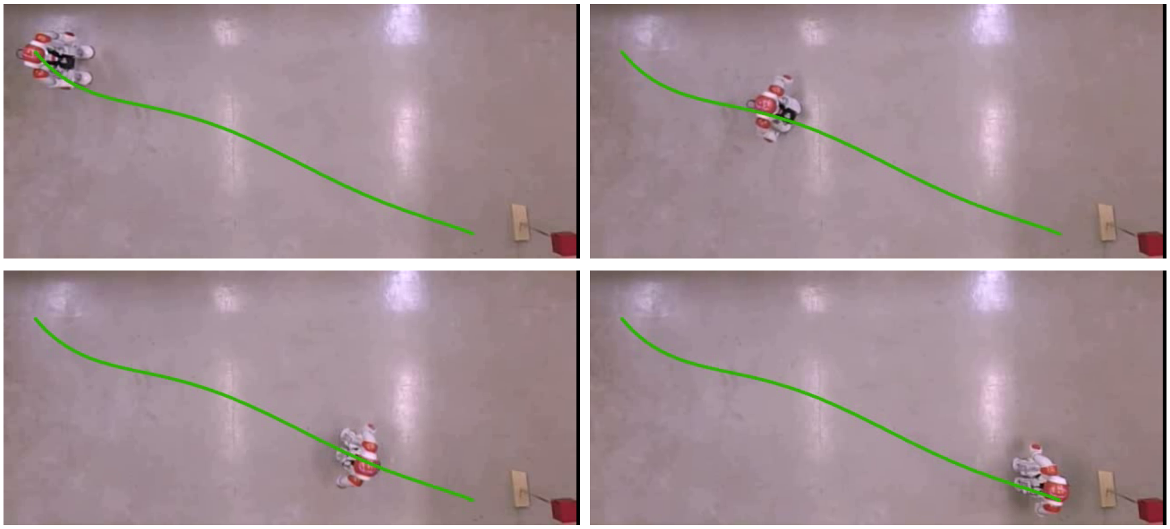

Figure 7: These images show the robot as it progresses through the open arena. The path of the robot is overlaid onto each of the images. The robot moves along a straight path to the goal.

exaggerates the farther an object is from the center of the frame. As the best fit path is a result of tracking the orange part of the Nao during the experiment, the path is distorted. This is especially apparent in Figure 9, where the path towards the bottom of the image seems to be farther from the obstacles than one might expect.

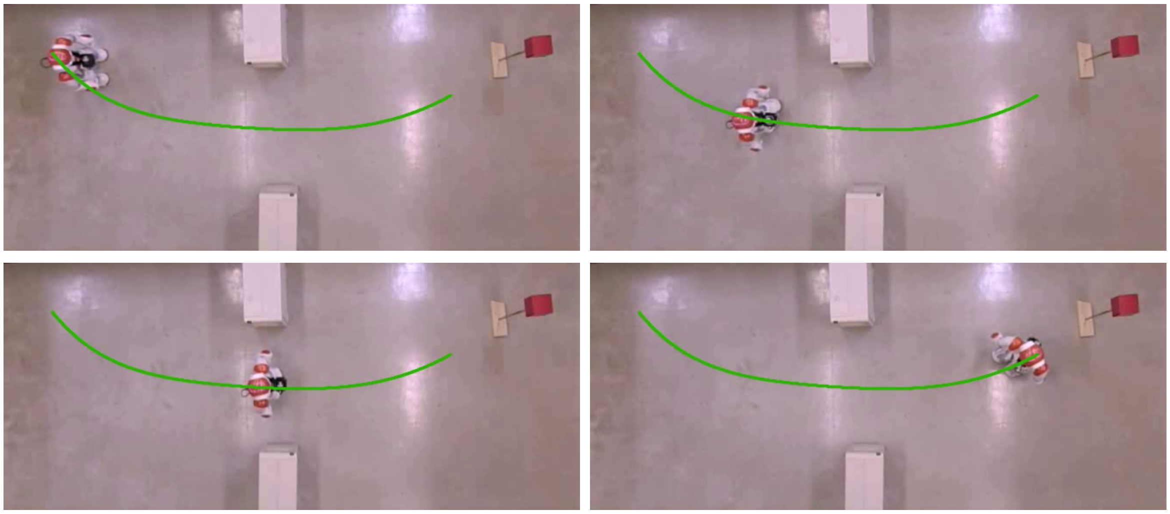

Figure 8: These images show the robot as it progresses through the narrow opening arena. The path of the robot is overlaid onto each of the images. The robot moves towards the middle of the narrow opening and then towards the goal.

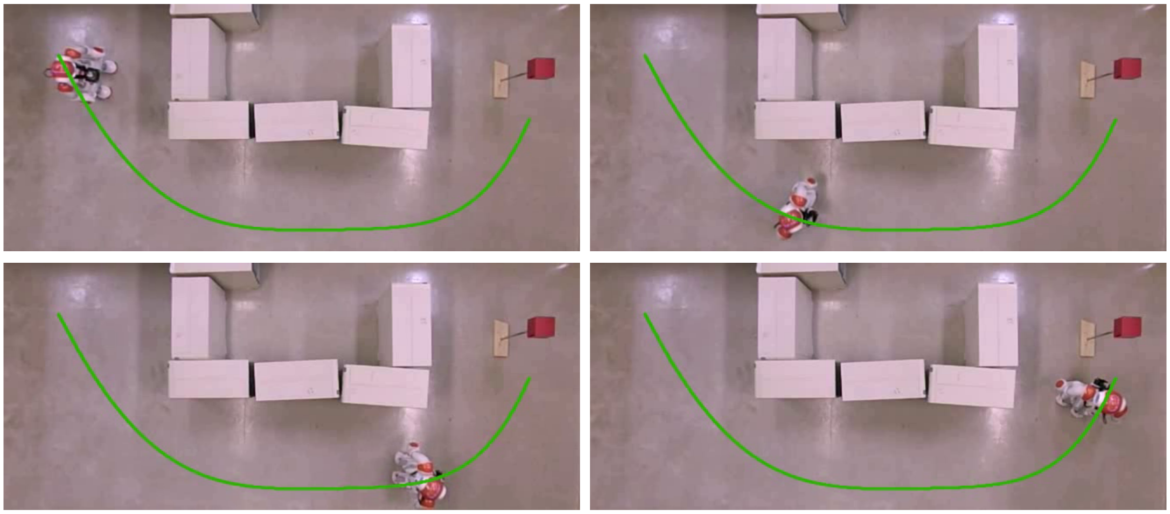

Figure 9: These images show the robot as it progresses through the large obstacle arena. The path of the robot is overlaid onto each of the images. The robot moves down from its starting location, along the corridor, and to the goal. Camera parallax distorts the path and makes it seem that the robot walks farther away from the obstacles during the middle of the run. The upper-right and lower-left images show that this is not the case.

1.3.2 Tracking Analysis

Using the global camera data from each of the experiments, the robot was tracked through each video to produce an approximation to the navigated path. The OpenCV library was used to process the images in Python. Briefly, the procedure used to produce the path from each video was:

1. Extract orange pixels from frame into a new image.
2. Filter that image using a closing kernel to reduce noise.
3. Find centroids of remaining pixel "blobs".
4. Eliminate blobs with small areas and join nearby blobs.
5. Process every video frame to produce array of centroids.
6. Group centroids using a density-based clustering algorithm.
7. Manually select the group corresponding to the robot path.
8. Fit a polynomial to the centroid clusters as the robot path estimate using least-squares.

A fifth-order polynomial was used to approximate the path of the robot because it was the lowest order polynomial fitting that produced a good result. The results from this procedure can be seen in Figures 10, 11, and 12 for the open arena, narrow opening arena, and large obstacle arena, respectively. The centroid samples and best-fit path are overlaid onto one another to show how the path approximates the path of the robot from the samples. In these plots, the samples clearly illustrate the oscillations in the walking gait. While every bipedal gait will produce some oscillations in the center-of-mass motion, the gait instabilities are amplified by the addition of the LIDAR mass as discussed in Section 1.1.1. This effect is especially evident in Figure 11 towards the middle of the plot. This corresponds to the area near the left side of the narrow opening. While it is not known why the magnitude of the oscillations at this point are so large, it is possible that they are not due to the nearby obstacle but due to an irregularity in the arena floor in this region. This is theorized because the large obstacle arena contained apertures which were the same width as the narrow opening. Despite this, Figure 12 does not show any oscillations approaching the magnitude of those seen in Figure 11.

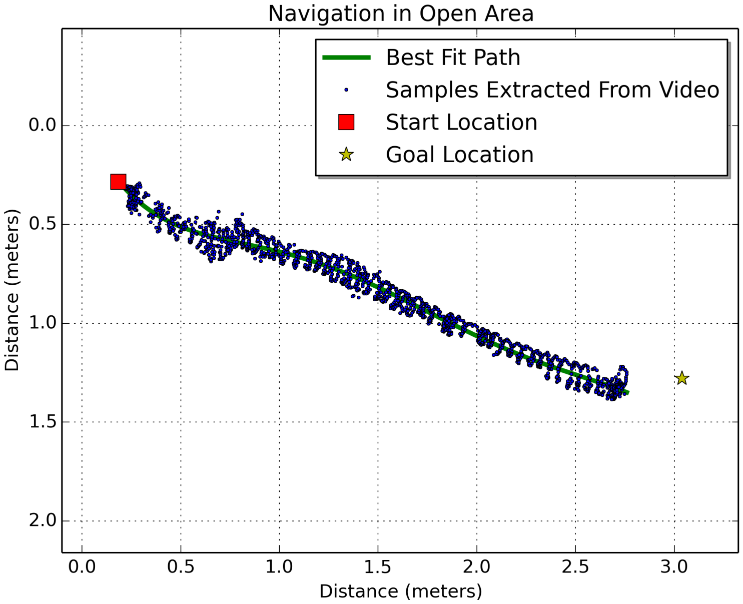

Figure 10: This plot shows the centroid samples extracted from the global camera video from the open arena experiment. A fifth order polynomial has been fit to the samples and overlaid onto the plot. The robot starting point is shown as a red square on the left and the goal point is shown as a yellow star on the right. The sample centroids clearly show the wobble in the gait. The robot stopped approximately $0.8m$ from the goal.

Lastly, as mentioned in Section 1.1.2, despite the Goal Stopping Radius from Table 1 being $0.3m$, the robot stops approximately $0.8m$ from the goal. This is close to double the parameter value, which is what was predicted as the cube was about twice as big as the "red ball" tracker expected it to be. The apparent size of the cube when approached from one of the corners, seen with greatest effect during the narrow opening experiment, would have also contributed to the robot stopping farther from the goal than intended.

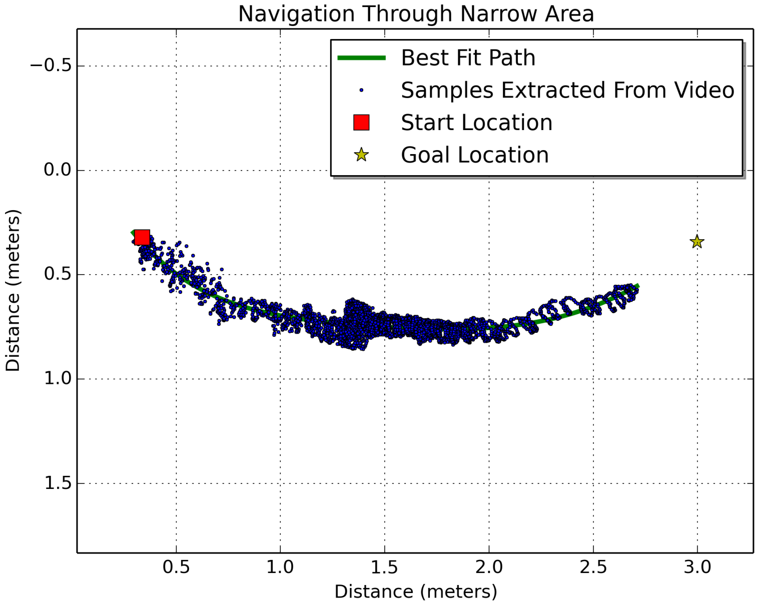

Figure 11: This plot shows the centroid samples and best-fit path from the narrow opening arena experiment. The oscillations shown by the samples can be seen to be the largest in the $(1.4, 0.75)$ region of the plot.

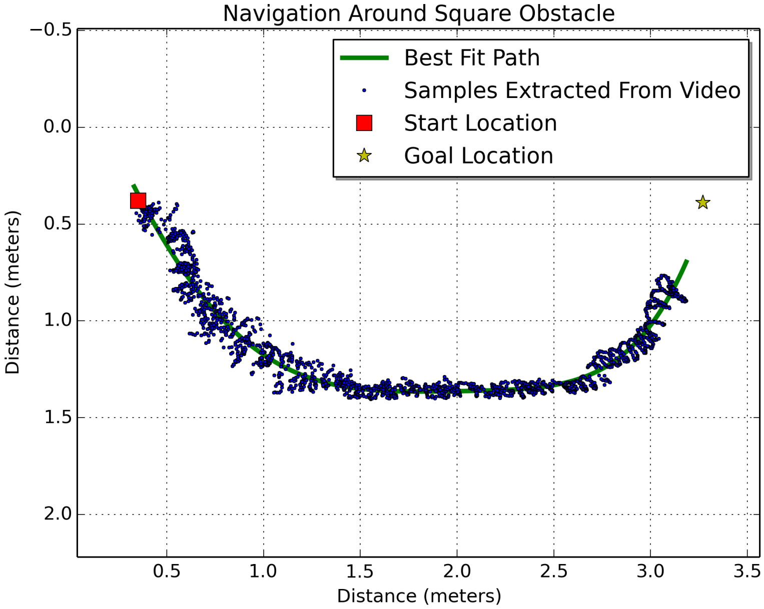

Figure 12: This plot shows the centroid samples and best-fit path from the large obstacle arena experiment. The oscillations reduce along the path towards the bottom of the plot, but this is more of a limitation of the image processing procedure and when viewing the video, it is clear the oscillations are of similar magnitude to those seen in the rest of the experiment.